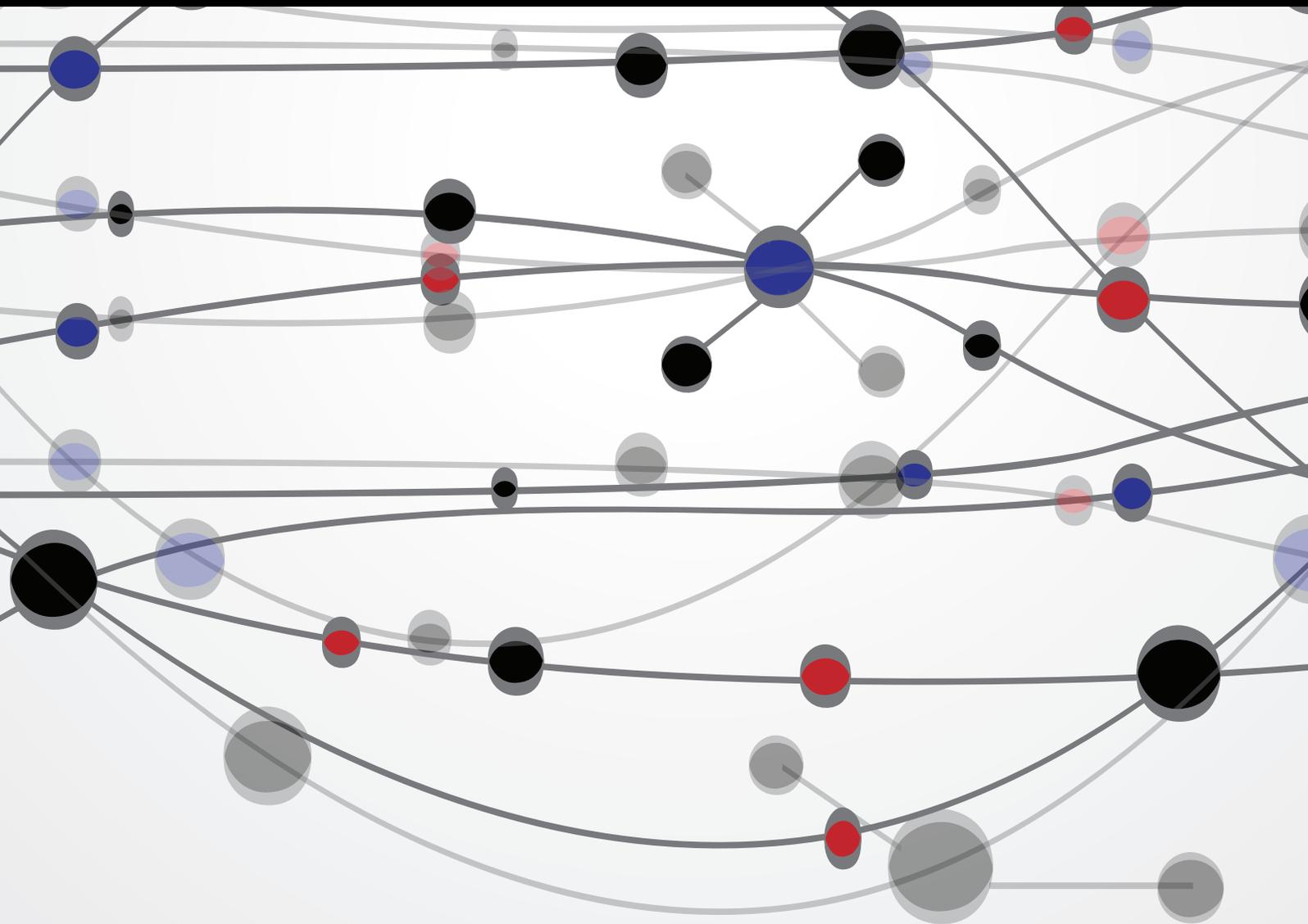


Recent Advances in Information Security

Guest Editors: Fei Yu, Chin-Chen Chang, Muhammad Khurram Khan, Tianjie Cao, and Mirjana Ivanovic





Recent Advances in Information Security

The Scientific World Journal

Recent Advances in Information Security

Guest Editors: Fei Yu, Chin-Chen Chang,
Muhammad Khurram Khan, Tianjie Cao, and Mirjana Ivanovic



Copyright © 2014 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in "The Scientific World Journal." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Contents

Recent Advances in Information Security, Fei Yu, Chin-Chen Chang, Muhammad Khurram Khan, Tianjie Cao, and Mirjana Ivanovic
Volume 2014, Article ID 562492, 4 pages

Security Enhanced Anonymous Multiserver Authenticated Key Agreement Scheme Using Smart Cards and Biometrics, Younsung Choi, Junghyun Nam, Donghoon Lee, Jiye Kim, Jaewook Jung, and Dongho Won
Volume 2014, Article ID 281305, 15 pages

Self-Adaptive Trust Based ABR Protocol for MANETs Using Q-Learning, Anitha Vijaya Kumar and Akilandeswari Jeyapal
Volume 2014, Article ID 452362, 9 pages

Spatiotemporal Access Model Based on Reputation for the Sensing Layer of the IoT, Yunchuan Guo, Lihua Yin, Chao Li, and Junyan Qian
Volume 2014, Article ID 671038, 15 pages

An Analysis of Security System for Intrusion in Smartphone Environment, Maya Louk, Hyotaek Lim, and HoonJae Lee
Volume 2014, Article ID 983901, 12 pages

SmartMal: A Service-Oriented Behavioral Malware Detection Framework for Mobile Devices, Chao Wang, Zhizhong Wu, Xi Li, Xuehai Zhou, Aili Wang, and Patrick C. K. Hung
Volume 2014, Article ID 101986, 11 pages

An Action-Based Fine-Grained Access Control Mechanism for Structured Documents and Its Application, Mang Su, Fenghua Li, Zhi Tang, Yinyan Yu, and Bo Zhou
Volume 2014, Article ID 232708, 13 pages

Quality of Protection Evaluation of Security Mechanisms, Bogdan Ksiezopolski, Tomasz Zurek, and Michail Mokkalas
Volume 2014, Article ID 725279, 18 pages

Malware Analysis Using Visualized Image Matrices, KyoungSoo Han, BooJoong Kang, and Eul Gyu Im
Volume 2014, Article ID 132713, 15 pages

Modeling the Propagation of Mobile Phone Virus under Complex Network, Wei Yang, Xi-liang Wei, Hao Guo, Gang An, Lei Guo, and Yu Yao
Volume 2014, Article ID 207457, 14 pages

Fuzzy-Based Trust Prediction Model for Routing in WSNs, X. Anita, M. A. Bhagyaveni, and J. Martin Leo Manickam
Volume 2014, Article ID 480202, 11 pages

Towards Dynamic Remote Data Auditing in Computational Clouds, Mehdi Sookhak, Adnan Akhunzada, Abdullah Gani, Muhammad Khurram Khan, and Nor Badrul Anuar
Volume 2014, Article ID 269357, 12 pages

Combining Digital Watermarking and Fingerprinting Techniques to Identify Copyrights for Color Images, Shang-Lin Hsieh, Chun-Che Chen, and Wen-Shan Shen

Volume 2014, Article ID 454867, 14 pages

A Comprehensive Review on Adaptability of Network Forensics Frameworks for Mobile Cloud Computing, Suleman Khan, Muhammad Shiraz, Ainuddin Wahid Abdul Wahab, Abdullah Gani, Qi Han, and Zulkanain Bin Abdul Rahman

Volume 2014, Article ID 547062, 27 pages

A Model Based Security Testing Method for Protocol Implementation, Yu Long Fu and Xiao Long Xin

Volume 2014, Article ID 632154, 10 pages

A Game-Theoretic Response Strategy for Coordinator Attack in Wireless Sensor Networks, Jianhua Liu, Guangxue Yue, Shigen Shen, Huiliang Shang, and Hongjie Li

Volume 2014, Article ID 950618, 10 pages

Calculating the Number of Cluster Heads Based on the Rate-Distortion Function in Wireless Sensor Networks, Mingxin Yang, Jingsha He, and Yuqiang Zhang

Volume 2014, Article ID 602875, 7 pages

Countermeasures to Avoid Noncooperation in Fully Self-Organized VANETs, Jezabel Molina-Gil, Pino Caballero-Gil, and Cándido Caballero-Gil

Volume 2014, Article ID 589563, 10 pages

A Regev-Type Fully Homomorphic Encryption Scheme Using Modulus Switching, Zhigang Chen, Jian Wang, Liqun Chen, and Xinxia Song

Volume 2014, Article ID 983862, 12 pages

Automating Risk Analysis of Software Design Models, Maxime Frydman, Guifré Ruiz, Elisa Heymann, Eduardo César, and Barton P. Miller

Volume 2014, Article ID 805856, 12 pages

A Provably Secure Revocable ID-Based Authenticated Group Key Exchange Protocol with Identifying Malicious Participants, Tsu-Yang Wu, Tung-Tso Tsai, and Yuh-Min Tseng

Volume 2014, Article ID 367264, 10 pages

Efficient and Provable Secure Pairing-Free Security-Mediated Identity-Based Identification Schemes, Ji-Jian Chin, Syh-Yuan Tan, Swee-Huay Heng, and Raphael C.-W. Phan

Volume 2014, Article ID 170906, 14 pages

Date Attachable Offline Electronic Cash Scheme, Chun-I Fan, Wei-Zhe Sun, and Hoi-Tung Hau

Volume 2014, Article ID 216973, 19 pages

A Secure RFID Authentication Protocol Adopting Error Correction Code, Chien-Ming Chen, Shuai-Min Chen, Xinying Zheng, Pei-Yu Chen, and Hung-Min Sun

Volume 2014, Article ID 704623, 12 pages

High Capacity Reversible Watermarking for Audio by Histogram Shifting and Predicted Error Expansion, Fei Wang, Zhaoxin Xie, and Zuo Chen
Volume 2014, Article ID 656251, 7 pages

Dual Key Speech Encryption Algorithm Based Underdetermined BSS, Huan Zhao, Shaofang He, Zuo Chen, and Xixiang Zhang
Volume 2014, Article ID 974735, 7 pages

Efficient Certificate-Based Signcryption Secure against Public Key Replacement Attacks and Insider Attacks, Yang Lu and Jiguo Li
Volume 2014, Article ID 295419, 12 pages

Average Gait Differential Image Based Human Recognition, Jinyan Chen and Jiansheng Liu
Volume 2014, Article ID 262398, 8 pages

A Secure and Fair Joint E-Lottery Protocol, Chin-Ling Chen, Yuan-Hao Liao, and Woei-Jiunn Tsaur
Volume 2014, Article ID 139435, 14 pages

Password-Only Authenticated Three-Party Key Exchange with Provable Security in the Standard Model, Junghyun Nam, Kim-Kwang Raymond Choo, Junghwan Kim, Hyun-Kyu Kang, Jinsoo Kim, Juryon Paik, and Dongho Won
Volume 2014, Article ID 825072, 11 pages

A Complete Hierarchical Key Management Scheme for Heterogeneous Wireless Sensor Networks, Chien-Ming Chen, Xinying Zheng, and Tsu-Yang Wu
Volume 2014, Article ID 816549, 13 pages

Reducing Side Effects of Hiding Sensitive Itemsets in Privacy Preserving Data Mining, Chun-Wei Lin, Tzung-Pei Hong, and Hung-Chuan Hsu
Volume 2014, Article ID 235837, 12 pages

Separable and Error-Free Reversible Data Hiding in Encrypted Image with High Payload, Zhaoxia Yin, Bin Luo, and Wien Hong
Volume 2014, Article ID 604876, 8 pages

A Coverage and Slicing Dependencies Analysis for Seeking Software Security Defects, Hui He, Dongyan Zhang, Min Liu, Weizhe Zhang, and Dongmin Gao
Volume 2014, Article ID 463912, 10 pages

Identifying Network Public Opinion Leaders Based on Markov Logic Networks, Weizhe Zhang, Xiaoqiang Li, Hui He, and Xing Wang
Volume 2014, Article ID 268592, 8 pages

On the Improvement of Wiener Attack on RSA with Small Private Exponent, Mu-En Wu, Chien-Ming Chen, Yue-Hsun Lin, and Hung-Min Sun
Volume 2014, Article ID 650537, 9 pages



Novel Image Encryption Scheme Based on Chebyshev Polynomial and Duffing Map,

Borislav Stoyanov and Krasimir Kordov

Volume 2014, Article ID 283639, 11 pages

Privacy-Preserving Location-Based Query Using Location Indexes and Parallel Searching in Distributed Networks, Cheng Zhong, Lei Liu, and Jing Zhao

Volume 2014, Article ID 751845, 7 pages

Editorial

Recent Advances in Information Security

**Fei Yu,¹ Chin-Chen Chang,² Muhammad Khurram Khan,³
Tianjie Cao,⁴ and Mirjana Ivanovic⁵**

¹ Peoples' Friendship University of Russia, Moscow 117198, Russia

² Feng Chia University, Taichung 40724, Taiwan

³ King Saud University, Riyadh 92144, Saudi Arabia

⁴ China University of Mining and Technology, Xuzhou 221000, China

⁵ University of Novi Sad, 21000 Novi Sad, Serbia

Correspondence should be addressed to Fei Yu; hunanyufei@126.com

Received 20 August 2014; Accepted 20 August 2014; Published 30 December 2014

Copyright © 2014 Fei Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recent years have witnessed rapid developments of the information communication technology and the next-generation Internet. We have seen the potential and value for new technologies from a variety of digital networks to various interconnected digital devices, which make it possible to realize the ubiquitous network and society. But, in such environment, copyright infringement behaviors, such as illicit copying, malicious distribution, unauthorized usage, and free sharing of copyright-protected digital contents, will also become a much more common phenomenon. Researchers, content industry engineers, and administrators attempt to resort to the state-of-the-art technologies and ideas to protect valuable digital contents and services assets against attacks and IP piracy in the emerging ubiquitous network.

This special issue aims to bring together related research works in the realm of ubiquitous network and multimedia contents security and further investigates and discusses trusted solutions and secure techniques for ubiquitous network as well as some open issues faced by digital contents, owners/rights, holders/multimedia, and services providers, who dedicate themselves to protect their intellectual property rights and business benefits.

This special issue includes a collection of 38 papers selected from 151 submissions to 17 countries or districts (Australia, Bulgaria, Canada, China, France, Hong Kong, India, Israel, Korea, Macau, Malaysia, Pakistan, Poland, Saudi Arabia, Spain, Taiwan, and USA). All submitted papers followed the same standard (peer-reviewed by at least three independent reviewers) as applied to regular submissions.

In the paper entitled “*Privacy-preserving location-based query using location indexes and parallel searching in distributed networks*” by C. Zhong et al., an efficient location-based query algorithm of protecting the privacy of the user in the distributed networks is given. This algorithm utilizes the location indexes of the users and multiple parallel threads to search and select quickly all the candidate anonymous sets with more users and their location information with more uniform distribution to accelerate the execution of the temporal-spatial anonymous operations and it allows the users to configure their custom-made privacy-preserving location query requests.

In the paper entitled “*Novel image encryption scheme based on Chebyshev polynomial and Duffing map*” by B. Stoyanov and K. Kordov, a novel image encryption algorithm based on dynamical chaotic systems is proposed. The developed encryption scheme combines Chebyshev polynomial based on permutation and substitution and Duffing map based on substitution. A precise security analysis on the novel encryption algorithm is given.

The paper entitled “*On the improvement of Wiener attack on RSA with small private exponent*” by M.-E. Wu et al. proposes a method, called EPF, to estimate the prime factors of an RSA modulus. With EPF, the cost of exhaustive search can further reduce to 2r-6 bits. As a conclusion, the authors would like to point out that with the continuous improvements in computational capability, the security levels are expected to be higher with the assistance of EPF and the security analysis should be considered more carefully.

In the paper entitled “*Identifying network public opinion leaders based on Markov logic networks*” by W. Zhang et al., a network opinion leader recognition method based on relational data was put forward and an opinion leader recognition system integrating public opinion data acquisition module, data characteristic selection, and fusion module as well as opinion leader discovery module based on Markov logic networks was designed.

In the paper entitled “*A coverage and slicing dependencies analysis for seeking software security defects*” by H. He et al. based on Reverse Data Dependence Analysis Model to extract data dependencies between program statements by analyzing the advantages and disadvantages of the more generally applicable CBFL and program structure slicing method, for shortcomings, the authors proposed a new fault localization method CPSS while retaining the advantages.

The paper entitled “*Separable and error-free reversible data hiding in encrypted image with high payload*” by Z. Yin et al. proposed a separable and error-free reversible data-hiding scheme in encrypted image, which significantly outperforms the previous methods in the three aspects of payload, PSNR, and error rate.

In the paper entitled “*Reducing side effects of hiding sensitive itemsets in privacy preserving data mining*” by C.-W. Lin et al., a novel hiding-missing-artificial utility (HMAU) algorithm is proposed to hide sensitive itemsets through transaction deletion. The transaction with the maximal ratio of sensitive to nonsensitive one is thus selected to be entirely deleted.

The paper entitled “*A complete hierarchical key management scheme for heterogeneous wireless sensor networks*” by C.-M. Chen et al. proposes a complete hierarchical key management scheme which only utilizes symmetric cryptographic algorithms and low cost operations for heterogeneous cluster-based WSN. The analysis and experiments demonstrate that the proposed scheme is secure and efficient; thus, it is suitable for heterogeneous cluster-based WSN.

The paper entitled “*Password-only authenticated three-party key exchange with provable security in the standard model*” by J. Nam et al. presents the first three-party PAKE protocol whose security is proven without any idealized assumptions in a model that captures insider attacks. The model that the authors used to prove the security of our protocol allows the adversary to ask corrupt queries and thus captures insider attacks as well as forward secrecy.

The paper entitled “*A secure and fair joint e-lottery protocol*” by C.-L. Chen et al. presents a novel joint e-lottery protocol using the multisignature and verifiable random function. Having been proved, the new mechanism can achieve the requirements of general electronic lotteries.

In the paper entitled “*Average gait differential image based human recognition*” by J. Chen and J. Liu, an average gait differential image based human recognition method is proposed. The Kernel idea of AGDI is to apply the average of differential image as the feature image and use the two-dimensional principal component analysis to extract features.

The paper entitled “*Dual key speech encryption algorithm based underdetermined BSS*” H. Zhao et al. presents a new dual key encryption scheme based on the underdetermined

BSS problem. The proposed algorithm for speech signals encryption can resist traditional attacks against the encryption system and, owing to approximate calculation, decryption becomes faster and more accurate.

The paper entitled “*High capacity reversible watermarking for audio by histogram shifting and predicted error expansion*” by F. Wang et al. presents a novel reversible audio watermarking algorithm based on improved prediction error expansion and histogram shifting. Experiments show that this algorithm improves the SNR of embedded audio signals and embedding capacity, drastically reduces location map bits length, and enhances capacity control capability.

The paper entitled “*A secure RFID authentication protocol adopting error correction code*” by C.-M. Chen et al. presents a lightweight mutual authentication protocol adopting error correction code for RFID. Compared with other lightweight protocols, the proposed protocol provides stronger resistance to tracing attacks, compromising attacks, and replay attacks.

The paper entitled “*Date attachable offline electronic cash scheme*” by C.-I Fan et al. proposes a provably secure and efficient offline e-cash scheme with date attachability based on the blind signature technique, where expiration date and deposit date can be embedded in an e-cash simultaneously.

The paper entitled “*Efficient and provable secure pairing-free security-mediated identity-based identification schemes*” by J.-J. Chin et al. proposed two SM-IBI schemes that have an instant revocation feature and are very efficient. The authors’ schemes outperform the only pairing-based SM-IBI currently known and are provably secure in the random oracle model against both passive and active/concurrent attackers.

In The paper entitled “*A provably secure revocable ID-based authenticated group key exchange protocol with identifying malicious participants*” T.-Y. Wu et al. have fused the Tseng-Tsai R-IDPKS system and a noninteractive confirmed computation technique to propose the first RID-AGKE protocol with identifying malicious participants. The framework and security notions for RID-AGKE protocols have been defined to formalize the possible threats and attacks.

Paper “*Automating risk analysis of software design models*” by M. Frydman et al. describes an approach to reduce the need for costly human expertise to perform risk analysis in software, which is common in secure development methodologies, by automating threat modeling. Reducing the dependency on security experts aims at reducing the cost of secure development by allowing non-security-aware developers to apply secure development with little to no additional cost, making secure development more accessible.

The paper entitled “*A Regev-type fully homomorphic encryption scheme using modulus switching*” by Z. Chen et al. recommends concrete parameter values of our proposed scheme and provide security analysis. The result shows that the modified FHE scheme is more efficient than the original Brakerski scheme in the same security level.

The paper entitled “*Countermeasures to avoid noncooperation in fully self-organized VANETs*” by J. Molina-Gil et al. proposes a new vehicular communication system based on mobile phones for fully distributed and decentralized networks. In these networks, communications depend on

individual nodes, which could decrease the efficiency and reliability of transmitted information.

The paper entitled “*Calculating the number of cluster heads based on the rate-distortion function in wireless sensor networks*” by M. Yang et al. proposes a method for the calculation of the number of cluster heads based on the rate-distortion function after establishing an energy consumption model according to the data fusion framework in WSNs.

The paper entitled “*A game-theoretic response strategy for coordinator attack in wireless sensor networks*” by J. Liu et al. proposes an adaptive coordinator selection algorithm using game and fuzzy logic aiming at both minimizing the average number of hops and maximizing network lifetime. The proposed game model consists of two interrelated formulations: a stochastic game for dynamic defense and a best response policy using evolutionary game formulation for coordinator selection.

The paper “*A model based security testing method for protocol implementation*” by Y. L. Fu and X. L. Xin proposes an extended model of IOLTS to describe the legal roles and intruders of security protocol implementations and then combine them together to generate the suitable test cases to verify the security of protocol implementation. The proposed model inherits the clarity of finite automata and can describe the security properties and most of the protocol behaviors with the definition of transition.

The paper entitled “*A comprehensive review on adaptability of network forensics frameworks for mobile cloud computing*” by S. Khan et al. discussed the functions, approaches, and structures of current NFFs. The authors conclude that new research roadmaps and programs are required to overcome the issues and challenges faced by CSPs. Standardized rules, secure reference models, protocols, trust architectures, legal contemplation, technological development, and a global regularity body should be established. These requirements can be achieved by harmonizing the efforts of industrial experts, academic researchers, and investigators under legal entity bodies.

The aim of the paper entitled “*Combining digital watermarking and fingerprinting techniques to identify copyrights for color images*” by S.-L. Hsieh et al. is to present a copyright identification scheme for color images that takes advantage of the complementary nature of watermarking and fingerprinting. The experimental results showed that when the watermarked image suffers moderate attacks, *watermarking verification* alone is enough to identify the copyright and there is no need for *fingerprinting verification*.

The paper entitled “*Towards dynamic remote data auditing in computational clouds*” by M. Sookhak et al. proposes an effectual RDA technique based on algebraic signature properties for cloud storage system and also presents a new data structure capable of efficiently supporting dynamic data operations like append, insert, modify, and delete. The comparative analysis with the state-of-the-art RDA schemes shows that the proposed scheme is secure and highly efficient in terms of the computation and communication overhead on the auditor and server.

The paper entitled “*Fuzzy-based trust prediction model for routing in WSNs*” by X. Anita et al. proposed FTTPR

protocol to effectively thwart black hole attack, on-off attack, conflicting behavior attack, and bad-mouthing attack. It employed a fuzzy-based trust prediction model to predict the future behavior of a neighboring node based on its historical behavior, trust fluctuations, and recommendation inconsistency.

In the paper entitled “*Modeling the propagation of mobile phone virus under complex network*” by W. Yang et al., the M-SIS and M-SIR propagation models for mobile phone viruses are proposed, combining with the structural characteristics of the complex network. The M-SIR propagation model is suitable to describe the vulnerability-exploiting mobile phone virus. It reflects the characteristic of the mobile virus, which spreads by exploiting vulnerabilities, and the mobile phone can be immune to the mobile phone virus after virus removal and patching.

The paper entitled “*Malware analysis using visualized image matrices*” by K. S. Han et al. proposes a novel malware visual analysis method that contains not only a visualization method to convert binary files into images, but also a similarity calculation method between these images. The proposed method generates RGB-colored pixels on image matrices using the opcode sequences extracted from malware samples and calculates the similarities for the image matrices.

The paper entitled “*Quality of protection evaluation of security mechanisms*” by B. Ksiezopolski et al. proposes a model for QoP evaluation of security mechanisms. Owing to this model, one can quantify the influence of particular security mechanisms on ensuring security attributes. An additional contribution of the paper is the implementation of the security mechanisms evaluation tool (SMETool) which supports the presented method.

A novel approach described to aid breast cancer detection and classification using digital mammograms is presented by X.-S. Zhang in his paper entitled “*A new approach for clustered MCs classification with sparse features learning and TWSVM*.” The proposed method is based on sparse feature learning and representation, which expresses a testing sample as a linear combination of the built vocabulary (training samples).

The paper entitled “*An action-based fine-grained access control mechanism for structured documents and its application*” by M. Su et al. presents an action-based fine-grained access control mechanism for structured documents. By defining the objective describing model, it could support the permission management on chapters, pages, sections, words, and pictures of structured documents.

The paper entitled “*SmartMal: a service-oriented behavioral malware detection framework for mobile devices*” by C. Wang et al. presents SmartMal—a novel service-oriented behavioral malware detection framework for vehicular and mobile devices. The proposed framework relies on client-server architecture, the client continuously extracts various features and transfers them to the server and the server’s main task is to detect anomalies using state-of-art detection algorithms.

In the paper entitled “*A new gravitational particle swarm optimization algorithm for the solution of economic emission dispatch in wind-thermal power system*” by S. Jiang et al., a new

hybrid optimization approach, namely, gravitational particle swarm optimization algorithm (GPSOA), is proposed to solve economic emission dispatch (EED) problem including wind power.

The paper entitled “*An analysis of security system for intrusion in smartphone environment*” by M. Louk et al. uses Android OS because the operating system is frequently attacked by cybercriminals. Monitoring, detecting, tracking, and notification are used not only to check new application before being installed into the smartphone, but also to detect suspicious behavior activity in real time.

The paper entitled “*Spatiotemporal access model based on reputation for the sensing layer of the IoT*” by Y. Guo et al. proposes a model that combines space and time with reputation to control access to the information within the sensing layer of the IoT. This model is called spatiotemporal access control based on reputation (STRAC).

The paper entitled “*Security enhanced anonymous multi-server authenticated key agreement scheme using smart cards and biometrics*” by Y. Choi et al. proposes a security enhanced anonymous multiserver authenticated key agreement scheme which addresses all the weaknesses identified in Chuang and Chen’s scheme.

The paper entitled “*Self-adaptive trust based ABR protocol for MANETs using Q-learning*” by A. V. Kumar et al. focuses on computing a score using Q-learning to weigh the trust of a particular node over associativity based routing (ABR) protocol.

The paper entitled “*Efficient and privacy-preserving metering protocols for smart grid systems*” by H. J. Jo and D. H. Lee proposes two protocols. The first protocol is based on the signcryption algorithm and achieves weak confidentiality; it is robust against node compromise attacks. The second one is an extended version of the first one; it satisfies strong confidentiality by using the Paillier homomorphic encryption algorithm.

Acknowledgments

In particular, we would like to acknowledge the program committee members of Sixth International Symposium on Information Processing (ISIP 2013) and Seventh International Workshop on Computer Science and Engineering (WCSE 2014). This issue contains revised and expanded versions of selected quality papers presented either at the Sixth International Symposium on Information Processing (ISIP 2013) or at the Seventh International Workshop on Computer Science and Engineering (WCSE 2014). We wish to express our deepest thanks to the program committee members for their help in selecting papers for this issue and especially the referees of the extended versions of the selected papers for their thorough reviews under a tight time schedule. The first conference, ISIP 2013 took place on December 21-22, 2013, in Changsha, China, and was cosponsored by Jiangxi University of Science and Technology, China; Peoples’ Friendship University of Russia, Russia; South China University of Technology, China; Feng Chia University, Taiwan; Henan Polytechnic University, China; Nanchang Hang Kong University, China; and Jiangxi University of Science and

Technology, China. WCSE 2014, took place on June 19-20, 2014, in Changsha, China, and was cosponsored by Peoples’ Friendship University of Russia, Russia; Feng Chia University, Taiwan; Fudan University, China; South China University of Technology, China; Henan Polytechnic University, China; Nanchang Hang Kong University, China; and Jiangxi University of Science and Technology, China. In closing, we would like to take this opportunity to thank the authors for the efforts they put in the preparation of the papers and in keeping the deadlines set by editorial requirements. We hope that you will enjoy reading this special issue as much as we did putting it together.

Fei Yu
Chin-Chen Chang
Muhammad Khurram Khan
Tianjie Cao
Mirjana Ivanovic

Research Article

Security Enhanced Anonymous Multiserver Authenticated Key Agreement Scheme Using Smart Cards and Biometrics

**Younsung Choi,¹ Junghyun Nam,² Donghoon Lee,¹ Jiye Kim,¹
Jaewook Jung,¹ and Dongho Won¹**

¹ Department of Computer Engineering, Sungkyunkwan University, 2066 Seoburo, Suwon, Gyeonggi-do 440-746, Republic of Korea

² Department of Computer Engineering, Konkuk University, 268 Chungwondaero, Chungju, Chungcheongbuk-do 380-701, Republic of Korea

Correspondence should be addressed to Dongho Won; dhwon@security.re.kr

Received 14 March 2014; Revised 28 July 2014; Accepted 29 July 2014; Published 8 September 2014

Academic Editor: Fei Yu

Copyright © 2014 Younsung Choi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An anonymous user authentication scheme allows a user, who wants to access a remote application server, to achieve mutual authentication and session key establishment with the server in an anonymous manner. To enhance the security of such authentication schemes, recent researches combined user's biometrics with a password. However, these authentication schemes are designed for single server environment. So when a user wants to access different application servers, the user has to register many times. To solve this problem, Chuang and Chen proposed an anonymous multiserver authenticated key agreement scheme using smart cards together with passwords and biometrics. Chuang and Chen claimed that their scheme not only supports multiple servers but also achieves various security requirements. However, we show that this scheme is vulnerable to a masquerade attack, a smart card attack, a user impersonation attack, and a DoS attack and does not achieve perfect forward secrecy. We also propose a security enhanced anonymous multiserver authenticated key agreement scheme which addresses all the weaknesses identified in Chuang and Chen's scheme.

1. Introduction

With the rapid growth of internet technology, a system providing various services using the network often consists of many different servers around the world. The distribution of the remote system hardware allows its users to access resources efficiently and conveniently. In multiple server environments, an authentication mechanism is required to achieve a high level of security [1]. Lamport [2] first proposed a password authentication scheme for communication through an insecure channel. However, Lamport's scheme requires the server to manage a password table and is, thus, vulnerable to stolen-verifier attacks. To resist this attack, several researchers proposed improved password-based authentication schemes using smart cards. But, these schemes are still easily broken by simple dictionary attacks due to the low entropy of passwords and because the information stored in smart cards could be extracted by physically

monitoring power consumption [3, 4]. Therefore, many other researchers have combined users' biometrics and passwords to enhance the security of their user authentication schemes for multiserver environments; see, for example, references [5–7] for earlier work in this domain. Every human being has a different biometrics, and thus, it is difficult for the adversary to compute the biometric information [8, 9].

Relatively recently, D. Yang and B. Yang [10] and Yoon and Yoo [11] independently introduced a biometric-based multiserver authentication scheme. But, these schemes still do not consider user anonymity which has been identified as a major security property for privacy protection in many applications, including location-based services, anonymous web browsing, e-voting, and mobile roaming services. Moreover, D. Yang and B. Yang's scheme requires users to perform expensive exponentiation operations, while Yoon and Yoo's scheme, as demonstrated by He [12], is vulnerable to a privileged insider attack, a masquerade attack, and a stolen smart card attack.

Recently, Chuang and Chen [13] proposed an anonymous multiserver authenticated key agreement scheme to address the weaknesses in the D. Yang and B. Yang's scheme [10] and the Yoon-Yoo scheme [11]. This scheme is based on nonces and is very efficient in that it only requires users to perform hash function evaluations. Chuang and Chen claimed that their scheme satisfies all the desired security-related properties: anonymity, absence of verification tables, mutual authentication, resistance to forgery attack, resistance to modification attacks, resistance to replay attacks, fast error detection, resistance to off-line guessing attacks, resistance to insider attacks, simple and secure password choice and modification, biometric template protection, and session key agreement. However, we found that Chuang and Chen's scheme has various security problems. According to our analysis given in this paper, Chuang and Chen's scheme is vulnerable to a masquerade attack, a smart card attack, a user impersonation attack, and a denial-of-service (DoS) attack and does not achieve perfect forward secrecy. To solve these security problems with Chuang and Chen's scheme, we propose an improved anonymous multiserver authenticated key agreement scheme using a smart card together with biometrics and passwords.

The remainder of this paper is organized as follows. Section 2 describes security and efficiency requirements for anonymous user authentication schemes in multiserver environments. Section 3 briefly reviews Chuang and Chen's authentication scheme, while Section 4 provides a detailed security analysis on the scheme. Section 5 presents our security-enhanced authentication scheme and shows how the security weaknesses of Chuang and Chen's scheme are addressed in our scheme. Section 6 analyzes our scheme in terms of both security and efficiency. Section 7 concludes the paper.

2. Requirements for Multiserver Authentication Schemes

Most conventional password authentication methods, when they are deployed in a multiple server environment, require each network user not only to log into various remote servers repetitively but also to remember many sets of identities and passwords. Such inefficiency and complexity easily lead to the exposure of users' identities and passwords and necessarily make it difficult to manage the shared secret keys among the involved participants. Moreover, those conventional authentication methods usually do not provide user anonymity. In contrast, an anonymous multiserver authentication scheme is designed to allow users to be authenticated by multiple servers via only one registration with the registration center [1]. Figure 1 shows a framework of an anonymous user authentication system in a multiserver environment.

2.1. Security Properties. Various security requirements for a multiserver authentication scheme have been suggested in the previous studies [1, 7, 10, 13–24]. The most essential security properties include the following.

- (S1) *Anonymity:* anonymity is of increasing importance and is achieved when the user's identity is not disclosed to an unauthorized party.
- (S2) *Mutual authentication:* mutual authentication means that the two parties, user and server, authenticate each other. That is, both user and server are assured of each other's identity.
- (S3) *Session key agreement:* the user and server securely agree on a session key to be used for protecting their subsequent communications.
- (S4) *Perfect forward secrecy:* perfect forward secrecy means that a session key derived from a set of long-term keys will not be compromised if one of the long-term keys is compromised in the future.

2.2. Attack Resistance. To achieve these security properties, a multiserver authentication scheme has to resist various kinds of attacks. The most typical attacks include the following

- (A1) *Replay attack:* an adversary intercepts data transmissions for the purpose of making use of that data in some manner. Typically, this type of attack involves copying and possibly altering the data in various ways before releasing it for delivery to the intended recipient.
- (A2) *Modification attack:* an adversary intercepts the authentication message and attempts to modify it for illegal authentication.
- (A3) *Stolen-verifier attack:* an adversary steals the password-verifier from the server and directly uses it to masquerade as a legitimate user.
- (A4) *Off-line guessing attack:* an adversary guesses a password and verifies it in an off-line environment. The information stored in the smart card is often used in such an attack.
- (A5) *Forgery attack:* a malicious yet legitimate user attempts to forge an authentication message of another legitimate user.
- (A6) *Insider attack:* an insider attack literally means an attack mounted by a malicious insider. Malicious insiders have a distinct advantage over external adversaries because they have an authorized system access and also may be familiar with the network architecture and system policies/procedures. Typically, malicious insiders want to acquire users' private information such as their password and biometrics.
- (A7) *Masquerade attack:* an adversary is authenticated by the server using a fake user ID.
- (A8) *Smart card attack:* an adversary is authenticated by the server by using only the information obtained from a user's smart card but without the password or biometrics of the user.
- (A9) *User impersonation attack:* an adversary impersonates a legitimate user using only the user's smart card but without the password or biometric of the user.

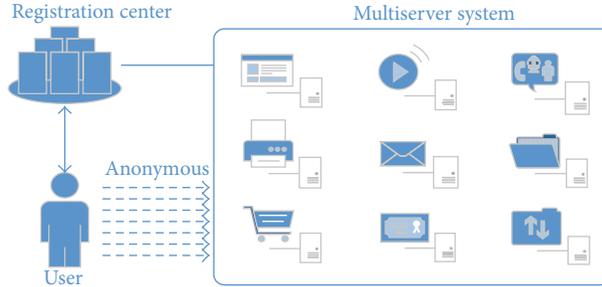


FIGURE 1: Framework of a multiserver authentication system.

(A10) *DoS Attack*. A DoS attack is any event that diminishes or eliminates a network’s capability of performing its expected function. In other words, an adversary mounts a DoS attack to make the server unavailable.

2.3. *Efficiency Measures*. Efficiency is an important consideration in evaluating any schemes or protocols. The efficiency of a multiserver authentication scheme can be measured by the following metrics.

- (E1) *Single registration*: a single point of registration ought to allow users to gain access to all the servers in the system.
- (E2) *Simple and secure password modification*: the system should allow users to choose and change their passwords easily and securely. In other words, each user should be able to change their passwords without the help of any third trusted party once the authenticity of the user is verified by its smart card.
- (E3) *Fast error detection*: the smart card needs to check the user’s incorrect password or any other discrepancy quickly.
- (E4) *Low computational cost*: the computational cost incurred by the scheme should be minimized for the participants.

3. A Review of Chuang and Chen’s Scheme

This section describes Chuang and Chen’s anonymous multiserver authenticated key agreement scheme which involves four phases: server registration, user registration, login and authentication, and password change. For convenience, the notations used throughout this paper are summarized in Notation Section.

3.1. *The Server Registration Phase*. The application server sends the RC a join message if it would like to become an authorized server. Then, the RC replies with the key (PSK) to the server through a secure channel. And then, the authorized server uses the PSK to check the user’s authentication message. If the server needs to obtain the PSK from the RC to perform the authentication phase every session, authentication delay and the communication cost between the RC and the servers will increase substantially,

but this scheme and proposed scheme register only once so they are efficient.

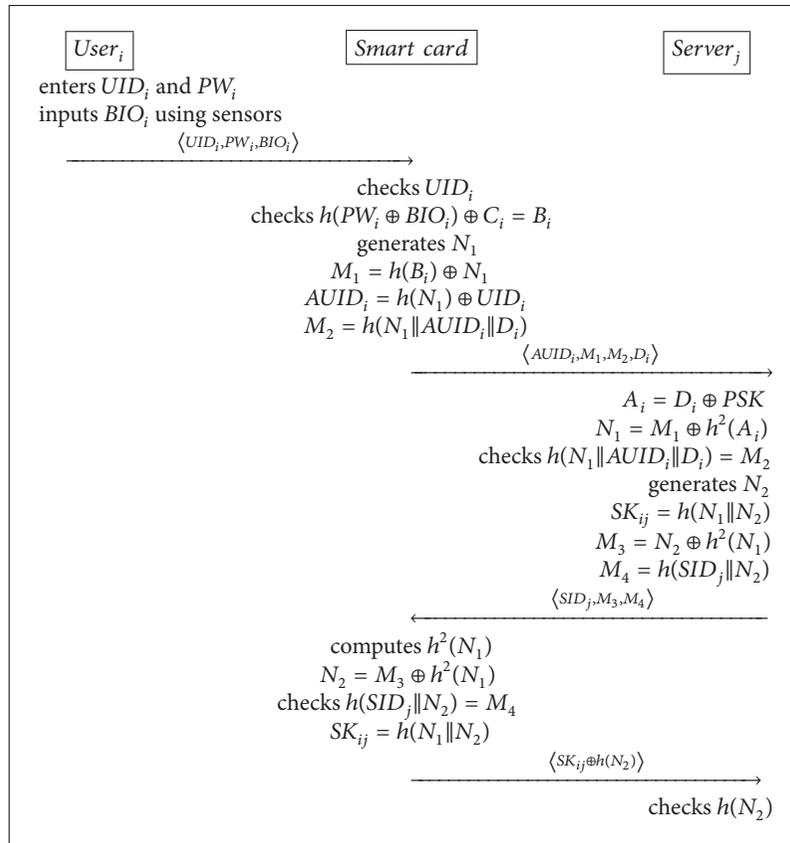
3.2. *The User Registration Phase*. For a user $user_i$, this phase is performed only once when $user_i$ registers itself with the registration center RC.

- (1) $user_i$ chooses his identity UID_i and password PW_i freely and inputs his biometrics BIO_i and sends the identity $user_i$ and $h(PW_i \oplus BIO_i)$ to RC via a secure channel.
- (2) RC computes $A_i = h(UID_i \| x)$ and $B_i = h^2(UID_i \| x) = h(A_i)$ and $C_i = h(PW_i \| BIO_i) \oplus B_i$ and $D_i = PSK \oplus A_i$ and issues $user_i$ a smart card loaded with $\langle UID_i, h(\cdot), B_i, C_i, D_i \rangle$.

3.3. *The Login and Authentication Phase*. In this phase, $user_i$ logs in to the smart card and is authenticated by server j . In login phase, is executed to check the user’s legality. The smart card can detect an error event immediately using the user’s identification, password, and biometrics information. And then, the smart card computes $\langle AUID_i, M_1, M_2, D_i \rangle$ for the authentication. In authentication phase, the smart card sends authentication messages to the server j after the user i finishes the login phase successfully. The smart card never send user’s real identity to execute the authentication phase for providing the user’s anonymity. During the phase, the session-key establishment is conducted between $user_i$ and server j . Algorithm 1 depicts how the login and authentication phase works.

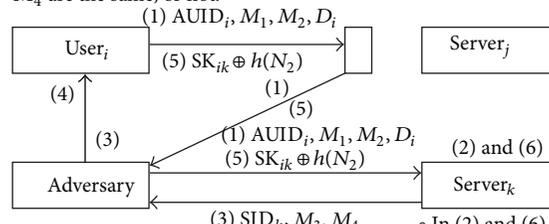
3.4. *The Password Change Phase*. One of the general guidelines to get better password security is to ensure that passwords are changed at regular intervals. Chuang and Chen’s scheme allows legitimate users to freely change their passwords:

- (1) $user_i$ inserts his smart card into a card reader and enters both the current password PW_i and the new password PW_i^* .
- (2) The smart card checks UID_i and $h(PW_i \oplus BIO_i) \oplus C_i = B_i$.
- (3) The smart card computes $C_i^* = C_i \oplus h(PW_i \oplus BIO_i) \oplus h(PW_i^* \oplus BIO_i)$ and replaces C_i with C_i^* .



ALGORITHM 1: Login and authentication phase of Chuang and Chen's scheme.

- In (4), user_i does not check whether server_k wants to be authenticated with user_i or not.
- User_i only checks whether the SID in message (4) and the SID in M₄ are the same, or not.



- In (2) and (6), server_k does not check whether user_i wants to be authenticate with server_k, or not.
- Adversary can be authenticated with server_k.

FIGURE 2: Masquerade attack on Chuang and Chen's scheme.

4. Security Vulnerabilities in Chuang and Chen's Scheme

We analyze Chuang and Chen's scheme and figure out some security vulnerabilities. Their scheme is vulnerable to the masquerade attack, smart card attack, user impersonation attack, and DoS attack and does not achieve perfect forward secrecy.

4.1. A Masquerade Attack. Chuang and Chen's scheme is vulnerable to user masquerade attack. An adversary can be authenticated to another server_k using the messages that user_i sends to server_j for authentication. Figure 2 describes the masquerade attack on Chuang and Chen's scheme. When the user_i wants to be authenticate with server_j, the user_i logs on the smart card and then sends a message (1) to the server_j. After an adversary intercepts the message (1), the adversary

will send it to another server $server_k$. This is because that message (1) does not include about the server $server_j$ as follows:

$$\begin{aligned} \text{Message (1)} &= \langle \text{AUID}_i, M_1, M_2, D_i \rangle, \\ \text{AUID}_i &= h(N_1) \oplus \text{UID}_i, \\ M_1 &= h(B_i) \oplus N_1, \\ M_2 &= h(N_1 \| \text{AUID}_i \| D_i), \\ D_i &= A_i \oplus \text{PSK}. \end{aligned} \quad (1)$$

So the server $server_k$ executes operation (2) and sends the message (3) to the adversary without any suspicion of the attack. The adversary forwards the message (3) to the user $user_i$. The user $user_i$ does not check the SID_j of the server $server_j$. It only checks the sameness with the SID of M_4 and the SID of the message (3) as follows:

$$\begin{aligned} \text{Message (3)} &= \langle \text{SID}_j, M_3, M_4 \rangle, \\ M_4 &= h(\text{SID}_j \| N_2). \end{aligned} \quad (2)$$

So the user $user_i$ executes operation (4) and sends message (5) to server $server_j$ without any suspicion of the attack. Then, an adversary intercepts the message (5) and sends it to another server $server_k$. Finally, the adversary can be authenticated with server $server_k$. Therefore, the adversary can masquerade as a legitimate user to server $server_k$. In this way, the scheme becomes vulnerable to the masquerade attack.

The server $server_k$ cannot check whether user $user_i$ wants to be authenticated by server $server_k$ or not. Thus server $server_k$ authenticates all legitimate messages though these message are not sent to server $server_k$. And user $user_i$ does not check whether server $server_j$ wants to be authenticated with user $user_i$. Thus user $user_i$ authenticates all legitimate messages though these message are sent by server $server_k$. The user $user_i$ only checks whether SID in message (3) and SID in M_4 are the same or not. To solve this problem, the destination of message is added to authentication messages. So the information about SID of server $server_j$ has to be added to the message (1), and this means that user $user_i$ want to be authenticated with server $server_j$, not server $server_k$. And the information about AUID of user $user_i$ has to be added to message (3); it means that the server $server_j$ wants to be authenticated with anonymous user $user_i$.

4.2. A Smart Card Attack. When an adversary gets or steals the user's smart card, the adversary can compute the session key between the user $user_i$ and server $server_j$ without the user's password or biometric information. So the adversary can decrypt the all encrypted communications between the user $user_i$ and server $server_j$ because the adversary can compute all previous session keys. Algorithm 2 describes the smart card attack on Chuang and Chen's scheme.

When the adversary obtains the user's smart card, the adversary can extract information about the smart card using a side-channel attack such as SPA (simple power analysis) or DPA (differential power analysis). The adversary can obtain B_i in the user's smart card and M_1, M_3 in the public

communication channel. Then, the adversary can compute N_1 using M_1 and $h(B_i)$ and N_2 using M_3 and $h^2(N_1)$. Finally, the adversary can determine the session key user and server using N_1 and N_2 . This scheme uses the combination values with a password and biometrics, so the adversary cannot compute the user's password. However, using the smart card attack, the adversary can compute the session key between the user $user_i$ and the server $server_j$ without the information about user's password or biometrics.

Kocher et al. and Messerges et al. pointed out that confidential information stored in all existent smart cards could be extracted by physically monitoring power consumption [3, 4]. If a user loses his smart card, all secrets in the smart card may be revealed to the adversary. Using this information, the adversary can determine the session key between the user $user_i$ and server $server_j$. To solve this problem, it is necessary to add authentication value that adversary cannot reveal using the side-channel attack. In other words, it is necessary to add the value that only legitimate user and server can compute using the secret information, which the adversary cannot know or compute.

4.3. A User Impersonation Attack. In Chuang and Chen's scheme, an adversary can be authenticated with the server using user's smart card without user's password or biometrics, so the adversary can impersonate the legitimate user. It is critical problem that the adversary can be authenticated with the server using user's smart card only. Figure 3 describes the user impersonation attack on Chuang and Chen's scheme. As described above, the adversary can illegally extract the secret values including B_i from the user's smart card by some means. And he can intercept the message (1) = $\langle \text{AUID}_i, M_1, M_2, D_i \rangle$ and acquire the AUID_i, M_1 , and D_i .

Next procedure for user impersonation attack occurs in the following steps. The adversary computes the N_1 using M_1 and $h(B_i)$. And then, he can figure out the UID_i using AUID_i and $h(N_1)$. Next, the adversary generates another random nonce N_{A1} and computes M_{A1}, AUID_{Ai} , and M_{A2} . Next, the adversary sends $\text{AUID}_{Ai}, M_{A1}, M_{A2}$, and D_i to server $server_j$. The adversary can be authenticate to server $server_j$ because he knows B_i, N_{A1} , and UID_i and the server $server_j$ cannot figure out the difference between the adversary and legitimate user. The user's password and biometric information are not used in authentication phase, so server $server_j$ authenticates the adversary without doubt. server $server_j$ does not store user's password or biometric information because Chuang and Chen's scheme is designed for anonymous user. Therefore, server cannot check the password or biometric information for authentication. To solve this problem, it is necessary to add the shared value between the user and servers. The share value can be computed by only the legitimate user using user's password and biometrics in login and authentication phase, and never be stored in the smart card.

4.4. A DoS Attack. The DoS attack is an attempt to make a machine or network resource unavailable to its intended users. Although the means to carry out motives for and targets of the DoS attack may vary, it generally consists of

(i) *Adversary* gets (steals) user's smart card.
 \Rightarrow Extracting the information of smart card.
 (Using SPA and DPA...etc) \Rightarrow Obtains B_i

(ii) *Adversary* gets M_1 and M_3 in public channel.
 $\Rightarrow N_1 = M_1 \oplus h(B_i)$
 $\Rightarrow N_2 = M_3 \oplus h^2(N_1)$
 $\Rightarrow SK_{ij} = h^2(N_1 \| N_2)$

(iii) *Adversary* can compute the session key SK_{ij} between $User_i$ and $Server_j$.

ALGORITHM 2: Smart card attack on Chuang and Chen's scheme.

(i) *Adversary* got M_{p1} and M_{p3} in previous public channel.
(ii) *Adversary* knew one of user's long-term secret: A_i
 \Rightarrow *Adversary* has A_i, M_{p1} and M_{p3}
 $\Rightarrow N_{p1} = M_{p1} \oplus h^2(A_i)$
 $\Rightarrow N_{p2} = M_{p3} \oplus h^2(N_{p1})$
 $\Rightarrow SK_{p1j} = h^2(N_{p1} \| N_{p2})$
(iii) *Adversary* can compute all of previous session key SK_{p1j} .

ALGORITHM 3: No perfect forward secrecy on Chuang and Chen's scheme.

efforts to temporarily or indefinitely interrupt or suspend services of a host connected to the networks. In Chuang and Chen's scheme, an adversary can implement the DoS attack without difficulty. Figure 4 describes DoS attack on Chuang and Chen's scheme. The adversary gets the previous message (1) from a legitimate user and sends it to the server_j. Then, the server_j executes operation (2) and sends message (3) to the user_i. The processes of operation (2) include executing the hash function 7 times, calculating the exclusive-or operation 3 times, and generating a random nonce once. The adversary can attempt to make the server or network resource unavailable if he uses a lot of intercepted authentication messages.

In Chuang and Chen's scheme, server_j does not check the freshness of authentication message from user_i. Thus, when an adversary sends the intercepted authentication messages to server_j, the server_j cannot know whether the message is current or outdated. So, server_j executes a lot of operations. To resist the DoS attack, the server_j has to check the freshness of messages using the timestamp or other means.

4.5. No Perfect Forward Secrecy. Perfect forward secrecy means that a session key derived from a set of long-term keys will not be compromised if one of the long-term keys is compromised in the future. Chuang and Chen's scheme does not achieve perfect forward secrecy. So the adversary can compute the all session key between the user_i and server_j if the adversary knows the one of long-term keys A_i in future. Algorithm 3 describes why Chuang and Chen's scheme does not achieve perfect forward secrecy. First, the adversary got M_{p1} and M_{p3} in previous communication between user_i and server_j. Next, the adversary knows one of user's long-term secrets A_i . So the adversary can calculate N_{p1} from $N_{p1} =$

$M_{p1} \oplus h^2(A_i)$ and N_{p2} from $N_{p2} = M_{p3} \oplus h^2(N_{p1})$. Finally, the adversary can compute the previous session key SK_{p1j} using N_{p1} and N_{p2} . Therefore, this scheme does not achieve perfect forward secrecy.

In Chuang and Chen's scheme, A_i is a secure shared key among RC and authenticated user_i. The RC computes A_i using UID_i and secret value x . And then, The RC sends the $h(A_i)$ to user_i within user's smart card. The $h(A_i)$ is unchanged even if user_i changes his password. So A_i is one of the long-term keys. If an adversary got the M_{p1} and M_{p3} in previous public channel and knows A_i at present, the adversary can compute the previous session key between the user_i and server_j. To solve this problem, it is needed that the adversary cannot compute the N_1 and N_2 using only A_i . By adding another secret information, it is necessary that the adversary cannot compromise the session key between user_i and server_j.

5. Our Proposed Scheme

Our proposed scheme improves Chuang and Chen's scheme in various aspects: (1) it checks the destination of messages and so it prevents the masquerade attack, (2) it withstands the smart card attack and the user impersonation attack even when the information in the smart card is disclosed, (3) it resists DoS attacks by checking the freshness of messages, and (4) it protects the security of previously-established session keys even when the adversary knows the long-term key A_i , thereby achieving perfect forward secrecy.

5.1. Countermeasures. The vulnerability of Chuang and Chen's scheme to the masquerade attack is due to the fact that

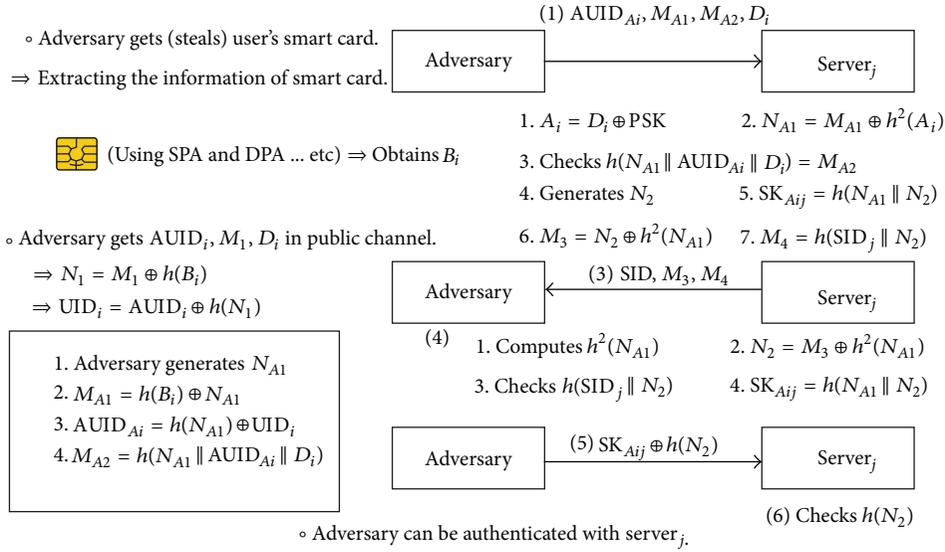


FIGURE 3: User impersonation attack on Chuang and Chen's scheme.

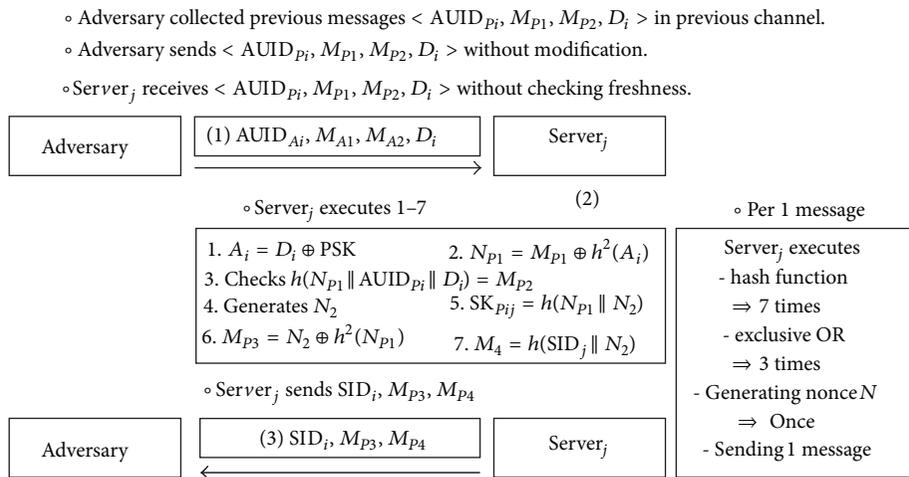


FIGURE 4: DoS attack on Chuang and Chen's scheme.

- (i) there is no way for server_j to check whether the user wants to be authenticated with it or with another server, server_k;
- (ii) user_i cannot check whether the server wants to be authenticated with him or with another user, user_j.

This design flaw allows the adversary to be authenticated with server_k using user_i's message directed to server_j. Therefore, to prevent the masquerade attack, we suggest to modify the computations of M_2 and M_4 from $M_2 = h(N_1 \parallel AUID_i \parallel D_i)$ and $M_4 = h(SID_j \parallel N_2)$ to

$$M_2 = h(N_1 \parallel AUID_i \parallel D_i \parallel SID_j),$$

$$M_4 = h(SID_j \parallel N_2 \parallel AUID_i).$$

(3)

The server ID, SID_j , and the anonymous user ID, $AUID_i$, are now included as part of the inputs of the hash function. The

inclusion of SID_j and $AUID_i$ allows server_j and user_i to confirm the destination of the messages M_2 and M_4 , respectively, and therefore effectively prevents the masquerade attack.

The DoS attack is possible because server_j performs all its operations without checking the freshness of incoming messages, and thus it can be prevented by modifying the computation of M_2 to

$$M_2 = h(N_1 \parallel AUID_i \parallel D_i \parallel SID_j \parallel T_i),$$

(4)

where T_i is the timestamp retrieved by user_i and sent to server_j. The inclusion of the timestamp T_i to the computation of M_2 enables server_j to check and confirm the freshness of the user's authentication message and prevents the DoS attack. Due to this modification, the authentication message of user_i should be also modified as follows:

$$\langle AUID_i, M_1, M_2, D_i \rangle \longrightarrow \langle AUID_i, M_1, M_2, D_i, T_i \rangle.$$

(5)

We next present a possible way of eliminating the vulnerability of Chuang and Chen's scheme to the smart card attack. Recall that this vulnerability is due to that the value B_i stored in the smart card together with M_1 and M_3 exchanged between user_{*i*} and server_{*j*} enables the adversary to compute N_1 and N_2 and thereby to derive the session key $SK_{ij} = h^2(N_1 \| N_2)$. Therefore, to prevent the smart card attack, we suggest to modify the computations of M_1 and M_3 from $M_1 = h(B_i) \oplus N_1$ and $M_3 = N_2 \oplus h^2(N_1)$ to

$$\begin{aligned} M_1 &= h(B_i) \oplus N_1 \oplus h(\text{PSK}), \\ M_3 &= N_2 \oplus h^2(N_1) \oplus h(\text{PSK}). \end{aligned} \quad (6)$$

With this modification, the adversary now cannot compute N_1 and N_2 without the hash value $h(\text{PSK})$. To make this countermeasure work, we add a new value $E_i = h(\text{PSK}) \oplus h(\text{PW}_i \oplus \text{BIO}_i)$ to user_{*i*}'s smart card so that only user_{*i*} can extract $h(\text{PSK})$ from its password and biometrics.

However, with the modifications described above, Chuang and Chen's scheme is still vulnerable to the user impersonation attack as the adversary can obtain $h(\text{PW}_i \oplus \text{BIO}_i)$ from B_i and $C_i = h(\text{PW}_i \oplus \text{BIO}_i) \oplus B_i$ which are stored in the smart card. To prevent the user impersonation attack, we modify the computation of C_i to

$$C_i = h(\text{PW}_i \oplus \text{BIO}_i) \oplus B_i \oplus h(\text{PSK}). \quad (7)$$

The adversary now cannot calculate $h(\text{PW}_i \oplus \text{BIO}_i)$ as it does not know $h(\text{PSK})$.

Finally, to provide the perfect forward secrecy in our proposed scheme, we modify the computation of D_i from $D_i = \text{PSK} \oplus A_i$ to

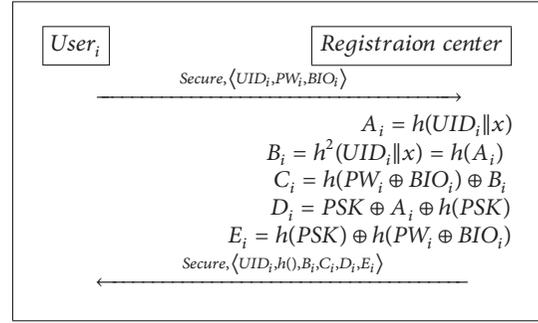
$$D_i = \text{PSK} \oplus A_i \oplus h(\text{PSK}). \quad (8)$$

With this modification, the adversary cannot derive PSK from the long-term key A_i and, thus, cannot compute N_1 , N_2 , and the previous session key $SK_{ij} = h(N_1 \| N_2)$.

The password update phase should be also modified for consistency purpose (see Section 5.5 for details). Combining all the modifications above together yields an improved authentication scheme described in the following subsections.

5.2. The Server Registration Phase. The application server sends a message for join to the RC when they want to become an authorized server. Then, the RC sends the key(PSK) to the server using secure communication. And then, the server is ready to compute $h(\text{PSK})$ for user authentication. Next, the authorized server uses the shared information like PSK and $h(\text{PSK})$ to check the user's legitimacy in authentication phase.

5.3. The User Registration Phase. The registration phase of proposed scheme is described in Algorithm 4. user_{*i*} needs to perform the user registration phase with the registration center using a secure channel. In this phase, RC sends to user_{*i*} the information about PSK and $h(\text{PSK})$. PSK is included in $D_i = \text{PSK} \oplus A_i \oplus h(\text{PSK})$. user_{*i*} can be authenticated



ALGORITHM 4: Our registration phase.

with server_{*j*} using D_i but cannot compute the PSK and A_i even if he knows the D_i and $h(\text{PSK})$. And user_{*i*} can calculate the $h(\text{PSK})$ using user's password and biometrics from $E_i = h(\text{PSK}) \oplus h(\text{PW}_i \oplus \text{BIO}_i)$. In other words, the user_{*i*} receives the hidden PSK and $h(\text{PSK})$ in D_i and E_i , respectively, included in smart card for user's login and authentication. Detailed steps are explained as follows.

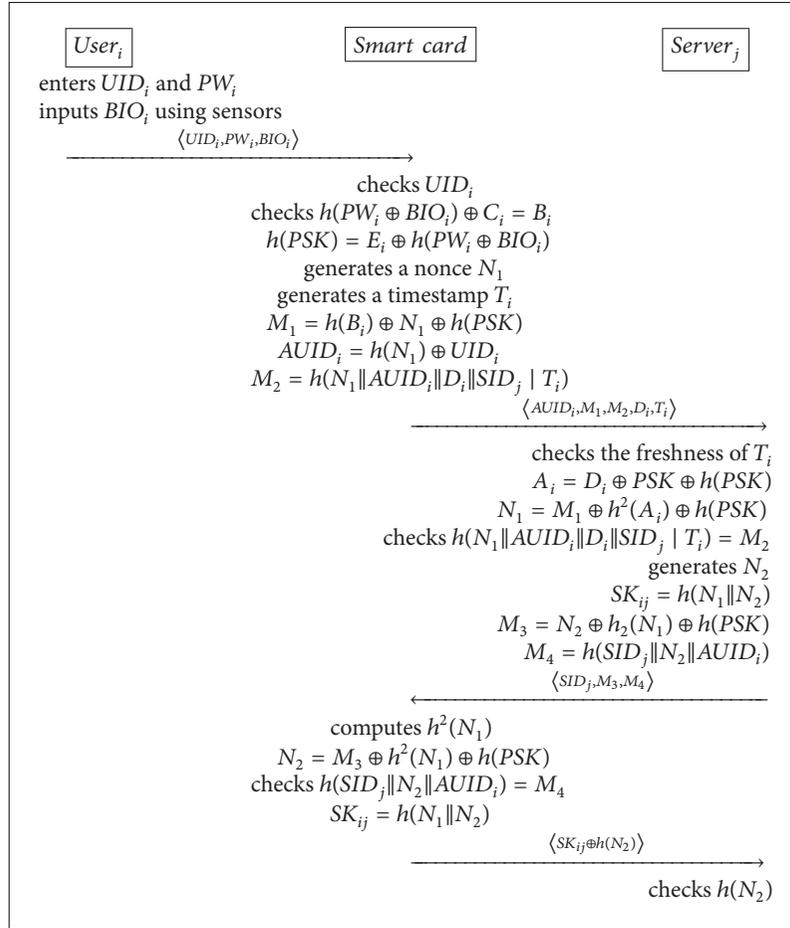
- (1) The user_{*i*} sends UID_i and $h(\text{PW}_i \oplus \text{BIO}_i)$ to the RC through a secure channel.
- (2) After receiving the user_{*i*}'s information, the RC computes the authentication parameters for the user_{*i*} as follows:

$$\begin{aligned} A_i &= h(\text{UID}_i \| x), \\ B_i &= h^2(\text{UID}_i \| x) = h(A_i), \\ C_i &= h(\text{PW}_i \oplus \text{BIO}_i) \oplus B_i, \\ D_i &= \text{PSK} \oplus A_i \oplus h(\text{PSK}), \\ E_i &= h(\text{PSK}) \oplus h(\text{PW}_i \oplus \text{BIO}_i). \end{aligned} \quad (9)$$

- (3) The RC stores these authentication parameters $\langle \text{UID}_i, h(), B_i, C_i, D_i, E_i \rangle$ in a smart card and sends the smart card to user_{*i*} via a secure channel.

The RC does not store the user's password or biometrics information. Therefore, our proposed scheme is secure against a stolen-verifier attack. The registered user cannot fake another legitimate user successfully though the user obtains these parameters $\langle \text{UID}_i, h(), B_i, C_i, D_i, E_i \rangle$. This is because that the user does not know the secret value x and PSK. The authenticated user can only compute $h(\text{PSK})$ using his password and biometrics.

5.4. The Login and Authentication Phases. The login and authentication phases for the proposed scheme are described in Algorithm 5. In the login phase, the smart card checks the legitimacy of the user. The smart card checks an error event immediately using identification, password, and biometric information. Detailed steps of the login phase are explained as follows.



ALGORITHM 5: Our login and authentication phase.

- (1) The user_i inserts his smart card into a card reader and enters his UID_i and PW_i. Then, the user_i inputs his biometric information BIO_i using the sensor.
- (2) The smart card checks the UID_i and confirms that B_i in smart card is same to h(PW_i ⊕ BIO_i) ⊕ C_i. If all information is accurate, then the smart card generates a random nonce N₁ and a timestamp T_i and computes the h(PSK) using E_i and h(PW_i ⊕ BIO_i). Next the smart card computes the following:

$$\begin{aligned}
 M_1 &= h(B_i) \oplus N_1 \oplus h(PSK), \\
 AUID_i &= h(N_1) \oplus UID_i,
 \end{aligned} \tag{10}$$

$$M_2 = h(N_1 || AUID_i || D_i || SID_j || T_i).$$

In the authentication phase, the smart card sends an authentication message to the server after the user_i finishes the login phase successfully. The proposed scheme only uses the anonymous identity AUID_i to perform the authentication phase. The detailed steps of the authentication phase are explained as follows.

- (3) The smart card sends the message ⟨AUID_i, M₁, M₂, D_i, T_i⟩ to the server_j for the user_i's authentication.

- (4) The server_j confirms the legality of the user_i and the freshness of authentication message. First, the server_j checks the freshness of T_i. If T_i is not fresh, the server_j rejects the user_i's request. The server_j uses PSK and h(PSK) to obtain A_i from the D_i. The server_j computes the value of N₁ (N₁ = M₁ ⊕ h²(A_i) ⊕ h(PSK)) and then confirms whether h(N₁ || AUID_i || D_i || SID_j || T_i) is same to M₂. If the result of M₂ is not same, the server_j terminates this session. Then, the server_j computes UID_i using h(N₁) and checks the legitimacy of UID_i. Next, the server_j generates a random nonce N₂ and computes the following:

$$SK_{ij} = h(N_1 || N_2),$$

$$M_3 = N_2 \oplus h^2(N_1) \oplus h(PSK), \tag{11}$$

$$M_4 = h(SID_j || N_2 || AUID_i).$$

- (5) The server_j sends back the authentication message ⟨SID_j, M₃, M₄⟩ to the smart card.
- (6) The smart card confirms the legality of the server_j. It computes h²(N₁) and then calculates N₂ using M₃,

$h^2(N_1)$, and $h(\text{PSK})$. Next, the smart card checks whether

$$h(\text{SID}_j \| N_2 \| \text{AUID}_i) = M_4. \quad (12)$$

Next, the smart card computes the session key SK_{ij} as $h(N_1 \| N_2)$. Finally, the smart card computes $\text{SK}_{ij} \oplus h(N_2)$.

- (7) The smart card sends the message $\langle \text{SK}_{ij} \oplus h(N_2) \rangle$ to the server j .
- (8) The server j uses the session key SK_{ij} for checking $\text{SK}_{ij} \oplus h(N_2)$, and if $h(N_2)$ is correct, the server j authenticates the user i . From now on, the server j can communicate securely with user i using the SK_{ij} .

5.5. The Password Change Phase. The password change phase for the proposed scheme is described in Algorithm 6. The proposed password change phase is executed when the user i wants to update his password. In this phase, the user i can easily change his password without any assistance from the registration center. Detailed processes are as follows.

- (1) The user i inserts his smart card into a card reader and enters both the current password PW_i and the new password PW_i^* with UID_i and BIO_i .
- (2) The smart card checks UID_i and computes $h(\text{PSK}) = E_i \oplus h(\text{PW}_i \oplus \text{BIO}_i)$ and then checks whether

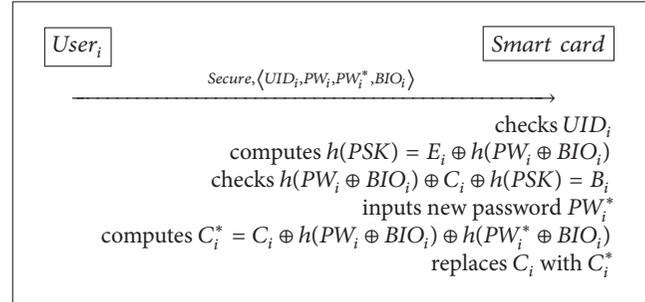
$$h(\text{PW}_i \oplus \text{BIO}_i) \oplus C_i \oplus h(\text{PSK}) = B_i. \quad (13)$$
- (3) The smart card computes $C_i^* = C_i \oplus h(\text{PW}_i \oplus \text{BIO}_i) \oplus h(\text{PW}_i^* \oplus \text{BIO}_i)$ and then replaces C_i with C_i^* .

6. Analysis of Our Scheme

An anonymous multiserver authenticated key agreement scheme has three important requirements: the security properties, the attack resistance, and the efficiency, so it needs to analyze the proposed scheme using them. In this section, we explain how the proposed scheme is satisfied with the requirements and compare the proposed scheme with other authentication schemes.

6.1. Security Properties

- (S1) *Anonymity*: in the proposed scheme, an adversary cannot compute the user's real identity UID_i without $h(N_1)$ because the real identity of user i is always converted using $\text{AUID}_i = h(N_1) \oplus \text{UID}_i$. Only legitimate server can compute and check the user's real identity, because the server has the PSK and can compute the N_1 from $N_1 = M_1 \oplus h^2(A_i) \oplus h(\text{PSK})$ using the PSK, M_1 , and A_i . Thus, only authorized server confirms the UID of user. As a result, the adversary cannot obtain the user's real identity, but legitimate user i can anonymously be authenticated with server j .



ALGORITHM 6: Our password change phase.

- (S2) *Mutual authentication*: the mutual authentication means that two parties authenticate each other. In proposed scheme, the user and server authenticated each other using N_1 , N_2 , $h(\text{PSK})$, and D_i . In the authentication phase, the server authenticates the user if the M_2 is correct as follows:

$$M_2 = h(N_1 \| \text{AUID}_i \| D_i \| \text{SID}_j \| T_i). \quad (14)$$

And the user authenticates the server using M_4 and N_2 ; it checks whether the M_4 is correct as follows:

$$M_4 = h(\text{SID}_j \| N_2 \| \text{AUID}_i). \quad (15)$$

Though an adversary intercepts the messages and wants to fake a legitimate user/server, the adversary cannot compute the accurate values, so it cannot send valid reply message to the user/server. This is because that the adversary does not know the secret key PSK, $h(\text{PSK})$ and random nonce N_1 and N_2 .

- (S3) *Session key agreement*: in the proposed scheme, the user and server can share the session key after the authentication phase. Then, they can communicate securely using the shared session key, which encrypts the communication packets. The session key is generated using $h(N_1 \| N_2)$. N_1 and N_2 change in every session, so session key is different in each session. Therefore, it is difficult for the adversary to compute the session key from the intercepted messages.
- (S4) *Perfect forward secrecy*: the proposed scheme computes the session key between the user i and server j as follows:

$$A_i = D_i \oplus \text{PSK} \oplus h(\text{PSK}),$$

$$N_1 = M_1 \oplus h^2(A_i) \oplus h(\text{PSK}),$$

$$N_2 = M_3 \oplus h^2(N_1) \oplus h(\text{PSK}),$$

$$\text{SK}_{ij} = h(N_1 \| N_2).$$

Though the user's long-term key A_i is compromised, the adversary cannot compute N_1 or N_2 because the adversary cannot calculate the $h(\text{PSK})$ and PSK,

so it cannot generate session key between user_i and server_j. Therefore, the proposed scheme achieves perfect forward secrecy. Table 1 shows the analysis on the security properties of various multisever authenticated key agreement schemes.

6.2. Attack Resistance

(A1) *Replay attack resistance*: the proposed scheme is secure against replay attack by adding the random nonce N_1 and the timestamp T_i into the message. Though an adversary intercepts the previous authentication message $\langle \text{AUID}_i, M_1, M_2, D_i, T_i \rangle$ and sends it to the server, the server can check the illegality of the request using checking N_1 and T_i as follows:

$$\text{checks } M_2 = h(N_1 \parallel \text{AUID}_i \parallel D_i \parallel \text{SID}_j \parallel T_i). \quad (17)$$

So the proposed scheme can prevent the replay attack using N_1 and T_i because the adversary cannot compute another M_2 in T_i

(A2) *Modification attack resistance*: the adversary can intercept the authentication message and attempt to modify it for illegal authentication. Using a one-way hash function, the proposed scheme checks whether authentication information is modified or not. The adversary cannot obtain the random nonce N_i or $h(\text{PSK})$, so the adversary cannot compute a legitimate authentication message. Therefore, the server and user can check whether the authentication message is modified by the adversary or not. Therefore, the proposed scheme is secure against modification attack.

(A3) *Stolen-verifier attack resistance*: the registration center and application servers do not have the user's ID/password table or the biometrics. The application server server_j authenticates the legitimate user using $h(\text{PSK})$ and D_i . Therefore, the adversary cannot obtain the authentication information about legitimate users even if the adversary gets the authority to access the database of the RC or application servers. Thus, proposed scheme is secure against stolen-verifier attack.

(A4) *Off-line guessing attack resistance*: an adversary can extract the information stored in smart card using a side-channel attack such as SPA or DPA. So the adversary can know $\text{UID}_i, B_i, C_i, D_i,$ and E_i , but he cannot figure out a user's password because $h(\text{PSK})$, $\text{PSK}, \text{BIO}_i,$ and x are unknown to the adversary. In proposed scheme, the user's password is always used with the biometrics of the user; $h(\text{PW}_i \oplus \text{BIO}_i)$, which are protected by the one-way hash function. Therefore, the adversary cannot calculate the user's password because biometric information has high entropy. Moreover, the adversary cannot figure out the biometrics because it is impossible for any two people to have the same biometrics template. Therefore, the proposed scheme is secure on off-line guessing attack.

(A5) *Forgery attack resistance*: a legitimate user cannot attempt to forge another legitimate user. The legitimate user_i can know his parameters $\langle \text{UID}_i, B_i, C_i, D_i, E_i, \text{PW}_i$ and $\text{BIO}_i \rangle$. However the user_i cannot calculate another user's real identity because another user's anonymous identity AUID_i changes in every session and is protected using a random nonce; $\text{AUID}_i = h(N_1) \oplus \text{UID}_i$. Therefore, the proposed scheme is secure against the forgery attack.

(A6) *Insider attack resistance*: in the proposed scheme, the user_i never send plain PW_i and BIO_i to the registration center RC. The user_i sends only $h(\text{PW}_i \oplus \text{BIO}_i)$, so the RC cannot obtain the user's password or biometrics. And the RC cannot compute the PW_i using $h(\text{PW}_i \oplus \text{BIO}_i)$ because the biometric information has high entropy. Moreover, $h(\text{PW}_i \oplus \text{BIO}_i)$ is sent through a secure channel and needs not store in the database of RC. So, it is difficult for even insider adversary to figure out user's PW_i and BIO_i . Therefore, the proposed scheme is secure against the insider attack.

(A7) *Masquerade attack resistance*: the masquerade attack means that an adversary is authenticated with the legitimate server using a fake or real authentication information such as the authentication messages. In Chuang and Chen's scheme, the adversary uses the authentication message between user_i and server_j to gain unauthorized access of server_k. This problem occurred because user_i and server_j cannot check the destination of authentication message. To solve this problem, the proposed scheme uses AUID_i and SID_j including M_2 as follows:

$$M_2 = h(N_1 \parallel \text{AUID}_i \parallel D_i \parallel \text{SID}_j \parallel T_i). \quad (18)$$

AUID_i includes UID_i . So the server_j can check whether user_i wants to be authenticated with server_j or not. And also M_4 include AUID_i and SID_j as follows:

$$M_4 = h(\text{SID}_j \parallel N_2 \parallel \text{AUID}_i). \quad (19)$$

So the user_i can check whether server_j wants to be authenticated with user_i or not. The adversary cannot compute M_2 and M_4 because the adversary cannot compute N_1 and N_2 . Therefore the proposed scheme is resistant to the masquerade attack.

(A8) *Smart card attack resistance*: In the proposed scheme, the smart card stores various information such as $\langle \text{UID}_i, B_i, C_i, D_i, E_i, h(\cdot) \rangle$. An adversary can obtain all information stored in user's smart card using SPA or DPA. But the adversary cannot compute the session key between user_i and server_j using M_1 and M_3

TABLE I: Comparison of security properties.

Security properties	D. Yang and B. Yang scheme [10]	Yoon and Yoo scheme [11]	Chuang and Chen scheme [13]	Our scheme
(S1) Anonymity	×	×	○	○
(S2) Mutual authentication	○	○	○	○
(S3) Session key agreement	○	○	○	○
(S4) Perfect forward secrecy	○	○	○	○

because the adversary cannot compute $h(\text{PSK})$ using obtained information as follows:

$$\begin{aligned}
 N_1 &= M_1 \oplus h(B_i) \oplus h(\text{PSK}), \\
 N_2 &= M_3 \oplus h(N_1) \oplus h(\text{PSK}), \\
 \text{SK}_{ij} &= h(N_1 \| N_2).
 \end{aligned} \tag{20}$$

Though the adversary obtains B_i and M_1 , the adversary cannot compute N_1 because of the ignorance about $h(\text{PSK})$. Thus the adversary cannot compute N_2 and SK_{ij} . Therefore the proposed scheme is secure against smart card attack.

- (A9) *User impersonation attack resistance*: in Chuang and Chen's scheme, an adversary can impersonate the legitimate user using only user's smart card because the adversary can be authenticated to the server_j using user's smart card without user's password or biometrics. However, the proposed scheme uses $h(\text{PSK})$ for protecting D_i , N_1 , N_2 , M_1 , and M_3 . For example, even though the adversary knows M_1 and B_i in $M_1 = N_1 \oplus h(B_i) \oplus h(\text{PSK})$, the adversary cannot compute N_1 without $h(\text{PSK})$, so he cannot generate the SK_{ij} . The adversary cannot know $h(\text{PSK})$ without user's password or biometric. So the adversary cannot impersonate a legal user. Therefore the proposed scheme is secure against the user impersonation attack.
- (A10) *DoS attack resistance*: the proposed scheme checks the freshness of message using timestamp, so it is useless that an adversary sends the previous message to the server. Moreover, the proposed scheme uses $M_2 = h(N_1 \| \text{AUID}_i \| D_i \| \text{SID}_j \| T_i)$ that includes timestamp T_i . The server can check the freshness and legality of M_2 because M_2 and the timestamp do not match even though the adversary sends the previous M_2 with the current timestamp. Therefore the proposed scheme is more secure against the DoS attack than Chuang and Chen's scheme.

The proposed scheme is more secure than Chuang and Chen's scheme against the masquerade attack, smart card attack, user impersonation attack, and DoS attack, and also it achieves perfect forward secrecy. Moreover, the proposed scheme is also satisfactory with regard to the anonymity, mutual authentication, session key agreement, replay attack resistance, modification attack resistance, stolen-verifier attack resistance, off-line guessing attack resistance, forgery attack resistance, and insider attack resistance.

Table 2 shows the analysis on attack resistance of various multisever authenticated key agreement schemes.

6.3. *Efficiency*. The efficiency measures include single registration, simple and secure password modification, fast error detection, and low computational cost. In performance, the proposed scheme has similar computational with Chuang and Chen's scheme. Chuang and Chen's scheme has slightly lower computational cost than the proposed scheme, but it is vulnerable to various attacks. The proposed scheme has a little higher computational cost, but it is more secure than Chuang and Chen's scheme. In other words, the proposed scheme solves security problems using similar computational cost as compared with Chuang and Chen's scheme.

- (E1) *Single registration*: in the proposed scheme, a user can be authenticated with various servers. However, the user does not need to register with every servers. To use the server's services, the user registers only one time with the registration center. The proposed scheme provides single registration so the user can anonymously use multiserver system using one registration.
- (E2) *Simple and secure password modification*: in the proposed scheme, the user can change the user's password conveniently so that it is easy for the user to change the password anytime. And, the password change phase does not need any communication with the RC. Moreover, an adversary cannot change the password even though the adversary can obtain the smart card and the user's password. This is because that the smart card can check the incorrect biometric information using PW_i , BIO_i , C_i , and B_i . The smart card verifies whether $h(\text{PW}_i \oplus \text{BIO}_i) \oplus C_i$ is the same to B_i as follows:

$$\text{checks } B_i = h(\text{PW}_i \oplus \text{BIO}_i) \oplus C_i. \tag{21}$$

- (E3) *Fast error detection*: during the login and password change phases, the smart card detects the error or mistake immediately when the adversary inputs the wrong identification, password, and biometrics information. The smart card can check the error or mistake without the RC's assistance. Therefore the proposed scheme provides fast error detection.

In Table 3, we use the following notations: “.”: that there is no computational cost in that phase, n : the number of users, m : the number of application servers, C_h : executing time of one-way hash function, C_F : executing time of the

TABLE 2: Comparison of attack resistance.

Attack resistance	D. Yang and B. Yang scheme [10]	Yoon and Yoo scheme [11]	Chuang and Chen scheme [13]	Our scheme
(A1) Replay attack	○	×	○	○
(A2) Modification attack	○	○	○	○
(A3) Stolen-verifier attack	○	○	○	○
(A4) Off-line guessing attack	○	×	○	○
(A5) Forgery attack	○	×	○	○
(A6) Insider attack	×	×	○	○
(A7) Masquerade attack	×	×	×	○
(A8) Smart card attack	○	×	×	○
(A9) User impersonation attack	○	○	×	○
(A10) DoS attack	×	×	×	○

TABLE 3: Comparison of efficiency measures.

Efficiency measures	D. Yang and B. Yang scheme [10]	Yoon and Yoo scheme [11]	Chuang and Chen scheme [13]	Our scheme
(E1) Single registration	○	○	○	○
(E2) S/S PW modification	○	○	○	○
(E3) Fast error detection	○	○	○	○
(E4) Low computational cost				
Registration user	.	C_h	C_h	C_h
Registration server
Registration RC	$n(3C_h + C_{EXP} + C_F)$	$(n + m)C_h$	$n(2C_h)$	$n(2C_h) + C_h$
Login user	$4C_h + C_{EXP} + C_F$	$2C_h + C_{ECC}$	$4C_h$	$4C_h$
Login server
Authentication user	$C_h + C_{EXP}$	$3C_h + C_{ECC}$	$5C_h$	$5C_h$
Authentication server	$3C_h + 2C_{EXP}$	$5C_h + 2C_{ECC}$	$8C_h$	$9C_h$
Authentication RC	.	$7C_h$.	.
PW change user	$3C_h + C_F$	$2C_h$	$3C_h$	$3C_h$
PW change RC

fuzzy extractor, C_{ECC} : executing time of the elliptic curve encryption or decryption operation, and C_{EXP} : executing time of the exponential operation. C_{EXP} is higher than C_{ECC} . And C_{EXP} and C_{ECC} are considerably higher than C_h . Therefore, the comparison of computational cost on above-mentioned operations is as follows:

$$C_{EXP} > C_{ECC} > C_h. \tag{22}$$

And the hash function is generally executed quickly, so it is about 1000 times faster than asymmetric encryption. In D. Yang and B. Yang’s scheme, the exponential operation is executed. In Yoon and Yoo’s scheme, the elliptic curve encryption or decryption operation is executed. But in Chuang and Chen’s scheme and proposed scheme, they use only one-way hash function. Therefore, Chuang and Chen’s scheme and proposed scheme are faster than both D. Yang and B. Yang’s scheme and Yoon and Yoo’s scheme. And our proposed scheme adds only one C_h on RC’s operation in the registration phase and also adds only one C_h on server’s operation in authentication phase in comparison with Chuang and Chen’s scheme. C_h has a little computational cost. Therefore, our proposed scheme has similar computational cost as compared with Chuang and Chen’s scheme, but Chuang and Chen’s scheme has security vulnerabilities on

the masquerade attack, smart card attack, user impersonation attack, and DoS attack as well as no perfect forward secrecy. Our proposed scheme similarly maintains the computational performance and solves the security problems of Chuang and Chen’s scheme. Therefore, the proposed scheme is the security enhanced anonymous multiserver authenticated key agreement scheme using the smart card and biometrics.

7. Conclusion

Chuang and Chen proposed an anonymous multiserver authenticated key agreement scheme. This scheme is efficient in that it only requires users to perform hash function evaluations but has various security vulnerabilities. So, we show that this scheme is vulnerable to a masquerade attack, a smart card attack, a user impersonation attack, and a DoS attack and does not achieve perfect forward secrecy. To solve the security problems of Chuang and Chen’s scheme, we propose a security enhanced anonymous multiserver authenticated key agreement scheme using smart cards and biometrics. And also, we show how the security weaknesses of Chuang and Chen’s scheme are addressed in our scheme and lastly analyze our scheme in terms of both security and efficiency.

Notations

x :	A secret value of the registration center
RC:	The registration center
UID_i :	The identification of user $_i$
SID_j :	The identification of server $_j$
$AUID_i$:	The anonymous identification of user $_i$
PW_i :	The password of user $_i$
BIO_i :	The biometrics information of user $_i$
$h()$:	A secure one-way hash function
M_i :	i th authenticator exchanged between user $_i$ and server $_j$
N_i :	A random nonce
PSK:	A secure pre-shared key among RC and servers
\parallel :	A string concatenation operation
\oplus :	A string XOR operation
\leftrightarrow :	Communication through a public channel
\leftrightarrow Secure:	Communication through a secure channel.

Conflict of Interests

The authors do not have a direct financial relation with any institution or organization mentioned in the paper that might lead to a conflict of interests for any of them.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT, and Future Planning (2014R1A1A2002775).

References

- [1] Y. P. Liao and S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards and Interfaces*, vol. 31, no. 1, pp. 24–29, 2009.
- [2] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.
- [3] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology—CRYPTO '99*, Lecture Notes in Computer Science, pp. 388–397, Springer, Berlin, Germany, 1999.
- [4] T. S. Messerges, E. A. Dabbish, and R. . . Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.
- [5] C. C. Chang and J. S. Lee, "An efficient and secure multi-server password authentication scheme using smart cards," *Computer Communications*, vol. 32, no. 4, pp. 611–618, 2009.
- [6] M. K. Khan and J. Zhang, "An efficient and practical fingerprint-based remote user authentication scheme with smart cards," in *Information Security Practice and Experience 2006*, pp. 260–268, Springer, Berlin, Germany, 2006.
- [7] W. C. Ku, S. T. Chang, and M. H. Chiang, "Further cryptanalysis of fingerprint-based remote user authentication scheme using smartcards," *Electronics Letters*, vol. 41, no. 5, pp. 240–241, 2005.
- [8] C.-T. Li and M.-S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010.
- [9] X. Li, J. W. Niu, J. Ma, W. D. Wang, and C. L. Liu, "Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 73–79, 2011.
- [10] D. Yang and B. Yang, "A biometric password-based multi-server authentication scheme with smart card," in *Proceedings of the International Conference on Computer Design and Applications (ICCCA '10)*, vol. 5, pp. 554–559, Qinhuangdao, China, June 2010.
- [11] E.-J. Yoon and K.-Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *The Journal of Supercomputing*, vol. 63, no. 1, pp. 235–255, 2013.
- [12] D. He, "Security flaws in a biometrics-based multi-server authentication with key agreement scheme," *IACR Cryptology ePrint Archive*, vol. 365, 2011.
- [13] M. C. Chuang and M. C. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1411–1418, 2014.
- [14] W. J. Tsaur, "A flexible user authentication scheme for multi-server internet services," in *Networking—ICN 2001*, pp. 174–183, Springer, Berlin, Germany, 2001.
- [15] L. Li, I. Lin, and M. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.
- [16] I. C. Lin, M. S. Hwang, and L. H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.
- [17] J. Kim, D. Lee, W. Jeon, Y. Lee, and D. Won, "Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks," *Sensors*, vol. 14, pp. 6443–6462, 2014.
- [18] J. Nam, J. Paik, and D. Won, "Security improvement on Wu and Zhu's protocol for password-authenticated group key exchange," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E94-A, no. 2, pp. 865–868, 2011.
- [19] W. Tsaur, C. Wu, and W. Lee, "An enhanced user authentication scheme for multi-server internet services," *Applied Mathematics and Computation*, vol. 170, no. 1, pp. 258–266, 2005.
- [20] T. Wu and C. Hsu, "Efficient user identification scheme with key distribution preserving anonymity for distributed computer networks," *Computers and Security*, vol. 23, no. 2, pp. 120–125, 2004.
- [21] W. S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 251–255, 2004.
- [22] J. Nam, K. K. R. Choo, J. Kim, H. K. Kang, J. Paik, and D. Won, "Password-only authenticated three-party key exchange with provable security in the standard model," *The Scientific World Journal*, vol. 2014, Article ID 825072, 11 pages, 2014.

- [23] Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 14, pp. 10081–10106, 2014.
- [24] W. Jeon, J. Kim, J. Nam, Y. Lee, and D. Won, "An enhanced secure authentication scheme with anonymity for wireless environments," *IEICE Transactions on Communications*, vol. 95, no. 7, pp. 2505–2508, 2012.

Research Article

Self-Adaptive Trust Based ABR Protocol for MANETs Using Q-Learning

Anitha Vijaya Kumar¹ and Akilandeswari Jeyapal²

¹ Dayananda Sagar College of Engineering, Bangalore 560078, India

² Sona College of Technology, Salem 636005, India

Correspondence should be addressed to Anitha Vijaya Kumar; anithasaravana79@yahoo.co.in

Received 14 March 2014; Revised 7 July 2014; Accepted 9 July 2014; Published 28 August 2014

Academic Editor: Fei Yu

Copyright © 2014 A. Vijaya Kumar and A. Jeyapal. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile ad hoc networks (MANETs) are a collection of mobile nodes with a dynamic topology. MANETs work under scalable conditions for many applications and pose different security challenges. Due to the nomadic nature of nodes, detecting misbehaviour is a complex problem. Nodes also share routing information among the neighbours in order to find the route to the destination. This requires nodes to trust each other. Thus we can state that trust is a key concept in secure routing mechanisms. A number of cryptographic protection techniques based on trust have been proposed. Q-learning is a recently used technique, to achieve adaptive trust in MANETs. In comparison to other machine learning computational intelligence techniques, Q-learning achieves optimal results. Our work focuses on computing a score using Q-learning to weigh the trust of a particular node over associativity based routing (ABR) protocol. Thus secure and stable route is calculated as a weighted average of the trust value of the nodes in the route and associativity ticks ensure the stability of the route. Simulation results show that Q-learning based trust ABR protocol improves packet delivery ratio by 27% and reduces the route selection time by 40% over ABR protocol without trust calculation.

1. Introduction

MANETs consist of a group of mobile nodes that communicate over wireless links. Since the nodes are mobile, the network topology may change rapidly and unpredictably over time. A major function in MANETs is the route discovery process, where a route from source node to destination node is discovered in order to transfer data packets. In MANETs, there are three classes of routing protocols: proactive, reactive, and hybrid [1]. The proactive protocols are table driven where each node maintains a route to every other node in the MANETs. Due to limited memory, processing power, and battery capacity, this protocol model is less preferred in MANETs. Reactive routing strategy is popular in wireless ad hoc networks because of its less overhead and on-demand nature. Dynamic source routing (DSR) [2], associativity based routing [3], and ad hoc on-demand distance vector (AODV) are the most popular examples of reactive routing protocols [4]. Hybrid protocols are the combination of the

other two and they use the proactive routing procedure to store the route when a routing is initialized and then use the reactive broadcasting to deliver the packet to its destination. Zone routing protocol (ZRP) is the popular hybrid routing protocol [5].

MANETs are deployed in very harsh or abnormal conditions and therefore the probability of network malfunctioning and uncontrolled behavior is very high. They are highly vulnerable to malicious attacks like black hole, grey hole, and Sybil attack [6]. The misbehaving node can be avoided during route discovery process by introducing trust scores of the neighboring nodes through the trust evaluation table. Usage of trust based routing protocol will improve the integrity of the received data at the receiver node [7]. The design of secure and stable routing protocols for MANETs is an active research area. The cooperation among nodes is necessary to sustain the integrity of network operations [8]. However, most of the nodes are selfish or malicious; thus secure routing is vital for protecting the routing protocols against malicious attacks [9].

We use the Q-learning technique to enable each node to learn how to adjust its route request forwarding rate according to its trust score. Since Q-learning does not always need the detailed model description in computation, it is a widely used reinforcement learning method. It is the best method used to analyze autonomous agents that self-adapt to varying external environments [10, 11]. This paper presents a self-adaptive Q-learning based trust ABR (QTABR) that calculates stable and trustworthy nodes for secure routing.

The rest of this paper is organized as follows. Section 2 gives an overview on the related work. Associativity based routing protocol is described in Section 3. Section 4 presents trust and Q-learning method. Our proposed algorithm is discussed in Section 5. Section 6 elaborates simulation environment and Section 7 discusses the results on the efficiency of our method. Finally, the last section concludes the paper and gives suggestion for further work in this area.

2. Related Work

Reactive routing protocols are more popular due to their dynamic properties. Among reactive routing protocols, associativity based routing (ABR) protocol is preferred because of its link quality. Toh [12] introduced ABR protocol as a loop and deadlock-free protocol with better routes during route reconstruction phase. Choi and Park [13] presented associativity based clustering and query stride protocol where multicast tree discovery, selection, and reconfiguration are based on the association stability. In the work presented by Kim et al. [14], route selection is based on the information obtained by GPS and presence of each mobile node within the transmission range. Their implementation reduces the excessive usage of beacon signals.

Govindan and Mohapatra [15] describe trust computations and trust dynamics such as trust propagation, aggregation, and prediction. Using these trust dynamics [16], various computations are done in distributed and centralized systems. In a multihop wireless network, optimization problems occur in multipath routing [17], link management, relay load, battery power, and secure routing. Hence there is a trade-off between the aforementioned parameters in real time scenarios. To meet these features, many research works are focussed on self-configuration networks based on reinforcement learning [18]. In Q-learning based self-configuration (QLS) management and the AODV-Q protocol [19], the QLS management architecture improves packet delivery ratio, end-to-end delay, and other quality of service performance metrics.

Reinforcement technique is also deployed in network management optimization [20–22]. Many works have been done in MANETs using Q-learning technique to enhance the performance of packets sent and received. It is mainly based on exploring the environment and learning over time to adapt to network changes. An agent is deployed in the protocol to update the changes in the routing tables.

In [16], reputation for each sensor node is determined using Q-learning which helps to find out malicious nodes. The probability of being a malicious node is calculated using

statistical and reinforcement techniques. This algorithm detects different types of malicious behaviour in a sensor node.

Reinforcement learning (RL) is a technique used to achieve adaptive routing in MANETs. Temporal difference based RL algorithm is used to achieve higher energy efficiency and less end-to-end delay [23]. In this approach energy-aware route discovery procedure over AODV reactive routing protocol is adapted in which each node adjusts its route request based on its energy table.

3. Associativity Based Routing

The ABR protocol is a reactive routing protocol with a metric called the degree of association stability. This associativity is a measure of a node's connectivity relationship with its neighbours over time and space. Each node in the network periodically transmits a beacon to its neighbours signifying its presence. ABR is a uniform routing protocol because of the fact that it provides the same importance to all nodes which participate in routing. A node caches an entry for each neighbour which records the number of beacons received. This information is stored in a variable termed "associativity tick" and is incremented each time a beacon is received. A node is said to exhibit a high state of mobility when it has low associativity ticks with its neighbours. However, if high associativity ticks are observed, the node is in the stable state and this is the ideal point to select the node to perform ad hoc routing. When a node or its neighbour moves to a new location, the node resets the associativity ticks. Associativity threshold [12] is computed as follows:

$$A_{\text{threshold}} = \left(\frac{2r}{pv} \right), \quad (1)$$

where r is the transmission range, v is the migrating speed, and p is the beaconing interval.

Association stability results when the number of beacons recorded is greater than $A_{\text{threshold}}$ (A_{th}). ABR protocol consists of three phases, namely, route discovery, route reconstruction, and route deletion. The first phase consists of broadcast query (BQ) and await-reply (REPLY) cycle. The query packet contains source ID, destination ID, intermediate ID, associativity ticks, hop count, sequence number, and a type field that identifies the type of the message. An intermediate node upon finding it as not the destination, it rebroadcasts the query packet to its neighbours. The destination upon receiving the query packet can find the best route to source by selecting the nodes with high associativity ticks and send the REPLY packet to source. Route reconstruction occurs when a link of an established route changes due to source, destination, and intermediate node migration. In ABR protocol, the selected route is long-lived due to the property of associativity. Even in the case of unexpected movements of nodes, ABR will quickly locate an alternate route. In the last phase, when the source node no longer requires the route to the destination, it sends a route deletion (RD) message and all the intermediate nodes on the way to the destination delete the route from the routing table.

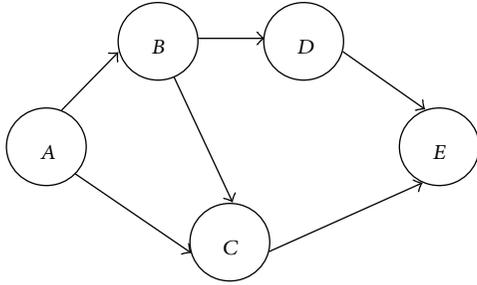


FIGURE 1: Trust computation.

4. Trust and Q-Learning

Trust has been defined as the belief or confidence or expectation on the honesty, integrity, ability, availability, and quality of service of target node's future activity/behavior [15]. Trust is calculated as a combination of direct and indirect trust [7, 24]. Direct trust is based on the acknowledgements received from the neighbour nodes during the transmission of data packets and control packets. Indirect trust is calculated by the recommendations received from the peer nodes.

The following computations of direct and indirect trust values are derived based on the scenario depicted in Figure 1. Direct trust of node A with node B is computed as follows:

$$T_{DAB}(t) = w_1 \times CSF_{AB}(t) + w_2 \times DSF_{AB}(t). \quad (2)$$

In (2) $T_{DAB}(t)$ is the trust of node B with respect to the neighbour node A, $CSF_{AB}(t)$ is the control signal forwarding ratio, $DSF_{AB}(t)$ is the data signal forwarding ratio between nodes A and B, and w_1 and w_2 are weights assigned to $CSF_{AB}(t)$ and $DSF_{AB}(t)$, respectively [24]. The weights $w_1, w_2 \geq 0$ and $w_1 + w_2 = 1$.

Indirect trust is computed as follows:

$$T_{IAB}(t) = 1 - (1 - T_{DAB}(t)) (1 - T_{ABC}(t)). \quad (3)$$

In (3) $T_{IAB}(t)$ is the indirect trust value of B with the recommendation of the neighbour node C [25]. $T_{ABC}(t)$ is the trust value sent to A by node C.

Q-learning is a form of model-free reinforcement learning. Usually a RL problem is formulated using Markov decision process (MDP) and is defined as a quintuple $(S, A, E, T, \text{ and } R)$, where S is the set of states (including finite and infinite states) of the system, A is the set of actions that agent performs and affect the system, E is the set of external events that the agent has no control over, and T is the transition function that associates each state, actions, and events. T is $S \times A \times E \times S$ and R is the reinforcement or reward function that describes the preference of certain states over other states [15, 22]. It indicates the real value obtained as feedback from environment.

In adaptive routing using reinforcement learning, routing decisions can be changed according to the network conditions [26]. Due to the infrastructureless feature of MANETs, adaptive nature is an advantage to handle the frequent changes in network topology and varying traffic load.

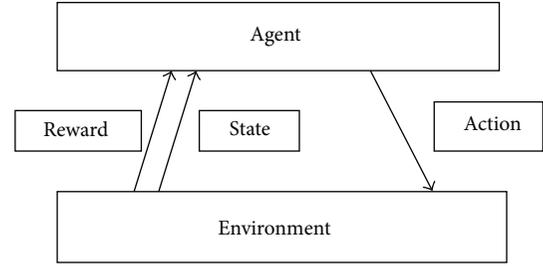


FIGURE 2: Q-learning task.

5. Proposed Q-Learning Model for Trust ABR

In Q-learning, an action is executed based on the reward received from the environment [19]. In our work, each node is formulated as an agent (Figure 2). It calculates the Q value based on the long-term and the aggregated reward. Reward is calculated from the associativity ticks and the trust value of each node with its neighbouring nodes. Generally the Q-learning score is defined as

$$Q(s, a) = (1 - \alpha) \cdot Q(s, a) + \alpha \cdot \max_a Q(s_{t+1}, a_{t+1}), \quad (4)$$

where α is the learning rate $(0 < \alpha \leq 1)$ which affects the Q-values. In this work the variable s_t represents the present state and s_{t+1} is the new state. The variable a_t represents the present action; a_{t+1} represents the action which led to s_{t+1} . In (4), $Q(s, a)$ is the Q value derived from the present state-action pair and $\max_a Q(s_{t+1}, a_{t+1})$ is the maximum Q value derived from future state-action pair. In QTABR, each agent has two Q values, namely, Q_p and Q_r . The former is the Penalty value of Q, when the trust value and associativity ticks of a particular node are less than the threshold. Q_r is the reward value of Q when trust value and associativity ticks are more than the threshold. T_v is the trust value calculated by combining the direct and indirect trust value in each agent. A timer is maintained at the source node during the route discovery process to maintain route discovery timeout (RDT).

The values for Q_r and Q_p are computed as follows:

$$Q_r = (1 - \alpha) \cdot Q(s_t, a_t)_r + \alpha \cdot Q(s_{t+1}, a_{t+1})_r, \quad (5)$$

where α is given by

$$\alpha = A_{th} + T_v, \quad (6)$$

$$T_v = T_{DAB} + T_{IAB}$$

$$Q_p = (1 - \alpha) \cdot Q(s_t, a_t)_p + \alpha \cdot Q(s_{t+1}, a_{t+1})_p.$$

Based on the values of Q_r and Q_p each node takes a decision to provide secure and long-lived route. In QTABR routing protocol, when a source node wants to send a message to destination, trust value based on Q-learning is computed to establish stable and secure route. It starts the route discovery process by sending a beacon message to its neighbours. Each agent consists of sequence of stages or episodes. Initially the RDT value is set to 30 seconds. The following procedure summarizes the activity of route discovery process.

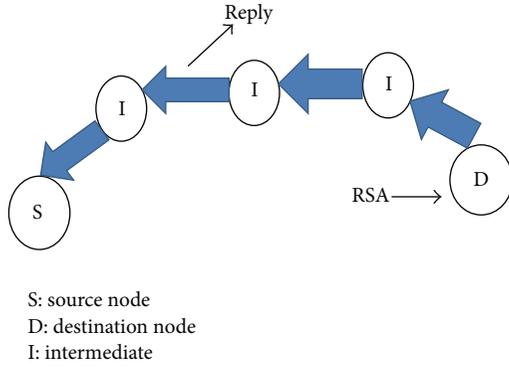


FIGURE 3: Destination node D sends a reply to source.

Step 1. During the first episode, the source node broadcasts the beacon message.

Step 2. In the next episode, the neighbouring nodes receive the beacon messages. Association stability is calculated based on the number of beacons recorded which is A_{th} .

Step 3. The agent simultaneously calculates the trust threshold value of each neighbouring node T_v as

$$T_v = T_v + T_{NN}, \quad (7)$$

where T_{NN} is the trust value of neighbouring nodes. If trust value of a neighbouring node is above threshold value (T_v), then the node is accepted as a neighbour. N_a is the total number of accepted neighbouring nodes.

Step 4. Based on A_{th} and T_v value of each node, Q_r and Q_p are calculated in each neighbouring node and broadcast query (BQ) control packet is propagated.

Step 5. After receiving successful BQ control packet at destination, increase the reward value of that route from the source.

Step 6. If RDT elapses, increase Q_p value of that route. The agent at the source checks Q_p value against threshold and if the difference is positive, it decides to discard the route. Link delay is automatically eliminated due to the incorporation of RDT.

Step 7. At the destination node, route selection algorithm (RSA) is used. RSA computes the best route based on metrics such as association stability, hop count, relay load, signal strength, power, and link delay. Trust value is incorporated into the given metrics.

Step 8. Q-learning agent reconfigures the routing parameters like beacon signal intervals, RDT, and RSA.

In the BQ reply control packet of our proposed algorithm, the route quality is added with the trust value as in Figure 4. The aggregated trust value of each node is calculated at the destination. In Table 1, different threshold values for different

Aggregate degree of association stability	Aggregate trust value	Route length	Aggregate relaying load
---	-----------------------	--------------	-------------------------

FIGURE 4: Format of QTABR-BQ reply control packet.

TABLE 1: Trust levels of nodes.

Level	Trust value	Meaning
1	{0, 0.29}	Malicious node
2	{0.3, 0.59}	Known node
3	{0.6, 0.79}	Companion node
4	{0.8, 1}	Trustworthy node

types of neighbors are shown. If T_v value is less than 0.3, it will be regarded as a malicious node and it would be blacklisted.

Figure 3 depicts a destination node sending a reply packet back over the selected route after computing aggregated trust. At the destination, route selection algorithm (RSA) decides the route with higher trust value and associativity threshold.

Figure 4 depicts the format of broadcast query (BQ) reply control packet in QTABR which includes the aggregated trust value. In BQ reply control packet, route length specifies the length of the route and aggregated relaying load is the total relay load in the specified path. Aggregate degree of association stability is an important parameter, which determines the stability of nodes in the corresponding path. Figure 5 illustrates the detailed process of route discovery phase in QTABR.

5.1. Time Complexity. Our algorithm works in two phases. In phase one associativity ticks and trust value from the beacon signals received from the neighbors using Q-learning are computed. The worst case complexity for each node using Q-learning for this phase is given by $O(e \cdot n)$, where “ e ” is the number of steps required to reach the secure and stable route and “ n ” is the number of neighboring nodes. During the second phase, each node gets b_i beacon packets. Therefore the time complexity of processing beacon packets is $O((e \cdot n) + \sum_{i=1}^K b_i)$, where “ K ” is the number of nodes in the path.

6. Simulation Environment

We have implemented the proposed Q-learning on the top of ABR routing protocol using NS-2 simulator and the simulation summary is given in Table 2. The simulation is carried out in the area of 1000 m*1000 m with 50 mobile nodes. In each scenario, the nodes move in a random direction using random way point model with a speed randomly chosen within the range of 0–20 m/s. We assumed a presence of 0–40% of malicious nodes in the network. ABR and QTABR results are obtained as the average of 25 runs for each protocol. Maximum pause time is considered for each run. During simulation, QTABR reconfigures RDT between 4 seconds and 12 seconds and beacon messages interval is between 1 second and 12 seconds.

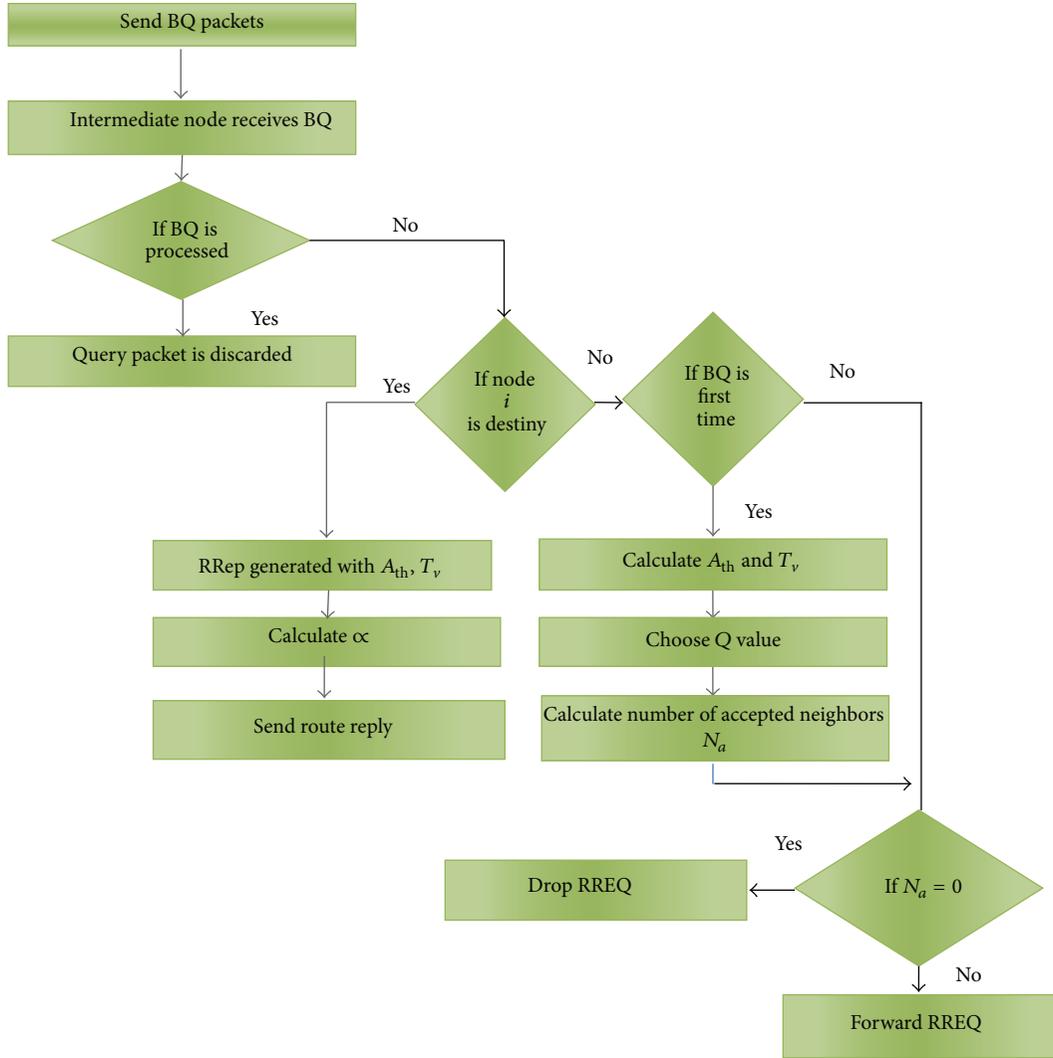


FIGURE 5: Flow diagram of route discovery phase in QTABR.

We conduct two sets of experiments to evaluate the performance of our approach. In the first set of experiments, we examined the longevity of the route. In the second set of experiments, malicious node detection and route selection time are examined. The accuracy of malicious detection by trust calculation and overhead of communication in trust computation is considered. Malicious node detection rate and transmission overhead are taken as metrics. Therefore, we distribute malicious nodes randomly and evaluate the performance of our proposed trust based Q-learning. In the former study, packet delivery ratio and packet dropping ratio are measured.

6.1. *Packet Delivery Ratio (PDR)*. This ratio gives an indication about network throughput. It is the ratio between the number of packets received (NPR) successfully and number of packets sent (NPS) and as in (8) PDR is directly proportional to long-lived route. Consider

$$PDR = \frac{NPR}{NPS}. \tag{8}$$

TABLE 2: Summary of NS-2 simulation parameters.

Simulation parameters	Values
Simulation area	1000 m * 1000 m
Number of nodes	Mobile nodes (MN) = 100
Malicious nodes	0–20 nodes
Mobility model	Random waypoint
Speed	Uniform (0–20) m/s
Pause time	0, 60, 120, 180, 240 in s
Transmission range	350 m
Wireless interface	IEEE 802.11b
Traffic flow	CBR
Transmission power	0.6 W
Reception power	0.3 W
Learning rate α	0.6
Simulation duration	10 min (for each run)

6.2. *Packet Dropping Ratio (PD)*. This is the percentage of packets dropped during data transmission. It is the ratio

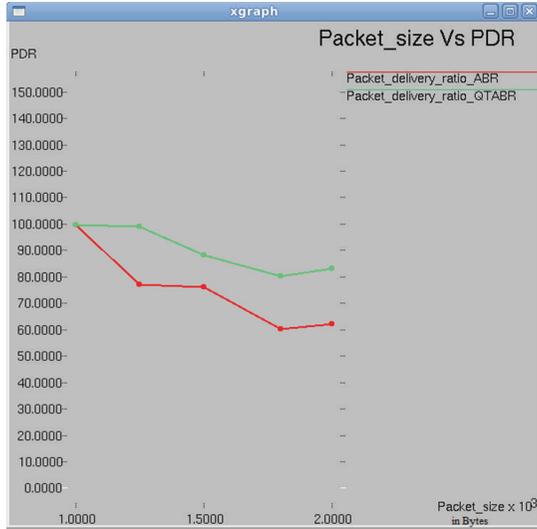


FIGURE 6: Packet size versus PDR.

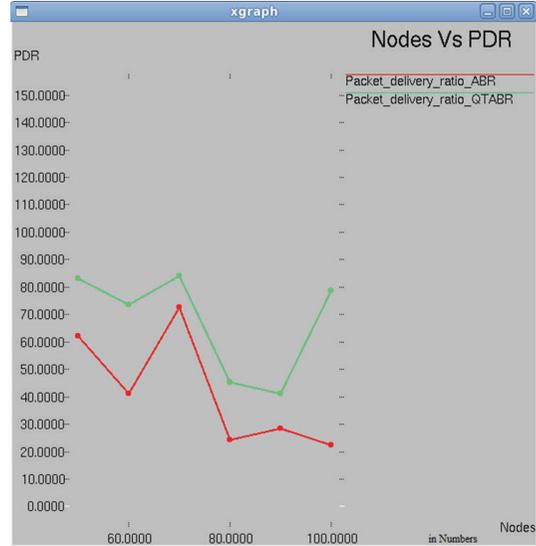


FIGURE 7: Nodes versus PDR.

between the number of packets dropped (NPD) and number of packets sent (NPS) and as in (9) packet dropping ratio is inversely proportional to longevity of the route. Consider

$$PD = \frac{NPD}{NPS} \tag{9}$$

7. Results and Discussions

In the first simulation setup, we measured the packet delivery ratio and packet dropping ratio that directly reflects the longevity of the route. The impact of varying packet size, number of nodes, pause time, and speed on packet delivery ratio is analyzed. For simplicity we mention the algorithm without learning as ABR and the algorithm with learning as QTABR.

7.1. Impact of Packet Size on Packet Delivery Ratio. In Figure 6, the results are plotted between packet size and packet delivery ratio. The experimental results of ABR show that, as the packet size increases to 1400 bytes, the PDR decreases to 74%. With QTABR, the PDR is 90%, and then it decreases with the increase in packet size. When the packet size increases to 2000 bytes the PDR gradually increases to 85%. Based on the observations, the use of larger packet size can increase the performance of ad hoc networks under QTABR.

7.2. Impact of Number of Nodes on Packet Delivery Ratio. Results in Figure 7 show the plot between number of nodes and packet delivery ratio. As the number of nodes increases to 70, PDR value increases steeply to 75% and 85%, using ABR and QTABR, respectively. With the experiments done with ABR, as the number of nodes increases to 80, PDR value drops to 22% and with QTABR, it is 45%. As the number of nodes increases to 100, the PDR value increases to 80% in QTABR's implementation which is 4 times greater than ABR's implementation.

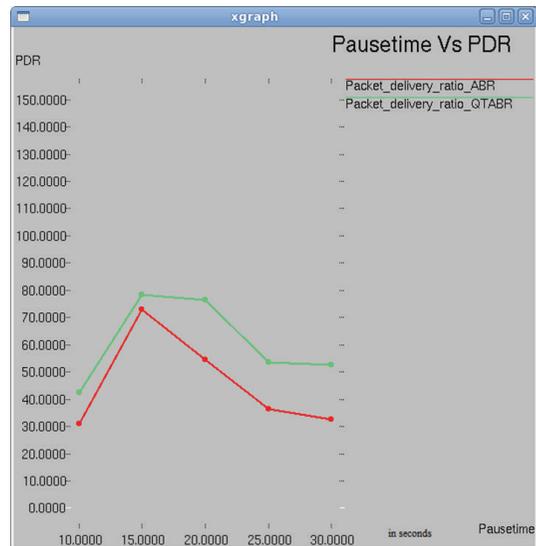


FIGURE 8: Pause time versus PDR.

7.3. Impact of Pause Time versus PDR. Results in Figure 8 depict the plot between pause time and packet delivery ratio. As the pause time increases to 25 ms the PDR drops to 55% and 30% with ABR and QTABR, respectively, but it increases gradually with increase in pause time in QTABR.

7.4. Impact of Speed on Packet Delivery Ratio. Figure 9 shows the plot of a graph between the speed time and packet delivery ratio. As the speed increases above 2 m/sec the PDR drops in both of the algorithm implementations. PDR increases to 100% under QTABR when speed is 4 m/sec.

7.5. Impact of Packet Size on Packet Dropping Ratio. Figure 10 depicts the graph between packet size and packet dropping



FIGURE 9: Speed versus PDR.

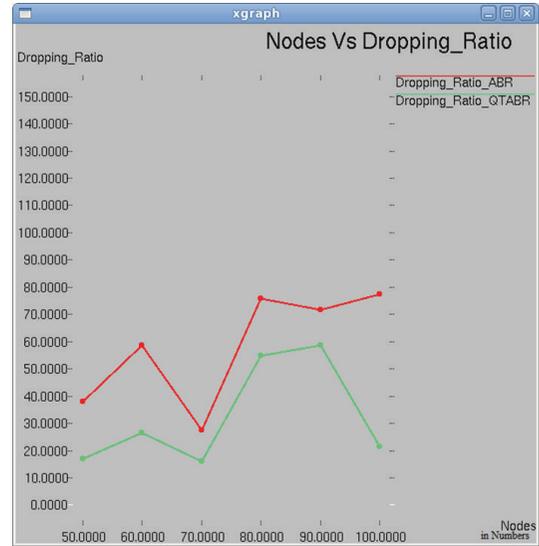


FIGURE 11: Number of nodes versus packet dropping ratio.

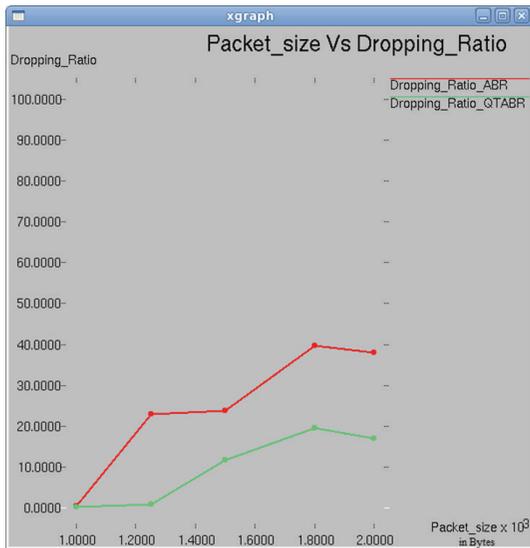


FIGURE 10: Packet size versus packet dropping ratio.

ratio. The graph clearly demonstrates the fact of reduced dropping ratio when QTABR is experimented against ABR.

7.6. Impact of Number of Nodes on Packet Dropping Ratio. The graph in Figure 11 illustrates the relationship between number of nodes and packet dropping ratio. In both of the algorithms the dropping ratio decreases steeply as the number of nodes reaches 70. When the number of nodes increases to 80, packet dropping ratio is 10% and 25% in ABR and QTABR, respectively. As the number of nodes increases to 100, the packet dropping ratio is 80% under ABR which is 4 times greater than QTABR's implementation.

In the second simulation setup, we increased the number of malicious nodes in the network from five to twenty. Figure 12 shows the percentage of detected malicious nodes in the network using QTABR. A node will be recognized as

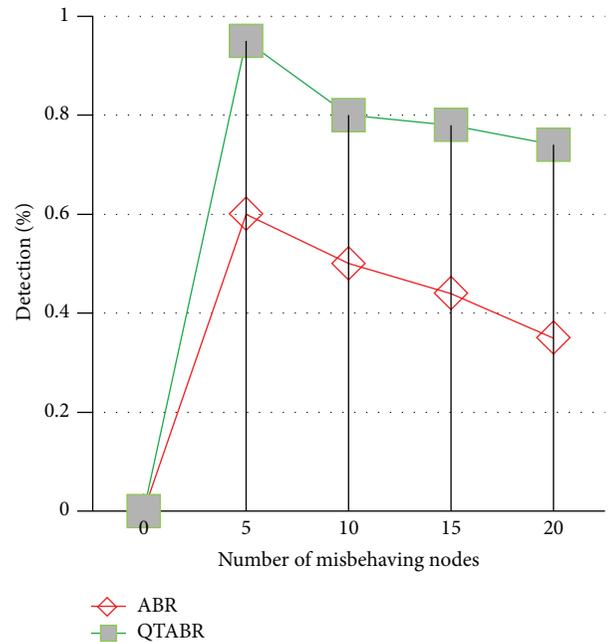


FIGURE 12: Percentage of detection of malicious nodes.

malicious node if its trust value (T_i) is less than 0.3. When the number of malicious nodes is small, most of the malicious nodes can be recognized using QTABR.

In Figure 12 QTABR routing protocol can explore more malicious nodes and can avoid that route. Thus, selection of secure route is almost 67% to 80% under the proposed condition which is always 10% to 15% more than the existing ABR protocol. Whenever there is a change in route due to link failure, the intermediate nodes should share this information. Rarely, intermediate nodes can also misbehave, and thus the percentage of detection decreases with the increase in the number of malicious nodes. The percentage of detection

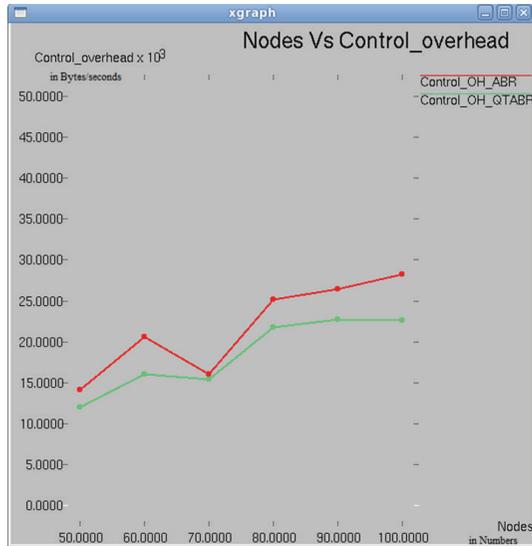


FIGURE 13: Number of nodes versus control overhead.

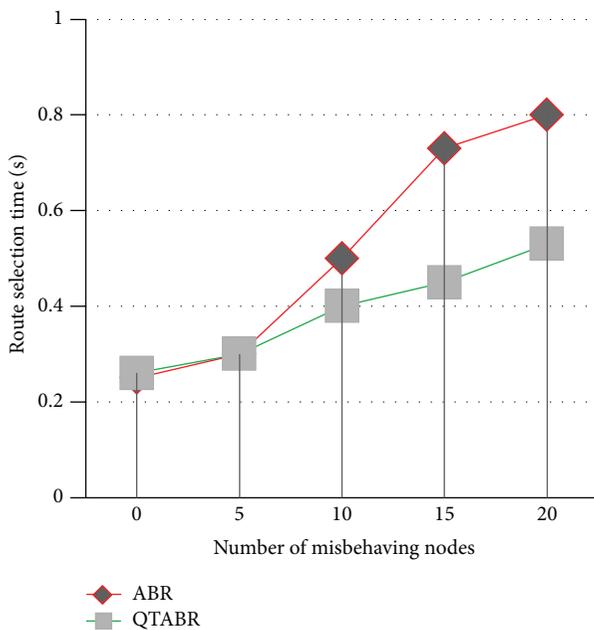


FIGURE 14: Route selection time.

decreases more in existing ABR as the number of malicious nodes increases.

In Figure 13, we compared the control overhead of traditional ABR and QTABR. Routing overhead is more, when there is less number of malicious nodes in the network; as the number of malicious nodes increases in a network, the proposed algorithm results in less control overhead. Experimental results suggest that when number of nodes is 70, control overhead is less in both of the routing protocols. It is the optimal value of nodes in the network that produces good output in terms of packet delivery ratio, dropping ratio, and control overhead.

We have evaluated route selection time, by random distribution of malicious nodes from five to twenty. Figure 14 shows that as the number of malicious nodes increases the route selection time in the proposed QTABR is less when compared to the existing ABR.

8. Conclusion and Future Work

We have proposed a Q-learning based trust routing scheme. The proposed scheme is promising as it increases packet delivery ratio and reduces route selection time. We have analyzed the performance of the proposed scheme for various numbers of misbehaving nodes. Security in the routing phase is enhanced by discovering a trustworthy route using QTABR routing protocol. This protocol has proven to offer several advantages. The foremost advantage is the performance of the protocol despite the presence of misbehaving nodes. Secondly, this is applicable to large heterogeneous networks, where the characteristics of the mobile nodes and application demands are different. Thirdly, since the agents are flexible in nature, they can be adapted to any changes with the minimal overhead trade-off. As revealed in this paper, the routing problem in MANETs requires the optimization of many conflicting objectives. This work can be further extended by applying Q-learning approach in all stages of ABR to improve the end-to-end routing.

Conflict of Interests

The authors do not have a direct financial relation with the commercial identity mentioned in the paper which might lead to a conflict of interests for any of them.

References

- [1] P. Papadimitratos and Z. J. Haas, "Secure data communication in mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 343–356, 2006.
- [2] D. B. Johnson, D. A. Maltz, Y. Chu, and J. G. Jetcheva, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," *InternetDraft*, February 2002.
- [3] F. A. M. Masoud, S. A. Shaar, A. Murad, and G. Kanaan, "Enhanced routing re-construction method for the associativity based routing protocol for mobile AD Hoc network (MANET)," *The American Journal of Computer Science*, vol. 2, no. 12, pp. 853–858, 2006.
- [4] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to byzantine failures," in *Proceedings of the ACM Workshop on Wireless Security*, pp. 21–30, September 2002.
- [5] Z. J. Haas and M. R. Pearlman, "The performance of query control schemes for the zone routing protocol," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 427–438, 2001.
- [6] A. V. Kumar, A. Jeyapal, and R. Gowda, "FPGA implementation of enhanced ABR protocol with auto defense towards malicious node in MANETs," *Internal Security*, pp. 137–154, 2012.
- [7] J.-H. Cho, A. Swami, and I.-R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 4, pp. 562–583, 2011.

- [8] Y. Cheng, C. Huang, and W. Shi, "Trusted dynamic source routing protocol," in *Proceedings of the IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '07)*, pp. 1632–1636, Athens, Greece, September 2007.
- [9] M. G. Zapata and N. Asokan, "Secure Ad hoc on-demand distance vector routing," *ACM Mobile Computing and Communications Review*, vol. 3, no. 6, pp. 106–107, July 2002.
- [10] Q. Zeng and Z. Yang, "An approach integrating simulation and Q-learning algorithm for operation scheduling in container terminals," *Journal of the Eastern Asia Society for Transportation Studies*, vol. 8, pp. 2225–2239, 2010.
- [11] D. Kim and S. Park, "Reinforcement learning-based dynamic adaptation planning method for architecture-based self-managed software," in *Proceedings of the ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '09)*, pp. 76–85, Vancouver, Canada, May 2009.
- [12] C. K. Toh, "Associativity-based routing for Ad-Hoc mobile networks," *Wireless Personal Communications*, vol. 4, no. 2, pp. 103–139, 1997.
- [13] Y. Choi and D. Park, "Associativity based clustering and query stride for on-demand routing protocols in ad hoc networks," *Journal of Communications and Networks*, vol. 4, no. 1, pp. 4–13, 2002.
- [14] D. Kim, Y. Choi, and C.-K. Toh, "Location-aware long-lived route selection in wireless ad hoc network," in *Proceedings of the 52nd Vehicular Technology Conference on IEEE-VTS Fall (VTC '00)*, vol. 4, pp. 1914–1919, 2000.
- [15] K. Govindan and P. Mohapatra, "Trust computations and trust dynamics in mobile adhoc networks: a survey," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 2, pp. 279–298, 2012.
- [16] O. Mistry, A. Gürsel, and S. Sen, "Comparing trust mechanisms for monitoring aggregator nodes in sensor networks," in *Proceedings of the 8th International Conference on Autonomous Agents and multiagent Systems*, 2009.
- [17] T. Hanriotakis, S. Tragoudas, and C. Kalapodas, "Security enhancement through multiple path transmission in ad hoc networks," in *Proceedings of the IEEE International Conference on Communications*, pp. 4187–4191, June 2004.
- [18] K. A. Yau, P. Komisarczuk, and P. D. Teal, "Reinforcement learning for context awareness and intelligence in wireless networks: review, new features and open issues," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 253–267, 2012.
- [19] D. Kumar, D. B. Ojha, and A. Kumar, "Securing MANETs by Q routing protocol," *International Journal of Engineering Research and Applications*, vol. 2, no. 6, pp. 884–889, 2012.
- [20] J. Dowling, E. Curran, R. Cunningham, and V. Cahill, "Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 35, no. 3, pp. 360–372, 2005.
- [21] K. Doya, K. Samejima, K. Katagiri, and M. Kawato, "Multiple model-based reinforcement learning," *Neural Computation*, vol. 14, no. 6, pp. 1347–1369, 2002.
- [22] A. Montresor, H. Meling, and O. Babaoglu, "Toward self-organizing, self-repairing, and resilient distributed systems," in *Future Directions in Distributed Computing*, vol. 2584 of *Lecture Notes in Computer Science*, pp. 119–123, 2003.
- [23] S. Chettibi and S. Chikhi, "Adaptive maximum-lifetime routing in mobile ad-hoc networks using temporal difference reinforcement learning," *Evolving Systems*, vol. 5, no. 2, pp. 89–108, 2014.
- [24] P. Rama Kishore, "Multiple trust in trust-based on-demand routing in mobile ad hoc networks," *International Journal on Advanced Computer Engineering and Communication Technology*, vol. 1, no. 2, 2012.
- [25] P. Gera, K. Garg, and M. Misra, "Trust-based multi-path routing for enhancing data security in MANETs," *International Journal of Network Security*, vol. 16, no. 2, pp. 102–111, 2014.
- [26] D. Marconett, M. Lee, X. Ye, R. Vemuri, and S. J. B. Yoo, "Self-adapting protocol tuning for multi-hop wireless networks using Q-learning," *International Journal of Network Management*, vol. 23, no. 2, pp. 119–136, 2013.

Research Article

Spatiotemporal Access Model Based on Reputation for the Sensing Layer of the IoT

Yunchuan Guo,^{1,2} Lihua Yin,^{1,2} Chao Li,^{1,2} and Junyan Qian³

¹ Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

² Beijing Key Laboratory of IOT Information Security, Beijing 100093, China

³ Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

Correspondence should be addressed to Lihua Yin; yinlihua@software.ict.ac.cn

Received 13 March 2014; Accepted 29 April 2014; Published 6 August 2014

Academic Editor: Fei Yu

Copyright © 2014 Yunchuan Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Access control is a key technology in providing security in the Internet of Things (IoT). The mainstream security approach proposed for the sensing layer of the IoT concentrates only on authentication while ignoring the more general models. Unreliable communications and resource constraints make the traditional access control techniques barely meet the requirements of the sensing layer of the IoT. In this paper, we propose a model that combines space and time with reputation to control access to the information within the sensing layer of the IoT. This model is called spatiotemporal access control based on reputation (STRAC). STRAC uses a lattice-based approach to decrease the size of policy bases. To solve the problem caused by unreliable communications, we propose both nondeterministic authorizations and stochastic authorizations. To more precisely manage the reputation of nodes, we propose two new mechanisms to update the reputation of nodes. These new approaches are the authority-based update mechanism (AUM) and the election-based update mechanism (EUM). We show how the model checker UPPAAL can be used to analyze the spatiotemporal access control model of an application. Finally, we also implement a prototype system to demonstrate the efficiency of our model.

1. Introduction

As a dynamic global ubiquitous network, the Internet of Things (IoT) links physical and virtual objects by integrating sensors, smart terminals, and global positioning systems (GPSs). Authoritative institutes predicate that the IoT will create hundreds of billions of dollars in savings and productivity gains for businesses, governments, and house-holds: Cisco believes that the IoT will create a US\$14.4 trillion business opportunity in 2020 (http://www.eetimes.com/document.asp?doc_id=1263115) and Groupe Speciale Mobile Association (GSMA) predicts that, in 2020, the connected life (one part of the IoT) will bring a US\$4.5-trillion global impact on people and businesses (<http://www.gsma.com/newsroom/gsma-announces-the-business-impact-of-connected-devices-could-be-worth-us4-5-trillion-in-2020/>).

Along with the increasingly rapid development of the IoT, security issues have also become increasingly serious,

especially when industrial controllers are either directly or indirectly connected to IoT. A typical example of this type of security compromise is the worm Stuxnet. Known as the first cyber-warfare weapon, Stuxnet was used to attack the Natanz uranium enrichment facility in Iran and is believed to have caused its production to drop by 15% in 2009 [1]. Obviously, security problems will cause a serious impact to the IoT.

As one of the key technologies involved in providing security, access control—determining *who* is allowed access, *when* access is permitted, and *where* access takes place—has been widely studied [2]. Access control models, which have been widely used, include role-based access control (RBAC) and usage control (UCON) [3, 4], Internet content control (ICCON) [5], Attribute-based access control [6], and user-driven access control [7].

Although these models succeed in the traditional Internet and operating systems, the IoT has raised several new and challenging issues surrounding the use of digital resources

and its following critical characteristics make the above models not efficient any more. (1) Uncontrollable environments: sensors could be deployed in unattended environments, where physical nodes are more likely lost and false messages are more easily injected and transmitted. (2) Sensor-node resource constraints: computing and storage resources for sensor nodes are usually very limited, thereby severely constraining their ability to store and process the sensed data. Therefore high-weight access control models for the Internet should be revised for the sensing layer of the IoT. (3) Unreliable communications: the wireless communication adopted by the sensor nodes is often unreliable and unstable; therefore nodes may not receive the authorization in time. As a result, security in the IoT becomes more severe.

To minimize these threats, we proposed spatiotemporal access control based on reputation (STRAC), which considers time, location, and reputation as key elements in deciding whether access is granted or not. STRAC uses a lattice structure to decrease the storage complexity of policy bases. To reduce the risk caused by unreliable communications, we proposed nondeterministic authorizations (i.e., pessimistic, optimistic, and trade-off authorizations) and stochastic authorization. We demonstrate that pessimistic and trade-off authorizations are secure and that optimistic and stochastic authorizations can improve the QoS. In order to correctly update the reputation of nodes, we propose two novel policies (authority-based updates and election-based updates), based on the “group” characteristics of the sensing layer, and we prove that our proposed policies are secure. Our experiments show the efficiency of our model.

2. Related Work

Research about access control for the sensing layer can be divided into two general categories: access control algorithms (ACAs) and access control models (ACMs). ACAs mainly focus on new node addition. New node addition algorithms prevent malicious nodes from joining the sensor network. For example, [8] uses the self-certified elliptic curve Diffie-Hellman protocol to establish a pairwise key between new sensor nodes and the controller node, which launches a two-way authentication with the new nodes. However, in this scheme, all nodes share a network-wide key. Once one node is compromised, the secret key for all nodes must be updated, thereby causing huge losses. In order to solve this problem, [9, 10] proposes a new dynamic access control protocol, which uses hash functions to reduce computations and communications between two nodes.

In ACMs, much effort is spent on extending RBAC for pervasive computing. Reference [11] proposes a dynamic role-based access control (DRBAC) model, which provides context aware access control by dynamically adjusting role assignments and permission assignments based on context information. However, important features of the IoT (i.e., location and time) are not considered. In order to make RBAC more pervasive, many researchers extend RBAC by introducing time and location [12–15], where [12, 14] imposes spatiotemporal constraints on user-role assignments and

permission assignments, and [15] introduces the concept of spatiotemporal zones and allows spatiotemporal constraints to be specified with prerequisite constraints. In addition, [16] adopts RBAC-based (role-based access control) authorization method using the thing’s particular role(s) and application(s) in the associated IoT network. Reference [17] designs a capability-based access control delegation model for the federated IoT network. Reference [18] focuses on a minimal use of computation, energy, and storage resources at wireless sensors and proposes a novel access control solution for wireless network services in Internet of Things scenarios.

Although RBAC is often extended for pervasive computing, these extensions cannot be widely adopted for the sensing layer of the IoT because of the PSPACE-completeness [19] of RBAC.

Other spatiotemporal models that are not based on RBAC are also proposed. Reference [20] uses composition algebra to regulate access to patient data and balances the rigorous nature of traditional access control systems with the “delivery of care comes first” principle. Recently, reputation has been incorporated into models of access control for cyber-physical systems as in [21, 22]; however, these particular models do not deal with the loss of nodes.

Our work differs from the above solutions in several ways. First, we consider reputation, rather than roles, as a fundamental factor of access control for the sensing layer of the IoT, because the behavior of selfish nodes can be directly modeled by reputation but not easily modeled by roles. Such a change is nontrivial. If a node becomes selfish, then we are only required to assign a lower reputation to it. Therefore, reputation is more suitable than roles in controlling access to the sensing layer.

Second, the existing access control models do not efficiently handle the problem caused by unreliable communications. We propose nondeterministic authorizations and stochastic authorizations to solve this problem. Our method reduces the security risks of security-critical systems when failing to receive the key authorization instructions.

Finally, the existing models for the IoT do not consider the group characteristics of the sensing layer. In our work, node’s reputation is cooperatively updated based on the group characteristics, thereby simplifying the reputation-update process.

3. Formalizing Time, Space, and Reputation

In order to construct a spatiotemporal access model based on reputation, we first formally define time, space, and reputation.

3.1. Reputation Description. Due to the limitations of storage and the computing resources, some nodes do not cooperate with others and demonstrate *selfishness*. In order to obtain more benefits, some nodes may attack others and demonstrate *misbehavior*. Because reputation (the opinion of one entity regarding another) can reflect both selfishness and misbehavior in interactions, it is adopted in modeling the behavior of nodes in our study.

Generally, from the aspect of reputation obtainment, reputation includes direct reputation (DR) and indirect reputation (IR), where DR and IR, respectively, refer reputation estimated by estimators based on their first-hand and second-hand experiences. From the aspect of goal, individual reputation should be distinguished with group reputation. Reference [23] surveys notions of reputation. In this paper, we only focus on individual and direct reputation, as follows.

We define reputation REP to have different ratings, and thus it can be denoted by a finite set; that is, $REP = \{rep_1, \dots, rep_n\}$, where rep_i is a reputation rating ($1 \leq i \leq n$). Given any rep_x and rep_y in REP , they are mutually comparable, that is, $rep_x \leq rep_y$ or $rep_y \leq rep_x$. Thus, REP is a total order set. For a given node, its reputation is formed and \leq updated through direct observations of its behavior and through feedback provided by other nodes. In this paper, we concentrate on general access models and do not discuss the methods of computing reputation in detail.

3.2. Time Description. In order to describe operations that can only be executed within a given time period, the notion of a calendar is adopted [24, 25]. A calendar consists of a countable set of contiguous intervals, for example, years, months, and days. Because two calendars can have different granularities, a subcalendar relationship can be established among them. That is, given two calendars c_1 and c_2 , c_1 is a subcalendar of (written as $c_1 \sqsubseteq c_2$), if and only if there exists a natural number, such that $c_2 = i \times c_1$. For example, days are a representative subcalendar of months. Obviously, \sqsubseteq is a partial order relation. A calendar base CB represents a set of calendars and generally changes with different contexts. For example, if a school curriculum is comprised of years, semesters, and weeks, then its CB is {years, semesters, weeks}.

Definition 1 (calendar time). Given $CB = \{c_1, \dots, c_n\}$, calendar time ct is defined as $ct = \sum_{i=1}^n n_i \cdot c_i$, where, $n_i \in N$ and for all $2 \leq i \leq n$, one has $c_i \sqsubseteq c_{i-1}$.

Let CT be a set of calendar times. Generally, any two calendar times are always comparable. That is, for any ct_x and ct_y in CT , one can have $ct_x \leq ct_y$ or $ct_y \leq ct_x$ (\leq is the total order relation). In the IoT, different types of time constraints exist, such as the earliest access time (*eat*), the latest access time (*lft*), the earliest finish time, and the latest finish time. In our study, *eat* and *lft* are adopted.

Definition 2 (time constraints). Time constraint $TC \subseteq CT \times CT$ is a set of two-dimensional vectors of calendar times, where the first dimension and the second dimension represent *eat* and *lft*, respectively. Time constraints must satisfy the following condition: for any $(ct_1, ct_2) \in TC$, $ct_1 \leq ct_2$.

Example 3. Given $TC = \{(ct_{11}, ct_{12}), (ct_{21}, ct_{22})\}$, and an event satisfies TC , if access time (from start time to end time) of the event falls entirely within the time range from ct_{11} to ct_{12} , or within the time range from ct_{21} to ct_{22} .

A time constraint with overlapping ranges $\{(1, 3), (2, 4)\}$ can be reduced to $\{(1, 4)\}$, based on the following definition.

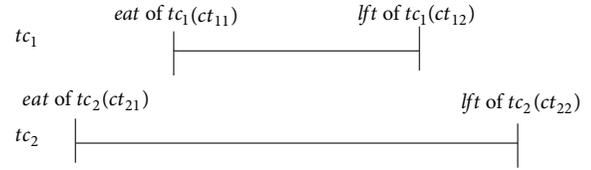


FIGURE 1: Time constraints.

Definition 4 (simplest time constraints). A time constraint TC is the simplest, if for any (ct_{11}, ct_{12}) and $(ct_{21}, ct_{22}) \in TC$, $\min\{ct_{12}, ct_{22}\} < \max\{ct_{11}, ct_{21}\}$.

Henceforth, one assumes that time constraints are always the simplest. Given two time constraints tc_1 and tc_2 shown in Figure 1, where (1) ct_{11} —*eat* of tc_1 —is greater than or equal to that of tc_2 and (2) ct_{12} —*lft* of tc_1 —is less than or equal to that of tc_2 . If an event satisfies tc_1 , then it will satisfy tc_2 ; this means that tc_1 is stricter than tc_2 . Thus, one has Definition 5.

Definition 5 (order relation \leq on TC). Given tc_1 and tc_2 in TC , $tc_2 \leq tc_1$ (meaning that tc_1 is stricter than tc_2), if and only if $ct_{21} \leq ct_{11}$ and $ct_{12} \leq ct_{22}$, where $tc_1 = (ct_{11}, ct_{12})$ and $tc_2 = (ct_{21}, ct_{22})$.

Proposition 6. \leq on TC is a partial order.

Definition 7 (order relation \leq on 2^{TC}). Given any TC_1 and TC_2 , $TC_1 \leq TC_2$ (meaning that TC_2 is stricter than TC_1), if and only if, for any $tc_x \in TC_2$, there exists $tc_y \in TC_1$ with $tc_y \leq tc_x$.

Proposition 8. \leq on 2^{TC} is a partial order.

In the real environment, time constraints can be constructed by way of union or intersection of some constraints. One defines the intersection and the union as follows.

Definition 9 (intersection). Intersection $\odot : 2^{TC} \times 2^{TC} \rightarrow 2^{TC}$ is a function from $2^{TC} \times 2^{TC}$ to 2^{TC} defined as

$$TC_1 \odot TC_2 = \{tc_x \odot_{TC} tc_y \mid tc_x \in TC_1 \wedge tc_y \in TC_2\}, \quad (1)$$

where

$$tc_x \odot_{TC} tc_y = \begin{cases} (x, y) & \text{if } x \leq y \\ \emptyset & \text{otherwise,} \end{cases} \quad (2)$$

$$tc_x = (ct_{x1}, ct_{x2}), \quad tc_y = (ct_{y1}, ct_{y2}),$$

$$x = \max(ct_{x1}, ct_{y1}), \quad y = \min(ct_{x2}, ct_{y2}).$$

Definition 10 (union). Union $\oplus : 2^{TC} \times 2^{TC} \rightarrow 2^{TC}$ is a function from $2^{TC} \times 2^{TC}$ to 2^{TC} defined as $TC_1 \oplus TC_2 = \{tc \mid tc \in TC_1 \vee tc \in TC_2\}$.

Because the time constraints obtained by computing $TC_1 \oplus TC_2$ are not always the simplest, one reduces them to the simplest form as follows. Given a time constraint TC , if there exists $(ct_{11}, ct_{12}), (ct_{21}, ct_{22}) \in TC$ with $\max\{ct_{11}, ct_{21}\} < \min\{ct_{12}, ct_{22}\}$, then both (ct_{11}, ct_{12}) and (ct_{21}, ct_{22}) are

deleted from TC and $(\min\{ct_{11}, ct_2lt_1\}, \max\{ct_{12}, ct_{22}\})$ are inserted into TC . Obviously, the original TC is semantically equivalent to the modified TC . Henceforth, one assumes that the intersection and the union of time constraints are always the simplest.

Proposition 11. *Given any tc_1 and tc_2 in TC , if TC is closed under \odot and \oplus , then $tc_1 \odot tc_2$ and $tc_1 \oplus tc_2$ are the supremum and the infimum of $\{tc_1, tc_2\}$, respectively.*

The definitions above concentrate on physical time. However, in some cases, logical time (such as work time or class time) is more important.

Definition 12. *timeAssigned: $LT \rightarrow 2^{TC}/\{\emptyset\}$ is a function mapping LT to the nonempty power set of TC , where $LT = \{lt_1, \dots, lt_n\}$ represents a set of names of logical time.*

Definition 13 (order relation \leq on LT). *Given any lt_1 and lt_2 in LT , $lt_1 \leq lt_2$ if and only if $timeAssigned(lt_1) \leq timeAssigned(lt_2)$.*

Proposition 14. *\leq on LT is a partial order.*

Definition 15 (intersection on LT). (Here, we do not differentiate the \odot of Definition 9 from the \odot of Definition 15, because they are easily distinguished; Similarly, we also do not differentiate \oplus of Definition 10 and \oplus of Definition 16). $\odot : LT \times LT \rightarrow LT$ is an intersection function mapping $LT \times LT$ to LT , defined as $lt_1 \odot lt_2 = lt$, where $timeAssigned(lt) = \{x \odot y \mid x \in timeAssigned(lt_1) \text{ and } y \in timeAssigned(lt_2)\}$.

Definition 16 (union on LT). $\oplus : LT \times LT \rightarrow LT$ is a union function mapping $LT \times LT$ to LT , defined as $lt_1 \oplus lt_2$, where

$$\bigcup_{\substack{x \in timeAssigned(lt_1) \\ y \in timeAssigned(lt_2)}} x \oplus y. \quad (3)$$

Proposition 17. *For any lt_1 and lt_2 in LT , if LT is closed under \odot and \oplus , then $lt_1 \odot lt_2$ and $lt_1 \oplus lt_2$ are the supremum and the infimum of $\{lt_1, lt_2\}$, respectively.*

Proof that the supremum of $\{lt_1, lt_2\}$ is $lt_1 \odot lt_2$: from Definition 15, we have $lt_1 \leq lt_1 \odot lt_2$ and $lt_2 \leq lt_1 \odot lt_2$; therefore, $lt_1 \odot lt_2$ is the upper boundary of $\{lt_1, lt_2\}$. Next, we prove that $lt_1 \odot lt_2$ is the least element of the upper boundary of $\{lt_1, lt_2\}$. Let $lt_1 \leq lt$ and $lt_2 \leq lt$; then, for any $x \in timeAssigned(lt)$, there exists $y_1 \in timeAssigned(lt_1)$ and $y_2 \in timeAssigned(lt_2)$, such that $y_1 \leq x$ and $y_2 \leq x$. According to Proposition 11, $y_1 \odot y_2 \leq x$. According to Definition 15, $lt_1 \odot lt_2 \leq lt$; therefore, so $lt_1 \odot lt_2$ is the supremum of $\{lt_1, lt_2\}$.

3.3. Location Description. In the IoT, physical locations are often distinguished from logical locations. Physical locations are divided into two classes: hierarchical (topological, descriptive, or symbolic), such as a room, and Cartesian (coordinate, metric, or geometric), such as GPS position [12, 14, 26]. Logical locations represent the boundaries of the logical space that corresponds to the physical space.

Let $PLOC = \{ploc_1, \dots, ploc_n\}$ be a set of physical locations, where $ploc_i$ ($1 \leq i \leq n$) is a specific physical location,

such as a 50×50 unit square area. Let $LLOC = \{lloc_1, \dots, lloc_n\}$ represent the set of logical locations, where each in $LLOC$ denotes the notion for one or more physical locations. Generally, relations between physical and logical locations are illustrated as a many-to-many map, denoted by $LP \subseteq PLOC \times LLOC$.

Definition 18. A function $LlocToPloc: LLOC \rightarrow 2^{PLOC}$ maps $LLOC$ to the power set of $PLOC$, returning all physical locations assigned to a given logical location. In other words, $ocToPloc(lloc) = \{ploc \mid (ploc, lloc) \in LP\}$.

Definition 19. A function $PlocToLloc: PLOC \rightarrow 2^{LLOC}$ maps $PLOC$ to the power set of $LLOC$, returning all assigned logical locations of a given physical location. In other words, $ocToLloc(ploc) = \{lloc \mid (ploc, lloc) \in LP\}$.

Given two logical locations, a containment relation may exist. This is defined as follows.

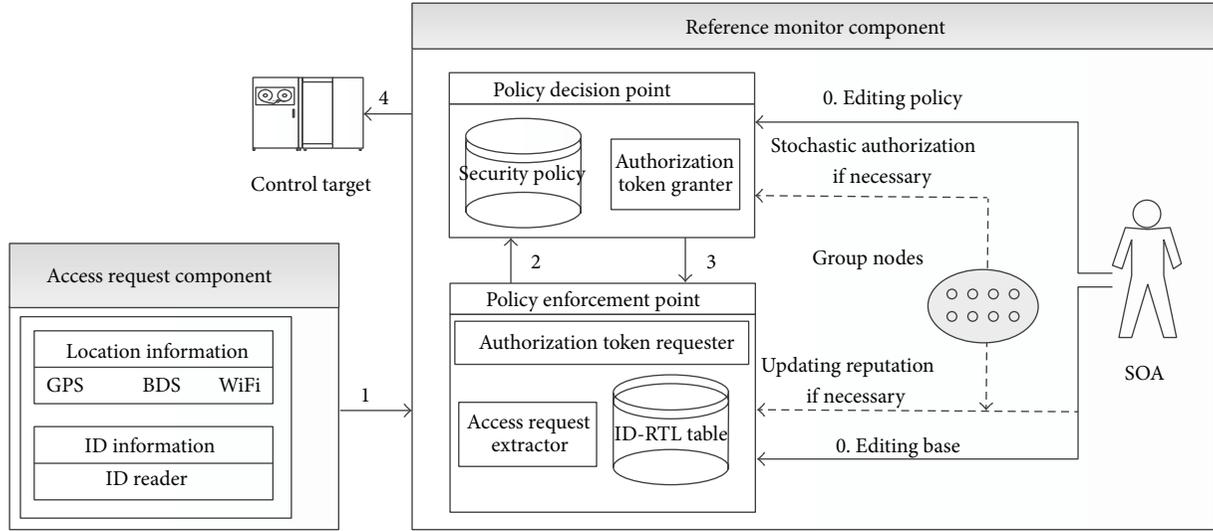
Definition 20. A logical location $lloc_i$ is contained in another logical location $lloc_j$, written as $lloc_i \subseteq lloc_j$, if and only if $LlocToPloc(lloc_i) \subseteq LlocToPloc(lloc_j)$.

Generally, physical locations of a given logical location are unchanged within a period; therefore, for simplicity, a logical location is used to denote its corresponding physical location. Similarly, one can define intersection \cap and union \cup based on logical locations, but one does not discuss them.

4. STRAC Framework

First, we provide an overview of the framework of our model. As shown in Figure 2, STRAC consists of two core components: the access request component (ARC) and the reference monitor component (RMC). The ARC of a node creates an access request, which has two forms: the first form includes five elements: the node's ID , the accessed object o , the expected operation op to be performed on o , the node's current location $ploc$ (generally, a node's location information may come from GPSs or wifi), and the node's reputation rep ; the second form includes three elements: the node's ID , the accessed object o , the expected operation op to be performed on o , and the node's current location $ploc$. In the first form, each node locally stores its reputation and physical location; in the second form, the reputation of each node is centrally stored in PEP (see the next paragraph for PEP) and its physical location is tracked by PEP.

RMC includes two modules: policy enforcement point (PEP) and policy decision point (PDP). The PEP module receives the user request, consults with the PDP module about the user authorization, and ensures that all access requests go through the PDP module. PEP is comprised of two submodules (access request extractor (ARE) and authorization token requester (ATR)) and two access tables which map the current time and physical locations to the logical time and the logical location, respectively (the two tables are called ID-RTL table in Figure 2). When the PEP module receives an access request from a user, ARE executes two steps: (1) it accepts the request and extracts the encapsulated



BDS: beidou navigation satellite system
 SOA: source of authority

FIGURE 2: Framework of STRAC.

location, ID , o , rep , op , and $ploc$ if the first type of access requests is adopted; it extracts ID , o , and $ploc$ if the second type is used, and (2) it queries the access database and returns the ID 's logical location $lloc$ and logical time lt (in addition to the three elements, it returns the reputation of node ID if the second type of access requests is used). ATR encapsulates this information (rep , lt , $lloc$, op , and o) and sends it to the PDP module to request an authorization token (AT). Once this request is granted, an AT will be returned, and the user can access the target resources using this AT. PEP maintains a list of users' ATs, and this list is updated at a specified interval. AT will be revoked when the user deactivates the task or when the location and time associated with the use are out of the allowed scope.

The PDP module, comprised of one submodule (Authorization Token Granter (ATG)) and one policy base, makes the authorization decision based on a set of rules or policies. When PDP receives an AT request, it extracts the information (rep , lt , $lloc$, op , and o) from the request and consults the security policies. If the policies denote that a node with reputation rep at the time period lt in the location $lloc$ has the right to perform the operation op on the target o , then ATG grants AT to access request.

The source of authority (SOA) is administrator or a group of administrators, who define the policies. SOA can also update the policy at runtime if necessary. In some cases, group nodes may also cooperatively update the node's reputation and make access decisions based on stochastic information (stochastic authorization in Figure 2).

Our model can be implemented with two alternative modes: ACI and AII. ACI focuses on authorization when complete information is available, while AII deals with authorization having only incomplete information. The precondition for ACI is that decision makers always can obtain authorization information in time. In other word, ACI

requires a stable communication. Contrarily, this precondition is not necessary in AII. AII is very useful because unstable communications in the IoT are considered to be persuasive phenomenon. Generally, while in the ACI mode both PEP and PDP are mounted in the gateway and ARC is integrated into terminal nodes. Contrarily, to implement the AII mode, ARC and two lightweight modules, PEP and PDP, are mounted into terminal nodes.

5. STRAC Model

5.1. Basic Components of STRAC. The basic STRAC model is comprised of the following components:

$ID = \{id_1, \dots, id_n\}$ is a set of node IDs;

$REP = \{rep_1, \dots, rep_n\}$ is a set of reputations; for example, $REP = \{goldreputation, silverreputation\}$;

$LT = \{lt_1, \dots, lt_n\}$ is a set of logical time constraints;

$LLOC = \{lloc_1, \dots, lloc_n\}$ is a set of logical locations; for example, $LLOC = \{administratorroom, meetroom\}$;

$OP = \{op_1, \dots, op_n\}$ is a set of operations; for example, $OP = \{open, close\}$;

$O = \{o_1, \dots, o_n\}$ is a set of target objects; for example, $O = \{TV, Microwave\}$;

$PERM \subseteq OP \times O$ is a set of permissions, where $(op, o) \in PERM$ means that op is performed on o . For example, if $PERM = \{(open, TV)\}$ is assigned to device id , then the device owns the permission to open TV;

$AccZone \subseteq REP \times LT \times LLOC$ is a set of access zones, where each access zone is a triple $(rep, lt, lloc)$;

$PA \subseteq PERM \times AccZone$ is a many-to-many map of connections between permissions and access zones.

$(x, y) \in PA$ means that any node with y has permission x . For example, $((op, o), (rep_1, lt_1, lloc_1)) \in PA$ denotes that a node satisfying the constraint of lt_1 with reputation rep_1 at location $lloc_1$ can execute operation op on object o ;

AccZoneAssigned: $PERM \rightarrow 2^{AccZone}$ assigns a permission level to access zones, where $AccZoneAssigned(perm) = \{AccZone \mid (perm, AccZone) \in PA\}$; that is, given a $perm$, function *AccZoneAssigned* returns all access zones which the $perm$ can access. For example, if $PA = \{((op_1, o_2), (rep_2, lt_1, lloc_2)), ((op_2, o_1), (rep_2, lt_2, lloc_2)), ((op_1, o_2), (rep_2, lt_3, lloc_1))\}$, then $AccZoneAssigned((op_1, o_2)) = \{(rep_2, lt_1, lloc_2), (rep_2, lt_3, lloc_1)\}$;

permAssigned: $AccZone \rightarrow 2^{PERM}$ assigns an access zone to permissions, where $permAssigned(AccZone) = \{perm \mid (perm, AccZone) \in PA\}$; that is, given an $acczone$, function *permAssigned* returns all permissions by which the $acczone$ can be accessed. For example, in the example of *AccZoneAssigned*, $permAssigned((rep_2, lt_1, lloc_2)) = \{(op_1, o_2), (op_2, o_1)\}$;

AccessRequest: $ID \times OP \times O \rightarrow \{true, false\}$ is a predicate; if it returns true, then node id requests permission to execute op on o . Recall that the storage of rep is either distributed or centralized. To model the two cases, we remove rep from access requests; for simplicity, we also remove the ID's location and reputation from access requests and encapsulate the three elements into the function *CurrentRTL* (we will discuss it next);

AllowRequest: $ID \times OP \times O \rightarrow \{true, false\}$ is a predicate; if it returns true, then node id is allowed to op on o ;

DenyRequest: $ID \times OP \times O \rightarrow \{true, false\}$ is a predicate; if it returns true, then node id is not allowed to execute op on o ;

RevokeRequest: $ID \times OP \times O \rightarrow \{true, false\}$ is a predicate; if it returns true, then the permission that node id executes op on o will be revoked;

CurrentRTL: $ID \rightarrow REP \times CT \times 2^{LLOC}$ is a function and returns id 's current reputation, its logical time, and logical locations (a node may be located in many different logical locations).

In order to return the logical locations of a node, its physical location must be first obtained, and then its logical locations can be computed using the function *PlocToLloc*. Because a physical location can be associated with many logical areas, *CurrentRTL* returns a set of logical locations.

5.2. Mechanism for Authorization and Revocation. Intuitively, if a node located in an appropriate area has an acceptable reputation and satisfies the given time constraints, its requests to execute some operations on an object should be allowed. In order to avoid using too many symbols, we overload the notation \in , as follows.

Given id , op , and o , let $CurrentRTL(id) = \{rep_{id}, ct_{id}, \{lloc_{id1}, \dots, lloc_{idn}\}\}$ and $AccZoneAssigned((op, o)) = \{(rep_1, lt_1, lloc_1), \dots, (rep_n, lt_n, lloc_n)\}$; $CurrentRTL(id) \in AccZoneAssigned((op, o))$ if and only if there exists $(rep_i, lt_i, lloc_i) \in AccZoneAssigned((op, o))$ with $rep_{id} = rep_i$, $ct_{id} \in lt_i$ ($ct_{id} \in lt_i$ if and only if $(ct_1, ct_2) \in timeAssigned(lt_i)$ holds, where $ct_1 \leq ct_{id} \leq ct_2$) and $lloc_i \in \{lloc_{id1}, \dots, lloc_{idn}\}$. The authorization schemes are as follows:

- (1) $AccessRequest(id, op, o) \wedge CurrentRTL(id) \in AccZoneAssigned((op, o)) \rightarrow AllowRequest(id, op, o)$
- (2) $AccessRequest(id, op, o) \wedge CurrentRTL(id) \notin AccZoneAssigned((op, o)) \rightarrow DenyRequest(id, op, o)$
- (3) $AllowRequest(id, op, o) \wedge CurrentRTL(id) \notin AccZoneAssigned((op, o)) \rightarrow RevokeRequest(id, op, o)$.

Formulas (1) and (2) show the following: (1) if the node id requests permission to execute op on o , and if $CurrentRTL(id)$ (the current reputation, access time, and location of id) satisfies the conditions to execute op on o , then the request will be allowed; (2) if the node id requests permission to execute op on o , but $CurrentRTL(id)$ does not satisfy the conditions to execute op on o , then the request will be denied. Formula (3) suggests that if node id has received the permission of executing op on o and $CurrentRTL(id)$ no longer satisfies the conditions to execute op on o longer, the permission will be revoked.

6. Access Lattice

Because terminal nodes could move into many areas at different times, enumerating all areas and periods of time rapidly increases the size of PA (as shown above, PA connections between permissions and the power set of access zones). As a result, the size of the PA table could exceed the storage capacity. In addition, querying a big table consumes more energy and computing resources, thereby decreasing the efficiency of queries and even reducing a node's lifetime. Thus, decreasing the size of permission access table is critical. In order to achieve this goal, we adopted the access lattice in this study.

We make the following realistic assumptions regarding the sensing layer of the IoT. (1) A node with a high reputation can be granted all permissions of a lower-reputation node. (2) If a task can be executed in a wide area or a longer time period, then it can be also executed in a narrow area or a shorter time period. These assumptions mean that if one node owns two access zones A and B , where A is stricter than B , then A can be omitted from the set of access zones, because any permission allowed under B is allowed under A . We chose the lattice to decrease the size of the permission access table, because it models the strict relationship among elements. In order to formally describe the access lattice, we first define the order relation.

Definition 21 (order relation \leq on $AccZone$). Given any $(rep_1, lt_1, lloc_1)$ and $(rep_2, lt_2, lloc_2) \in AccZone$, $(rep_1, lt_1, lloc_1) \leq (rep_2, lt_2, lloc_2)$, if and only if $rep_1 \leq rep_2$, $lt_1 \leq lt_2$, $lloc_2 \subseteq lloc_1$.

Theorem 22. *If (1) LT is closed under \odot and \oplus and (2) $LLOC$ is closed under \cap and \cup , then $(ACCBASE, \leq)$ is a lattice.*

Proof that there exists a supremum and an infimum for any $(rep_1, lt_1, lloc_1), (rep_2, lt_2, lloc_2) \in ACCBASE$: if $(rep_1, lt_1, lloc_1)$ and $(rep_2, lt_2, lloc_2)$ are comparable, then there exists a supremum and an infimum for them. Even if $(rep_1, lt_1, lloc_1)$ and $(rep_2, lt_2, lloc_2)$ are incomparable, there exists a supremum and an infimum for them. Because \leq on REP is a total order, there exists a supremum for $\{rep_1, rep_2\}$; let the supremum be rep_x , because LT is closed under both \odot and \oplus , and LOC is closed under \cap and \cup ; therefore, $(rep_x, lt_1 \odot lt_2, lloc_1 \cap lloc_2)$ is in $AccZone$ and is the upper boundary of $(rep_1, lt_1, lloc_1)$ and $(rep_2, lt_2, lloc_2)$. Let $(rep, lt, lloc)$ be another upper boundary of $(rep_1, lt_1, lloc_1)$ and $(rep_2, lt_2, lloc_2)$, then we have $loc \subseteq lloc_1$ and $loc \subseteq lloc_2$; therefore, $lloc \subseteq lloc_1 \cap lloc_2$. Because rep_x is the supremum of $\{rep_1, rep_2\}$, $rep_x \leq rep$. According to Proposition 17, we have $lt \leq lt_1 \odot lt_2$. Thus, $(rep_x, lt_1 \odot lt_2, lloc_1 \cap lloc_2) \preceq (rep, lt, lloc)$. Therefore, $(rep_x, lt_1 \odot lt_2, lloc_1 \cap lloc_2)$ is a supremum of $(rep_1, lt_1, lloc_1)$ and $(rep_2, lt_2, lloc_2)$. Similarly, $(rep_y, lt_1 \oplus lt_2, lloc_1 \cup lloc_2)$ is an infimum of $(rep_1, lt_1, lloc_1)$ and $(rep_2, lt_2, lloc_2)$, where rep_y is an infimum of $\{rep_1, rep_2\}$.

We assume that LT is closed under \odot and \oplus , $LLOC$ is closed under \cap and \cup . We redefine the permission function under a lattice, as follows:

$permAssigned_{lattice}: AccZone \rightarrow 2^{PERM}$ maps the access zones to permissions, and $ermAssigned_{lattice}(AccZone) = \{perm \in permAssigned(AccZone_x) \mid AccZone_x \preceq AccZone\}$.

Because $AccZoneAssigned_{lattice}$ and $ermAssigned$ are similar and easily distinguished from one another; therefore, $AccZoneAssigned$ is adopted to denote the two functions in the sequel. Similarly, $permAssigned$ is used to denote $permAssigned_{lattice}$.

Theorem 23. *Given any $(rep_1, lt_1, lloc_1)$ and $(rep_2, lt_2, lloc_2) \in AccZone$, if $(rep_1, lt_1, lloc_1) \preceq (rep_2, lt_2, lloc_2)$, then $permAssigned((rep_1, lt_1, lloc_1)) \subseteq permAssigned((rep_2, lt_2, lloc_2))$.*

Theorem 24. *Consider the following.*

If $AllowRequest(id, op, o) \wedge AccessRequest(id', op, o) \wedge CurrentRTL(id) \preceq CurrentRTL(id')$, then $AllowRequest(id', op, o)$.

If $DenyRequest(id, op, o) \wedge AccessRequest(id', op, o) \wedge CurrentRTL(id') \preceq CurrentRTL(id)$, then $DenyRequest(id', op, o)$.

If $RevokeRequest(id, op, o) \wedge AllowRequest(id', op, o) \wedge CurrentRTL(id') \preceq CurrentRTL(id)$, then $RevokeRequest(id', op, o)$.

Theorem 24 shows (1) if a node with a low reputation can execute op on o , then another node with a higher reputation is also able to perform the same operation; (2) if a node with a high reputation is unable to execute operation op on o , then a node with a lower reputation is also unable to do so; and (3) if access permissions are revoked from a node with a high reputation, then the corresponding permissions are also revoked from a node with a lower reputation.

The following example illustrates that the lattice can efficiently decrease the size of policy bases.

Example 25. Let $PERM = \{(Open, MicroWave), (SetParameter, MicroWave), (close, MicroWave)\}$, $ACCBSE = \{(rep_1, lt_1, lloc_1), (rep_2, lt_2, lloc_2), (rep_3, lt_3, lloc_3), (rep_4, lt_4, lloc_4)\}$. The access base for each permission is as follows: $AccZoneAssigned((close, MicroWave)) = \{(rep_1, lt_1, lloc_1), (rep_2, lt_2, lloc_2), (rep_3, lt_3, lloc_3), (rep_4, lt_4, lloc_4)\}$, $AccZoneAssigned((SetParameter, MicroWave)) = \{(rep_3, lt_3, lloc_3), (rep_4, lt_4, lloc_4)\}$ and $AccZoneAssigned((Open, MicroWave)) = \{(rep_4, lt_4, lloc_4)\}$. From above, the cardinality of $AccZoneAssigned((Close, MicroWave))$, $AccZoneAssigned((SetParameter, MicroWave))$, and $AccZoneAssigned((Open, MicroWave))$ is 4, 2, and 1, respectively.

Assuming that a lattice can be formed from these access zones as shown in Figure 3, then the access base for each permission could be changed as follows: $AccZoneAssigned((close, MicroWave)) = \{(rep_1, lt_1, lloc_1)\}$, $AccZoneAssigned((SetParameter, MicroWave)) = \{(rep_3, lt_3, lloc_3)\}$, and $AccZoneAssigned((Open, MicroWave)) = \{(rep_4, lt_4, lloc_4)\}$. This means that the cardinality of all three bases is 1. Given an access request $AccessRequest(id, Close, MicroWave)$ from node id with reputation rep_3 and assuming that node id is at the location $lloc_3$ and the current time satisfies lt_3 , then $AllowRequest(id, Close, MicroWave)$ will be true, because $(rep_1, lt_1, lloc_1) \preceq (rep_3, lt_3, lloc_3)$. From this example, we can deduce that the size of policy bases can be decreased using an access lattice.

Regarding the example given in the beginning of this section, if the above lattice is used, the storage complexity and the computing complexity are reduced to n and 1, respectively.

7. Authorization under Incomplete Information

In the above discussion, we mainly focused on authorization with complete information available (ACI), which is also called deterministic authorization. In other words, decision makers have the ability to obtain authorization information in time. However, this is not always the case. For example, when a node moves into a location where communication is unstable, it may not be able to obtain complete authorization information in time. In this case, decision makers have to choose whether to grant authorization or not, based on their own knowledge, such as historical experiences. This is related to authorization with incomplete information (AII). Lack of complete information presents the following challenges: (1) designing a secure authorization policy and (2) balancing security with QoS. To address these two challenges, we propose both nondeterministic authorizations and stochastic authorizations.

7.1. Nondeterministic Authorization. Nondeterministic authorization includes three alternative policies: pessimistic, optimistic, and compromise authorizations. Nodes run under the lowest permission levels in a pessimistic authorization, thus, providing only the most basic security.

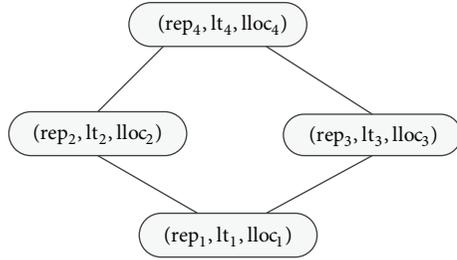


FIGURE 3: An access lattice.

Policy 1 (pessimistic authorization). Given an access lattice formed by $AccZone$ and an access request $AccessRequest(id, op, o)$, if $(op, o) \in permAssigned(glb(AccZone))$, then the request is allowed, where $glb(AccZone)$ represents the greatest lower bound of set $AccZone$ (i.e., the smallest elements of $AccZone$).

In order to execute Policy 1, all elements in $permAssigned(glb(AccZone))$ must be locally stored. Although additional storage space is required by Policy 1, because both the greatest lower bounds of a lattice are unique, the storage complexity of the access rule table in Policy 1 is $|op| \times |o|$; therefore, it is acceptable for the majority of weak-resources devices. Because $glb(AccZone)$ is granted to the lowest permissions, Policy 1 is always secure (see Theorem 27 for the proof); however, this policy might have a lower QoS. For example, if the least permission is ϕ (empty), then any request will be automatically denied.

Contrary to pessimistic authorization, optimistic authorization concentrates on QoS while ignoring security. In this policy, nodes are granted the highest allowable permissions. From Theorem 22, $lub(AccZone)$ is the largest element of $AccZone$, where $lub(AccZone)$ represents the least upper boundary of $AccZone$. Thus, the greatest permissions are granted to $lub(AccZone)$.

Policy 2 (optimistic authorization). Given an access lattice formed by $AccZone$ and an access request $AccessRequest(id, op, o)$, if $(op, o) \in permAssigned(lub(AccZone))$, then the request is allowed.

As with Policy 1, all elements of $permAssigned(lub(AccZone))$ must be locally stored. Because any node in this policy is granted the greatest permissions, Policy 2 is not considered to be secure (see Example 28 for an example). However, this policy can efficiently improve QoS. For example, if the number of nodes with access to a film in digital rights management (DRM) is only optionally counted, any node can access this film anytime and anywhere, even when communications have been interrupted. If the film is stored locally, Policy 2 should be adopted, because security is not as much of an issue; if Policy 1 was used, the benefits surrounding optionally counting accesses cannot be achieved. Thus, Policy 2 has a higher QoS.

In many cases, security and QoS are needed to be balanced. To achieve this goal, we propose three compromise authorization policies: (1) trade-off authorization based on

reputation ($TABR$), (2) trade-off authorization based on space ($TABS$), and (3) trade-off authorization based on time ($TABT$). We first discuss $TABR$.

Let $comp_rep: REP \rightarrow AccZone$ be a function from REP to $AccZone$, defined as $comp_rep(rep) = glb\{(rep_x, lt, lloc) \in AccZone \mid rep_x = rep\}$, representing the greatest lower boundary of access zones accessed by a node with reputation rep .

Policy 3 ($TABR$). Given an access lattice formed by $AccZone$ and an access request $AccessRequest(id, op, o)$ in AII , if $(op, o) \in permAssigned(comp_rep(rep))$, where rep is the reputation of node id , then the request is allowed.

Similarly, let $comp_loc: LLOC \rightarrow AccZone$ map from logical locations to $AccZone$, defined as $comp_loc(lloc) = glb\{(rep, lt, lloc_x) \in AccZone \mid lloc_x = lloc\}$, representing the greatest lower boundary of access zones accessed by a node in logical location $lloc$.

Policy 4 ($TABS$). Given an access lattice formed by $AccZone$ and access request $AccessRequest(id, op, o)$ in AII , if $(op, o) \in permAssigned(comp_loc(lloc))$, where $lloc$ is the current logical location of node id , then the request is allowed.

Let $comp_lt: LT \rightarrow AccZone$ be a function from LT to $AccZone$, defined as $comp_lt(lt) = glb\{(rep_x, lt, lloc) \in AccZone \mid lt_x = lt\}$, representing the greatest lower boundary of access zones accessed by a node satisfying time constraint lt .

Policy 5 ($TABT$). Given an access lattice formed by $AccZone$ and an access request $AccessRequest(id, op, o)$ in AII , if $(op, o) \in permAssigned(comp_lt(lt))$, where $ct() \in lt$ (function $ct: \phi \rightarrow CB$ returns the current time), then the request is allowed.

Because Policies 1–5 are related to incomplete information, their security must be formally analyzed.

Definition 26 (security under AII). A policy is secure under AII if for any access request $AccessRequest(id, op, o)$, $AllowRequest(id, op, o)$ under AII is the same as $AllowRequest(id, op, o)$ under ACI .

Theorem 27. *Policy 1 and Policies 3–5 are secure.*

Proof that Policy 1 is secure: given node id with reputation rep and an access request (id, op, o) , because (id, op, o) is allowed under Policy 1, we have $(op, o) \in permAssigned(lub(AccZone))$. Let $(rep, ct(), lloc)$ be the reputation, the current time, and the logical location of node id under ACI , where $(rep, lt, lloc)$ exists with $(rep, ct(), lloc) \in (rep, lt, lloc) ((rep_1, ct(), lloc_1) \in (rep_2, lt, lloc_2), \text{ if } rep_1 = rep_2 \text{ and } ct() \in lt \text{ and } lloc_1 = lloc_2)$. Because $glb(AccZone)$ is the least element of $AccZone$, we have $glb(AccZone) \preceq (rep, lt, lloc)$. According to Policy 1 and Theorem 23, we have $(op, o) \in permAssigned(glb(AccZone)) \subseteq permAssigned((rep, lt, lloc))$; therefore, $(rep, lt, lloc) \in AccZoneassigned((op, o))$, $AllowRequest(id, op, o)$ under ACI is true. That is, Policy 1 is secure.

Although Policy 1 and Policies 3–5 are secure, Policy 2 is insecure. An example is as follows.

Example 28. Let $AccZone = (rep, lt, loc)$ with $(rep, lt, loc) < lub(AccZone)$. According to Theorem 23, there exists (op, o) with $(op, o) \in permAssigned(lub(AccZone)) - permAssigned((rep, lt, loc))$. $AccessRequest(id, op, o)$ will be allowed if Policy 2 is adopted; however, this request will be denied under *ACI*. This means that Policy 2 is insecure. Because security obeys the “leaky bucket” principle, Policy 2 is not suitable for security-critical systems.

7.2. Stochastic Authorization. In some cases, authorization may be considered to be stochastic. For example, when an automatically driven car arrives at crossroads, its central controller selects the road with the most gains (including both time and fuel-saving gains) by computing the distance to the destination and forecasting the probability of road congestion. In this case, road congestion is stochastic; as a result, the authorization is stochastic. Although stochastic authorization is actual requirement in the IoT, no efforts are spent on it in existing studies. In our study, we propose the (to the best of our knowledge) expectation-based authorization (EBA), where decision makers evaluate access requests by analyzing its potential gains in the successor states that would occur if the authorizations were granted. If the potential gains of the request are greater than or equal to a given threshold, it will be allowed. *EBA* includes the following elements besides the components mentioned in Section 5.1:

- (1) $S = \{s_1, \dots, s_n\}$ is a set of states representing the potential successor states which the systems can arrive to after authorization.
- (2) $f: S \rightarrow R$ is a gain function, and $f(s)$ represents the gain of decision makers in state s , where R is a real number.
- (3) $p: S \rightarrow R$ is a probability distribution of state set S , and $p(s)$ denotes the probability that systems would reach state s .
- (4) Threshold r .

The authorization rule is as follows: for any Access Request (id, op, o) , if $\sum_{s=1}^S p(s) \times f(s) \geq r$, then this request is granted. If several access requests exist and only one is allowed, then the request with the maximum gain is granted permission. Additionally, if multiple requests could have the maximum gain, then the decision maker randomly chooses among those requests.

As with Policy 2, expectation authorization improves the QoS. In the above example of automatic-driving systems, pessimistic authorization would deny any access request to each road, whereas optimistic authorization would allow all access requests for all roads. This is unacceptable for the automatic-driving systems. Thus, nondeterministic authorization (pessimistic/optimistic authorization) is unsuitable for automatic-driving systems. Conversely, because both the road environment and the historical experiences are taken into consideration to ensure that only the request with the

TABLE 1: Syntax of update policies.

$update ::=$	Policies
$rep \triangleright_c update$	update policy
ϕ	empty policy
$c ::=$	Condition
basic predicate	basic predicate
T	True
F	False
$c \vee c$	Disjunction
$\neg c$	Negation

maximum gain is allowed, stochastic authorization is suitable for automatic-driving systems.

8. Mechanism for Updating Reputation

Because nodes of the IoT are easily tampered with, the reputation of the tampered nodes must be updated in time. In this section, we borrow the idea of downgrading in programming languages [27] and propose a new mechanism to update the reputation for the sensing layer of the IoT. In this mechanism, every node is bound to an update policy and any change to a node's reputation must be consistent with the bound policy; that is, the reputation can be updated, only if a given policy for that node is satisfied. The syntax of the general update mechanism is defined in Table 1.

The above definitions are based on Backus Normal Form; for example, $rep \triangleright_{c_1} rep \triangleright_{c_2} \phi$ is grammatically correct. In Table 1, if the empty policy (ϕ) is adopted for a node, then its reputation is preserved. Given $\triangleright_{c_1} update$ for a node means that if condition c_1 is true, then the reputation of the node will be updated to rep_1 , and the successor of the policy is updated to $update$. The formal semantics of this mechanism are as follows:

$$\begin{aligned}
 & rule_{general}: \\
 & \frac{c}{id : rep \triangleright_c update \implies id : update [rep/currentREP(id)]} \\
 & rule_{\phi}: \frac{}{id : \emptyset \implies \emptyset}, \tag{4}
 \end{aligned}$$

where $id : rep \triangleright_c update$ represents the node id as bound to $rep \triangleright_c renew$ and function $currentREP: ID \rightarrow REP$ returns the current reputation of a given node. $[rep/currentREP(id)]$ denotes that $currentREP(id)$ is substituted by rep . $rule_{general}$ indicates that if the predicate c is true, then the current reputation of node id will be updated to rep and its new policy is $update$. $rule_{\phi}$ stops the update. Specifically, when the update policy for a node is $rep \triangleright_T update$, then its reputation will be mandatorily updated to rep . In contrary, if the policy is $rep \triangleright_F update$, then the update will be forbidden.

In Table 1, basic predicates are coarse-grained and difficult to use. In order to solve this problem, we propose two submechanisms: authority-based update mechanisms (*AUM*) and election-based update mechanisms (*EUM*).

8.1. *Authority-Based Update Mechanisms (AUM)*. In AUM, only the authority node can update the reputation of other nodes. A node is called an authority node if its reputation is greater than or equal to a given threshold rep_{auth} , where rep_{auth} represents the authority reputation. To formally define semantics, we first provide the following definitions.

$currentTL: ID \rightarrow CB \times LLOC$ returns the current time and the set of current logical locations for a given node.

$UpdateTL \subseteq LT \times LLOC$ is a subset of the product of LT and $LLOC$. A node is granted the update permission, only if its current time and one of its current logical locations belong to the set $UpdateTL$. In other words, node id can update the reputation of other nodes, only if $currentTL(id) \in UpdateTL$ (let $currentTL(id) = (cb, \{lloc_1, \dots, lloc_n\})$ and $UpdateTL = \{(lt_{i_1}, lloc_{i_1}), \dots, (lt_{i_m}, lloc_{i_m})\}$, we define $currentTL(id) \in UpdateTL$, if and only if there exists $(lt_{i_k}, lloc_{i_k}) \in UpdateTL$ such that $cb \in lt_{i_k}$ and $lloc_{i_k} = lloc_{i_k}$, where $1 \leq k \leq n$).

The formal semantics of AUM are as follows:

$$\begin{aligned}
 & rule_{authority}: \\
 & currentREP(id_r) = rep_r \\
 & currentTL(id_r) \in UpdateTL \\
 & \frac{rep, currentREP(id) < rep_{threshold}}{rep, id : rep \triangleright_{authupdate(rep_r)} updatep \implies} \quad (5) \\
 & id : updatep [rep/currentREP(id)] \\
 & rule_{\phi}: \frac{}{id : \emptyset \implies \emptyset}.
 \end{aligned}$$

The predicate $authupdate(rep_r)$ is true, if and only if there exists $id \in ID$ with $currentREP(id_r) = rep_r$, $currentTL(id_r) \in UpdateTL$, $rep < rep_{threshold}$, $currentREP(id) < rep_{threshold}$, and $rep_{threshold} \leq rep_r$. The rule $rule_{authority}$ states that the reputation of node id can be updated to rep if the following conditions are satisfied: (1) $rep_{threshold} \leq currentREP(id_r)$; that is, node id_r is an authority node. (2) $currentTL(id_r) \in UpdateTL$; that is, node id_r satisfies the temporal and spatial constraint. (3) The current reputation of node id and its reputation after update are both greater than or equal to $rep_{threshold}$.

AUM can prevent a node's reputation from being illegally updated even if authority nodes are lost. For example, the authority node x in the location $lloc$ can update the reputation of others. If x is lost, it may not be in the location $lloc$. Let x be in the location loc_{theft} with $loc_{theft} \neq lloc$, therefore, making $currentTL(id) \notin UpdateTL$. This means that x no longer has the ability to update the reputation of other nodes.

8.2. *Election-Based Update Mechanisms (EUM)*. Although AUM can be used to decrease the risk caused by the loss of nodes, authority nodes are security-critical because of the huge risks involved if they are compromised. In order to solve this problem, we propose EUM. In EUM, a group of nodes with a lower reputation are able to update the reputation of others through elections. When the majority of voters (nodes) agree on an update, the reputation of a specific node can be

updated. In order to precisely define this mechanism, we first define the update function $f: ID \times ID \times REP \rightarrow \{0, 1\}$, which maps $ID \times ID \times REP$ to $\{0, 1\}$: If a voter id requests to update the reputation of candidate id_c to rep_c , then $f(id_c, id, rep_c) = 1$. The formal semantics of EUM is as follows:

$$\begin{aligned}
 & rule_{election}: \\
 & \frac{Y \geq r}{id_c : rep_c \triangleright_{Y \geq r} updatep \implies id : updatep [rep_c/currentREP(id_c)]} \quad (6)
 \end{aligned}$$

where $Y = \sum_{id}^{ID_{TL}} f(id_c, id, rep_c)$, $ID_{TL} = \{x \in ID \mid currentTL(x) \in UpdateTL, currentTL(x) \in UpdateTL\}$, and r is a natural number. $rule_{election}$ shows that if at least r nodes satisfying the given time constraints in the given areas request to update the reputation of node id to rep , then these requests will be approved and id 's reputation will be updated. Because $rule_{election}$ can update the reputation of any other node, r must be carefully defined, especially in networks where malicious nodes are dominant. Generally, r could be equal to $\lceil N/2 \rceil$ or $\lceil 2N/3 \rceil$, where N is the number of voters.

Example 29. Let the update policy of node id be $rep_0 \triangleright_{authupdate(rep_r)} rep_1 \triangleright_T \phi$; we have the following results.

- (1) If the current reputation of id is greater than $rep_{threshold}$, then the reputation of x will be preserved.
- (2) If authority node id_r , which is not located in the given areas or does not satisfy the given time constraints (i.e., $currentTL(id_r) \notin UpdateTL$), requests to update the reputation of node id will be denied and the reputation of id will be preserved.
- (3) If the current reputation of id is less than $rep_{threshold}$ and authority node id_x requests to update the reputation of node id , where $currentTL(id_x) \in UpdateTL$ and $rep_0 \leq rep_x$, then this request will be approved and the reputation of id will be updated to rep_0 .

Example 30. Let the update policy of node y be $rep_0 \triangleright_{Y \geq 10} rep_1 \triangleright_T \phi$ and let its current reputation be rep_0 . In this case, if at least 10 nodes at a given period and location send requests to update the reputation of x to rep_0 , then the reputation will be updated.

8.3. *Order Relation*. It is possible that a single node owns two update policies, with one policy being stricter than the other. In this case, an order relation can be constructed to decrease the consumption of computing resources. Let \leq_p denote that is stricter than; that is, if can be adopted, then can be used. \leq_p is recursively defined as follows (note that only $rule_{authority}$ is considered here):

$$\begin{aligned}
 & \leq_{pa} 1 : \phi \leq_p updatep \\
 & \leq_{pa} 2 : \\
 & \frac{rep_1 \leq rep_2, updatep_1 \leq_p updatep_2}{rep_1 \triangleright_{authupdate(rep_r)} updatep_1 \leq_p rep_2 \triangleright_{authupdate(rep_r)} updatep_2} \quad (7)
 \end{aligned}$$

$\leq_{pa} 1$ shows that the empty policy is the loosest because it can be used anywhere and anytime; $\leq_{pa} 2$ denotes that if (1) rep_2 is greater than or equal to rep_1 and (2) $updatep_2$ is stricter than $updatep_1$, then $rep_2 \triangleright_{authupdate(rep_r)} updatep_2$ is stricter than $rep_1 \triangleright_{authupdate(rep_r)} updatep_1$.

To study the properties of \leq_p , we define the necessary function $first$, which maps policy to the reputation, as follows:

$$first(p_{renew}) = \begin{cases} rep & \text{if } p_{renew} \equiv rep \triangleright_c p_{renew}' \\ \emptyset & \text{otherwise.} \end{cases} \quad (8)$$

Proposition 31. *If $updatep_1 \leq_p updatep_2$, then $first(updatep_1) \leq first(updatep_2)$.*

Proposition 32. *If $updatep_1 \leq_p updatep_2$ and $rep_1 \leq rep_2$, then $updatep_1[rep_1/first(updatep_1)] \leq_p updatep_2[rep_2/first(updatep_2)]$.*

Theorem 33. *(P, \leq_p) is a total order, where P is a set of policies.*

Theorem 34. *For any nonempty policies $updatep_1$ and $updatep_2$, such that $updatep_1 \leq_p updatep_2$, if there exists a nonempty policy $updatep_x$ with $id: updatep_2 \Rightarrow id: updatep_x$, then there also exists a nonempty policy $updatep_y$ with $id: updatep_1 \Rightarrow id: updatep_y$ and $updatep_y \leq_p updatep_x$.*

Proof. Let $updatep_2$ be $p_{20} \triangleright_{authupdate(rep_r)} updatep_x$. From $id: updatep_2 \Rightarrow id: updatep_x$ and rule_{authority}, we have $id_r \in ID$ with $currentREP(id_r) = rep_r$, $currentTL(id_r) \in UpdateTL$, $rep < rep_{thersshold}$, $currentREP(id) < rep_{thersshold}$, and $rep_{thersshold} \leq currentREP(id_r)$. Let $updatep_1$ be $p_{10} \triangleright_{authupdate(rep_r)} updatep_{yi}$. Because $updatep_1 \leq_p updatep_2$, we have $rep_{10} \leq rep_{20}$ and $updatep_{yi} \leq_p updatep_x$. Because rule_{authority}, we have $id: updatep_1 \Rightarrow id: updatep_{yi}$ [$rep_{10}/currentREP(id)$]. From Proposition 32, we have $updatep_y \leq_p updatep_x$, where $updatep_y = updatep_{yi}[rep_{10}/currentREP(id)]$. \square

9. Verification of Security Policy

The STRAC model has many features that could interact with each other, causing conflict and inconsistency between security policies. As a result, security policies must be verified before they are applied. Tediousness and proneness of manual analyses make automatic verification necessary. UPPAAL [28] is an integrated model checker for modeling, validation, and verification of real-time systems modeled as networks of timed automata. In this study, UPPAAL is used to verify whether the policy conforms to security requirements or not. When requirements are violated, the tool pictorially shows how the property has been violated and generates a counterexample to help security designers fix the policy.

To illustrate how to formally specify and verify a STRAC policy, we consider smart home applications. We assume the existence of four smart devices in this example: (1) a TV set (TV), (2) an air conditioning (AC) unit, (3) a microwave (MW), and (4) an electric rice cooker (RC). Each of these devices can be remotely controlled using mobile terminals.

TABLE 2: PA of smart home applications.

PERM	Access Zones
(open, TV)	$REP \times \{TVtime\} \times LLOC$
(open, AC)	$REP \times \{ACtime\} \times LLOC$
(open, MW)	$\{highRep\} \times \{MWtime\} \times LLOC$
(config, MW)	$\{highRep\} \times \{MWtime\} \times LLOC$
(open, RC)	$REP \times \{RCtime\} \times \{home\}$
$\{close\} \times O$	$REP \times LT \times LLOC$

We also stipulate that parents and their children can use these devices only in the office, school, and home. In order to provide the necessary security for these devices, we make the following security policies: (1) the TV set and the AC can be closed or opened either remotely or locally because of the low heat produced by such devices; (2) the RC can only be opened locally but can be closed remotely because of the high degree of heat that it produces; (3) the MW can be opened by parents either remotely or locally but cannot be opened by children remotely, because of the heat produced by the MW and the need to configure a time parameter, which can be performed only by parents; (4) because of the limitations regarding power load, the maximum number of devices that is able to run simultaneously is set to three; (5) every device only runs during the specified time.

9.1. Model. We integrate the components of STRAC as follows: $ID = \{0, 1, 2\}$ is a set of mobile terminal IDs based on the assumptions that (1) Terminal 0 is owned by children and is used in school and at home, (2) Terminals 1 and 2 are owned by parents and are used in multiple locations (school, office, and home), and (3) $REP = \{highRep, lowRep\}$, where $highRep$ and $lowRep$ denote high reputation and low reputation, respectively. We assume that the reputation of Terminal 0 is low and the others are high. $LT = \{TVtime, ACtime, MWtime, RCtime\}$ is a set of logic constraint times, for operating TV, AC, MW, and RC. $LLOC = \{home, office, school\}$, $OP = \{open, close, config\}$, $O = \{TV, AC, MW, RC\}$, and $PERM = \{open, close\} \times O \cup \{(config, MW)\}$; $AccZone$ is the product of REP , LT , and LOC . PA is a core component determined by control policies, as shown in Table 2. In Table 2, the product of the left and right columns of any row is an element of PA . For example, $\{(open, TV)\} \times REP \times \{TVtime\} \times LLOC \subseteq PA$. According to Table 2, $AccZoneAssigned$ and $permAssigned$ can be obtained (not discussed in this paper). For any access request, a decision can be made. For example, a child requesting to open MW in school will be denied.

9.2. Policy Verification. In the home applications discussed above, three kinds of entities exist: mobile terminals, RM (as shown in Section, RM denotes reference monitor), and the related devices (TV, AC, MW, and RC). In UPPAAL, these entities are denoted as *MobileTerm*, *RM*, and *Device*, respectively. In this scenario, mobile terminals send access requests to the RM. Upon receiving these requests, RM retrieves the stored access rule table, decides whether to

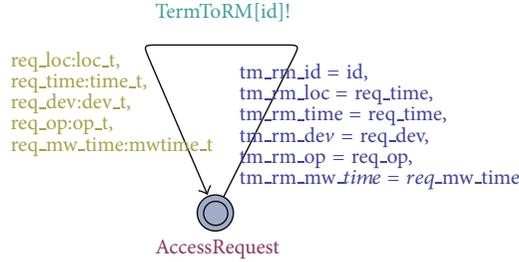


FIGURE 4: Timed automata of mobile terminals.

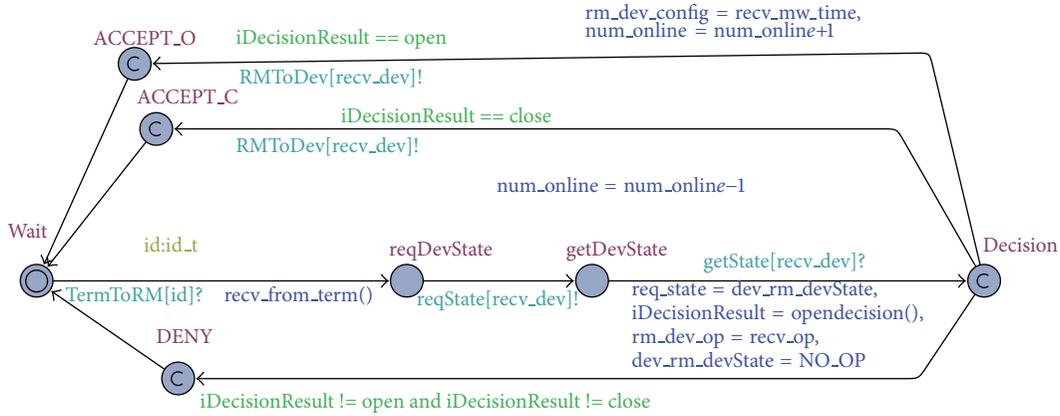


FIGURE 5: Timed automata of RM.

approve these requests, and sends the result to the controlled devices.

An access request consists of six parameters (as shown in Section 3.1, only three parameters are needed; however, in UPPAAL, we cannot get the current time; therefore, the *time* parameter is necessary. For simplicity, both *location* and *time* are provided here): *tm_rm_id* (ID of the mobile terminals), *tm_rm_loc* (location of the mobile terminals), *tm_rm.time* (current time), *tm_rm_dev* (target device), *tm_rm.op* (operation to perform on the target device), and *tm_rm.MW.time* (timer for *MW*). Figure 4 gives the timed automata of mobile terminals, where *id* is an input parameter, *TermToRM[id]* is a channel, and *loc.t* is a set of all locations (*home*, *school*, and *office*). The remaining variables are similar to *loc.t*.

As shown in Figure 5, *RM* receives access requests from the channel *TermToRM[id]*, checks the state (busy or free) of the controlled device, and makes a decision according to the reputation and location of the mobile terminals and the current time. This decision is then sent to the corresponding device.

Figure 6 illustrates the timed automata of controlled devices. The initial state of each device is *DevOff*. When the instruction *open* arrives, the corresponding device starts to run. If the instruction *close* arrives, the corresponding device stops running. In *MW*, *dev_config* is used to store the time parameter of the *MW* timer.

The composition (MobileTerm ||RM|| Device) of the above three timed automata forms a timed automation network, and we can use this network to verify policies. In our studies, three properties are verified: (1) “deadlock must be avoided in smart home applications.” This can be specified

as “A[] not deadlock” by using computation tree logic (CTL); (2) the number of simultaneously running devices is always less than or equal to three. This can be specified as “A[] not (Device(TV).DevOn and Device(AC).DevOn and Device(RC).Heating and Device(MW).DevOn)”; (3) when the parents open a microwave in office, the microwave would always switch to the *Ripe* state, which is “(tm_rm_id==1&&tm_rm_loc==office&& tm_rm.time==8&&tm_rm_dev==MW&& tm_rm.op== open) -->Device(MW).Ripe,” where the reputation of Terminal 1 (belonging to the parents) is high and the current time (*m_rm.time*==8) is within the work time of *MW*.

We use UPPAAL to verify the three properties, and the result shows that the first two properties are true. This means that the system is deadlock-free and that the number of simultaneously running devices is always less than or equal to three. Property (3) is proven to be false, meaning that, even when parents start the microwave while in the office, the microwave does not necessarily switch to the *Ripe* state. A counterexample is as follows: if parents start the microwave and the rice cooker at the same time in the office, the child will not be able to start the air conditioning and watch TV due to the power load limitation. In this case, children could stop the microwave, thus preventing it from switching to the *Ripe* state.

10. Experiments

We develop a prototype (shown in Figure 7) that implements STRAC. In our prototype, we use a network topology consisting of 10 terminal nodes (TeNs) uniformly deployed in

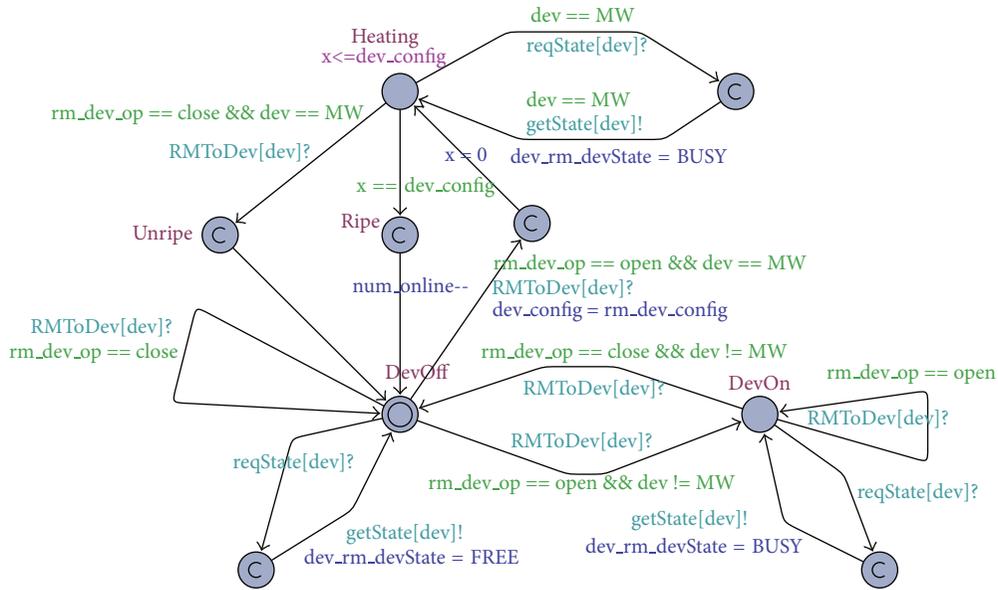


FIGURE 6: Timed automata of controlled devices.



FIGURE 7: STRAC prototype.

a \times area with TeNs surrounding a sink node (SiN), where Texas Instruments CC2530 chips with 8-KB RAM and 256-KB programmable flash are adopted for the TeNs and an ST Microelectronics STM32 chip with 64-KB RAM and 512-KB programmable flash was used for the SiN (CC2530 and STM32 chips are widely used as wireless transceivers and data processors in industry). The ZigBee protocol is used for communications between the TeNs and the SiN. The ARC component is mounted on the TeNs while RMC components are mounted on the SiN. ARC and RMC are written as traditional C programs.

Generally, heavyweight database servers, such as MySQL, are not appropriate for terminal nodes because of their limitations in storage and computing resources. To solve this problem, we implement a table query algorithm in our prototype and we also design two database tables: an ID-RTL

table that maps node ID to an RTL triple (the reputation of the node, the current logic time, and the current time location) and an access rule table that stores PA (as shown in Section 5, PA connects permissions with access zones). The ID-RTL table is integrated into the PEP components and is only accessible by TeNs and PDP, while the access rule table is mounted into PDP components and is only accessible by PEP. Our experiment setup is as follows: (1) reputation is set to 5 ratings, (2) TeNs could move into 100 different areas at 10 logical time periods, and (3) TeNs may perform 3 operations on 50 objects. After PA is explicitly defined, a series of experiments are conducted. The results reported in this section are averaged over 10 runs.

Experiment 1 (correctness of implementation). To evaluate correctness of implementation, we create two groups (A and B) of access requests in two local files: any request of Group A is in the access rule table; that is, any request from should be approved; all elements in Group are out of the access rule table; that is, any request from should be rejected. Groups A and B allow us to test whether our prototype works as anticipated. The experiment result shows that all requests from group A are approved and all requests from group B are rejected. This fact demonstrates the correctness of our implementation.

Experiment 2 (storage resources). In this experiment, two cases are considered in PA one is “given one permission, the number of its access zones is randomly generated” (called random scheme); the other is “given one permission, the number of its access zones is constant” (called constant scheme). The experiment result is as follows.

Random Scheme. If PA is compiled with the nonlattice strategy, 7500 access rules occupy 36.5 KB of programmable

flash; however, if the lattice strategy is adopted, only 3750 access rules are needed; as a result, 36.62 KB of programmable flash is consumed, which is about 100.3% of memory used by the nonlattice strategy.

Constant Scheme. If PA is compiled with the nonlattice strategy, 15000 access rules occupy 73.24 KB of programmable flash; however, if the lattice strategy is adopted, only 3900 access rules are needed; as a result, 30.48 KB of programmable flash is consumed, which is about 41.6% of memory used by the nonlattice strategy.

Note. In this experiment, PA is stored via a two-dimensional array. This means that for any two permissions, their consumed storages are the same and equal to $\text{sizeof}(\text{array})/\text{length}(\text{array})$. As a result, in the random scheme, the storage occupied by the lattice strategy is slightly greater than that of the lattice strategy. If PA is stored via files or pointers, the occupied storage will approach the constant scheme.

Experiment 3 (average response time, ART). To evaluate the response time of our scheme, we add a Group C of access requests: almost half of the access requests in Group C are in the access rule table. Each TeN continuously sends 1400 access requests to the SeN. The experiments show that the ART, which is the average delay from receiving a request to returning the response, is 21.4 ms if the nonlattice policy is adopted. The ART is reduced to 18.6 ms if the lattice policy is used, which is about 86.9% of the time used by the nonlattice strategy.

Experiments 2 and 3 show that the lattice policy is better overall than nonlattice policy and that the storage resource consumed and the response time of our model are acceptable for the sensing layer of the IoT.

11. Conclusions

The IoT presents new types of architectures, vulnerabilities, and requirements. Consequently, existing access control models have to be revised to accommodate these changes. Although spatiotemporal access models have been proposed in previous studies, most of them ignore some important characteristics of the sensing layer of the IoT. In this paper, we abstract the basic characteristics of the IoT's sensing layer and propose a model (called STRAC) that combines space and time with reputation for access control of the IoT's sensing layer. Our model solves the problem of deciding *when* and *where* to authorize access requests and *who* is able to access information by using spatial/temporal information, and it uses nondeterministic/stochastic authorizations to deal with unstable communications. These methods either provide better security or improve the QoS. In order to more precisely manage the reputation of nodes, we present a novel mechanism to update reputation and demonstrate its security. STRAC overcomes the inadequacies of existing access controls while acting as an access control foundation for the sensing layer of the IoT. In future work, we will design a scheme to find the optimal trade-off between security and QoS.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National High Technology Research and Development Program of China (Grant no. 2013AA014002), the National Natural Science Foundation of China (Grant nos. 61100181, 61100186, and 61262008), Guangxi Natural Science Foundation (Grant no. 2014GXNS-FAA118365), and the Key Project of Education Department of Guangxi.

References

- [1] T. Chen, "Stuxnet, the real start of cyber warfare?" *IEEE Network*, vol. 24, no. 6, pp. 2–3, 2010.
- [2] R. Sandhu, "The future of access control: attributes, automation, and adaptation," in *Computational Intelligence, Cyber Security and Computational Models*, vol. 246 of *Advances in Intelligent Systems and Computing*, p. 45, Springer, New York, NY, USA, 2014.
- [3] J. Park and R. Sandhu, "The UCON ABC usage control model," *ACM Transactions on Information and System Security*, vol. 7, no. 1, pp. 128–174, 2004.
- [4] A. Almutairi and F. Siewe, "CA-UCON: a context-aware usage control model," in *Proceedings of the 5th ACM International Workshop on Context-Awareness for Self-Managing Systems (CASEMANS '11)*, pp. 38–43, September 2011.
- [5] B. Fang, Y. Guo, and Y. Zhou, "Information content security on the internet: the control model and its evaluation," *Science in China F: Information Sciences*, vol. 53, no. 1, pp. 30–49, 2010.
- [6] K. Z. Bijon, K. Ram, and S. Ravi, "Constraints specification in attribute based access control," *Science*, vol. 2, no. 3, pp. 131–144, 2013.
- [7] F. Roesner, T. Kohno, A. Moshchuk, B. Parno, H. J. Wang, and C. Cowan, "User-driven access control: rethinking permission granting in modern operating systems," in *Proceedings of the IEEE 33rd Symposium on Security and Privacy (S and P '12)*, pp. 224–238, May 2012.
- [8] K. Sun, A. Liu, R. Xu, P. Ning, and D. Maughan, "Securing network access in wireless sensor networks," in *Proceedings of the 2nd ACM Conference on Wireless Network Security (WiSec '09)*, pp. 261–268, March 2009.
- [9] H.-F. Huang, "A novel access control protocol for secure sensor networks," *Computer Standards and Interfaces*, vol. 31, no. 2, pp. 272–276, 2009.
- [10] H. F. Huang and K. C. Liu, "A new dynamic access control in wireless sensor networks," in *Proceedings of the IEEE Asia-Pacific Services Computing Conference (APSCC '08)*, pp. 901–906, December 2008.
- [11] G. Zhang and M. Parashar, "Context-aware dynamic access control for pervasive applications," in *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, pp. 21–30, 2004.
- [12] L. Chen and J. Crampton, "On spatio-temporal constraints and inheritance in role-based access control," in *Proceedings of the ACM Symposium on Information, Computer and Communications Security (ASIACCS '08)*, pp. 205–216, March 2008.

- [13] S. Aich, S. Mondal, S. Sural, and A. K. Majumdar, "Role based access control with spatiotemporal context for mobile applications," in *Transactions on Computational Science IV*, vol. 5430 of *Lecture Notes in Computer Science*, pp. 177–199, Springer, New York, NY, USA, 2009.
- [14] M. Toahchoodee and I. Ray, "On the formalization and analysis of a spatio-temporal role-based access control model," *Journal of Computer Security*, vol. 19, no. 3, pp. 399–452, 2011.
- [15] R. Abdunabi, I. Ray, and R. France, "Specification and analysis of access control policies formobile applications," *IEEE Systems Journal*, vol. 7, no. 3, pp. 501–515, 2013.
- [16] J. Liu, Y. Xiao, and C. L. P. Chen, "Authentication and access control in the Internet of things," in *Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW '12)*, pp. 588–592, June 2012.
- [17] B. Anggorojati, P. N. Mahalle, N. R. Prasad, and R. Prasad, "Capability-based access control delegation model on the federated IoT network," in *Proceedings of the 15th International Symposium on Wireless Personal Multimedia Communications (WPMC '12)*, pp. 604–608, September 2012.
- [18] J. Á. M. Naranjo, P. Orduña, A. Gómez-Goiri, D. López-de-Ipiña, and L. G. Casado, "Enabling user access control in energy-constrained wireless smart environments," *Journal of Universal Computer Science*, vol. 19, no. 17, pp. 2490–2502, 2013.
- [19] S. Jha, N. Li, M. Tripunitara, Q. Wang, and W. H. Winsborough, "Toward formal verification of role-based access control policies," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 4, pp. 242–255, 2008.
- [20] C. A. Ardagna, S. de Capitani di Vimercati, S. Foresti, T. W. Grandison, S. Jajodia, and P. Samarati, "Access control for smarter healthcare using policy spaces," *Computers and Security*, vol. 29, no. 8, pp. 848–858, 2010.
- [21] D. Chen, G. Chang, D. Sun, J. Jia, and X. Wang, "Modeling access control for cyber-physical systems using reputation," *Computers and Electrical Engineering*, vol. 38, no. 5, pp. 1088–1101, 2012.
- [22] S. Misra and A. Vaish, "Reputation-based role assignment for role-based access control in wireless sensor networks," *Computer Communications*, vol. 34, no. 3, pp. 281–294, 2011.
- [23] L. Mui, A. Halberstadt, and M. Mohtashemi, "Notions of reputation in multi-agents systems: a review," in *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 280–287, July 2002.
- [24] E. Bertino, P. A. Bonatti, and E. Ferrari, "TRBAC: a temporal role-based access control model," *ACM Transactions on Information and System Security (TISSEC)*, vol. 4, no. 3, pp. 191–233, 2001.
- [25] J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "A generalized temporal role-based access control model," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 1, pp. 4–23, 2005.
- [26] H. Zhang, Y. He, and Z. Shi, "A formal model for access control with supporting spatial context," *Science in China F: Information Sciences*, vol. 50, no. 3, pp. 419–439, 2007.
- [27] S. Chong and A. C. Myers, "Security policies for downgrading," in *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS '04)*, pp. 198–209, October 2004.
- [28] G. Behrmann, A. David, and K. G. Larsen, "A tutorial on uppaal," in *Formal Methods For the Design of Real-Time Systems*, vol. 3185 of *Lecture Notes in Computer Science*, pp. 200–236, 2004.

Research Article

An Analysis of Security System for Intrusion in Smartphone Environment

Maya Louk,¹ Hyotaek Lim,² and HoonJae Lee²

¹ Department of Ubiquitous IT, Graduate School of Dongseo University, Sasang-Gu, Busan 617-716, Republic of Korea

² Division of Computer and Engineering Dongseo University, Sasang-Gu, Busan 617-716, Republic of Korea

Correspondence should be addressed to Maya Louk; mayalouk@gmail.com

Received 14 March 2014; Accepted 4 June 2014; Published 5 August 2014

Academic Editor: Fei Yu

Copyright © 2014 Maya Louk et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There are many malware applications in Smartphone. Smartphone's users may become unaware if their data has been recorded and stolen by intruders via malware. Smartphone—whether for business or personal use—may not be protected from malwares. Thus, monitoring, detecting, tracking, and notification (MDTN) have become the main purpose of the writing of this paper. MDTN is meant to enable Smartphone to prevent and reduce the number of cybercrimes. The methods are shown to be effective in protecting Smartphone and isolating malware and sending warning in the form of notification to the user about the danger in progress. In particular, (a) MDTN process is possible and will be enabled for Smartphone environment. (b) The methods are shown to be an advanced security for private sensitive data of the Smartphone user.

1. Introduction

Malware applications inhabit the application store and market. It does not only intrude via downloading or installing activity, but also intrude via access to particular website and SMS. Juniper research finds that 80% of Smartphone device will remain vulnerable for cyberattacks through 2013 [1]. This happens although there is an increasing in customer awareness toward the issue of mobile security products. According to Juniper, there are several factors upon the cause of low level of adoption for security products. It is expected that, by 2018, 1.3 billion mobile devices including smart phones, feature phones, and tablets are fortified by mobile security devices, up from around 325 million this year. According to the study by the Department of Homeland Security and the Federal Bureau of Investigation, as the dominant mobile operating system, android is the primary target for malware attacks because there are many users who are still using the older versions of the software [1]. According to the government agencies, 79 percent of the existing malwares are threatening android mobile system while the rest are haunting the other mobile systems [2].

The growth rate for threats targeting mobile platforms has increased dramatically: 40,059 of the 46,415 modifications and 138 of the 469 mobile malware families were added to our database in 2012 [3].

99% of mobile malware detections in 2012 were targeting android devices. For the next two years, it is clear that android will become the dominant target for malware attacks. Android operating system has become the most common operating system and the most interesting system to be attacked by malware-maker. The formula stands as follows: “the most prevalent OS” + “installation of software from any source” = “the greatest number of threats” [3].

Based on the research by Kaspersky lab and Juniper research, Figure 1 shows the Most targeted Mobile Operating System by intruders is Android (Figure 1(a)) and the most malware injected by intruders through Android is Trojan-SMS.AndroidOS.Opfake.bo (Figure 1(b)), this confirmed from the result in Table 1 and Table 2, where android hold the largest market share (Table 1) and the biggest threats modification by intruders. Based on this, we propose a new approach to analyze the behavior of malware in Smartphone. The idea will be running in android environment. The idea

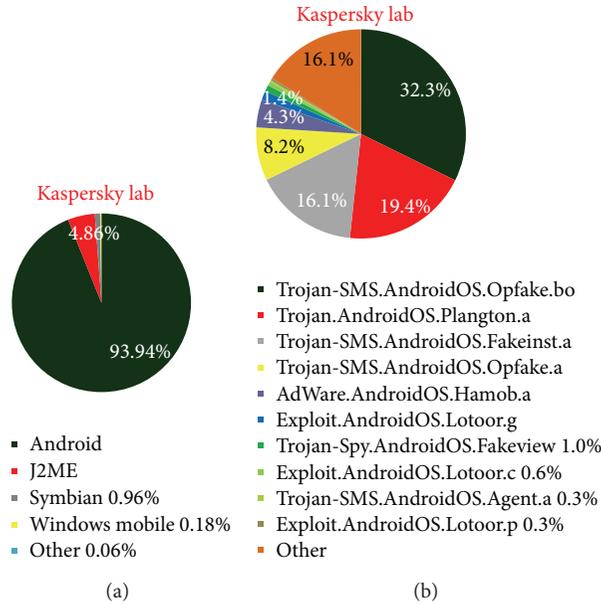


FIGURE 1: (a) Distribution of mobile threats by platform, 2004–2012; (b) the most frequently detected malicious programs targeting android.

TABLE 1: Market share among Smartphone OS-IDC worldwide quarterly mobile phone tracker, May 2013.

Operating system	2012 market share (%)	2013 market share (%)	Year over year change (%)
Android	59.1	75.0	79.5
iOS	23.0	17.3	6.6
Windows phone	2.0	3.2	133.3
Blackberry OS	6.4	2.9	-35.1
Linux	2.4	1.0	-41.7
Symbian	6.8	0.6	-88.5
Others	0.4	0.0	-83.3

TABLE 2: Modification of threats among Smartphone OS-IDC worldwide quarterly mobile phone tracker, May 2013.

Platform	Modification	Family
Android	43600	255
J2ME	2257	64
Symbian	445	113
Windows mobile	85	27
Others	28	10
Total	46415	469

consists of methods in monitoring, detecting malicious program inside the Smartphone, and tracking and notifying the user about the result and progress. Figure 2 illustrates how intruder works for repackaging a malware application process and also in Figure 3 illustrates the android installation file containing malware components to the mobile's user.

Based on this, MDTN is an interconnected process with two focuses. Malware application will be detected and any suspicious activity will be monitored in real time and notification will be sent to the user, all with the help of cloud computing system which is connected to the Smartphone for signature database. The outline of this work shows in Figure 4. The main contribution of this idea is methods in the form of MDTN which could be used by other researchers to track cyber intruders.

2. Related Work

There are a lot of researches about malware application up until 2013. Malware applications are being labeled (Kaspersky lab, Juniper research). Research about methods or species are also developed by several institutions (Cloud Security Alliance—CSA). According to CSA, malware could be deployed not only via website link, fake application, or smishing (SMS phishing), but also via Wi-Fi connectivity [4].

There are a lot of researchers who have been contributing ideas to improve security system to prevent data loss in mobile computing like Oliveira et al. via HoneyPotLabsac, a virtual honeypot for android which emulate intrusion detection on services like telnet, http, and SMS [5].

Some researchers provide their own security model [6–8]. The permission-based security model is one of the most important security models in android devices. The user could grant or deny the installation and the application itself specifies which resources of the device need to be used. Analysis and enforcement of this permission-based model have been proposed by various researchers [9, 10]. Burguera et al. [11] give a framework to detect malware on android platform. They monitor system call in Linux level and generate software behavioral patterns and classify these patterns by using

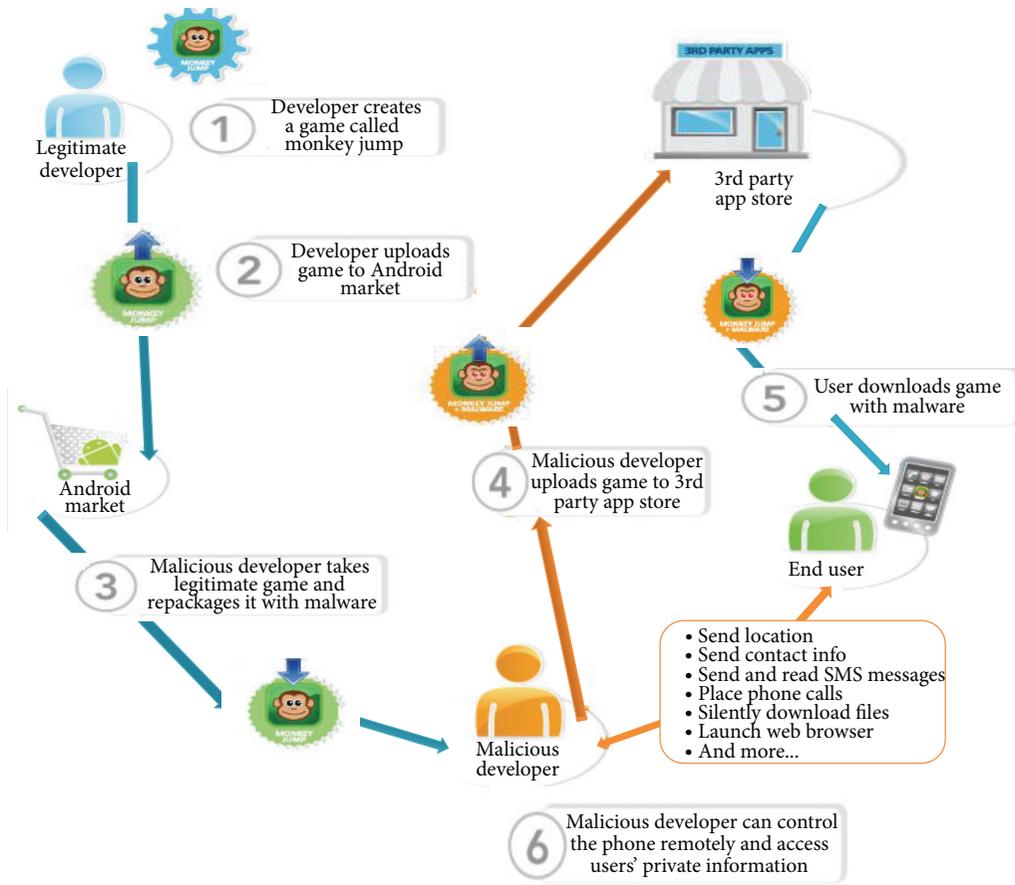


FIGURE 2: Repackaging applications process.

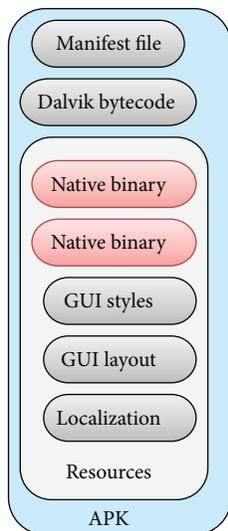


FIGURE 3: The android installation file containing malicious components.

cluster algorithm. Their method is efficient in detecting malware behavior seen from Linux kernel. Unfortunately there are several malware behaviors that cannot be seen from

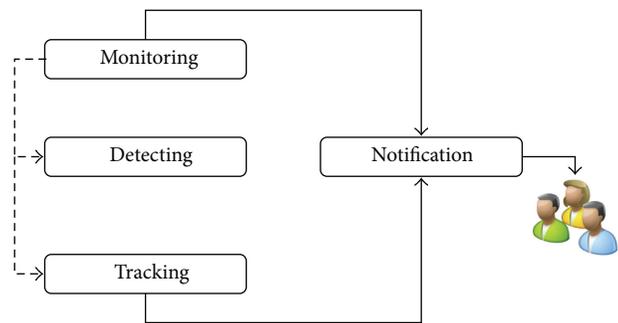


FIGURE 4: Outline of MDTN.

Linux level such as malicious SMS malware or malicious call malware. They are able to track a suspicious third-party application because they used dynamic analysis techniques to monitor sensitive information on android. The drawback of the system is that many normal applications may be considered as malware. Enck et al. presented TaintDroid in [12]. Their system used dynamic analysis techniques to monitor sensitive information on android. Thus, they can track a suspicious third-party application that uses sensitive data like GPS location information or address book

information. The shortcoming of their method is that an application with sensitive data may be considered as malware [8, 9, 12]. Lee et al. [13] have found that white list Smartphone environment contributes the idea of a white list server to store the identity of any application in the database so the server may recognize friendly application not as a malware. Infections are blocked by using reputation-based collected data and information. Marforio et al. [14] have been working on a coordinated attack against modern Smartphone system and this can lead to disclosure of user private data to third parties. Discussion about countermeasures can be used for protection against these kinds of attacks.

Metamorphic malware has become a subject in an ongoing and underdevelopment research from 2009. You and Yim have been contributing in the construction of malware obfuscation techniques. They explain a few general techniques in obfuscation malware [17].

3. Research Framework

The primary goal of this research study is to investigate the security risks associated with the use of android. It will contribute to regulate data in mobile computing on Smartphone, especially android mobile system. The proposed solution will detect attacks (viruses, worms, Trojan horses, and metamorphic malware) and prompt users to take actions to prevent breaches. Any suspicious activity that may reveal personal information to third parties or unknown entities will be reported to users to prevent potential attacks. This research study is different in that it will leverage previously proposed and implemented defense strategies and present an enhanced protection framework that will address android's vulnerabilities and risks. Furthermore, this project will extend the existing knowledge about android Smartphone's security and provide in-depth understanding of how to effectively manage emerging threats and fend off attacks, an issue that has long been realized and pointed out by security researchers and required more extensive research.

Android's threats are further amplified by the fact that users are limited to use their Smartphone for basic services and functions, such as email and SMS/MMS. Android's open-source nature further increases security vulnerabilities because cybercriminals could easily exploit this feature to modify the core applications and inserted malicious software to cause damage and monetary loss.

Research Question

- (i) What are the parameters to monitor, detect, and track?

From the result and related work which has already been done and still in progress, the possible parameters are to monitor all applications, through detection using static analysis code and signature behavior stored within the database that is controlled by cloud computational. By this scenario, using Smartphone to execute MDTN will become efficient though the drawback is to bypass malware because of

the unavailability of behavior and new signature outside of database.

- (ii) Has the MDTN process been approved to protect sensitive data from the Smartphone user?
A few ideas and implementation to detect malware have been executed, though the idea for detection using static behavior analysis that connected to cloud computational is still new. Based on research and text book about malware detection, MDTN process is logically possible and the mechanism has been executed well.
- (iii) How many malwares that could be anticipated using MDTN?
For the time being, only generally known malwares are recognized by behavior signature and static analysis code.
- (iv) Has MDTN fulfilled the security requirement about confidentiality, integrity, availability, authenticity, and accountability?
Confidentiality, integrity, availability, authenticity, and accountability are general requirements for security issue; thus MDTN has to fulfill these prerequisites.

4. Malware Behavior in Smartphone Environment

There are two methods of an intruder to steal data from Smartphone as follows.

- (i) Trojanized apps: cybercriminals will download an app from mobile store and then reupload the app into the app site with injected malicious malware.
- (ii) Malicious apps: cybercriminals will create malicious apps under the disguise of popular mobile app and upload them to the mobile store [18].

Vennon, a GTC engineer at Smobile Systems, has stated that malware is categorized based on what the malware does once it has infected a system. The categories are as follows [19].

- (i) Virus: a virus is defined as a destructive or malicious program that lacks the capacity to self-reproduce.
- (ii) Worm: this is a malicious code that can control system vulnerability or a network in order to automatically duplicate to another system.
- (iii) Trojan: a Trojan allows an attacker to obtain unauthorized access or remote access to a system while it appears to be executing a required operation.
- (iv) Spyware and adware: this destructive application conceals itself from the user while it collects information about the user without the user's permission.
- (v) Phishing apps: this malware is disguising itself as a legitimate site but containing mobile phishing that could steal user credential data. Malicious application is discovered by the user after installation and infection.

- (vi) Bot processes: hidden processes can execute completely invisible to the user, run executables, or contact botmasters for new instructions. Botnet strives to hijack and control infected devices.
- (vii) Mobile malware symptoms: signs of a malware infection can include unwanted behaviors and degradation of device performance. Performance issues such as frozen apps, failure to reboot, and difficulty connecting to the network are also common. Mobile malware can eat up battery or processing power, hijack the browser, send unauthorized SMS messages, and freeze or brick the device entirely.

Schmidt et al. [20] have announced the evolution of malware up to 2008. The malicious Linux binary itself is packed as “raw resource” into this Java application, for example, as png file, which can be seen on Figure 3. After installation, the Java application has to be executed once in order to rename the resource file into the appropriate binary. After renaming the file, the file has to be made executable which is currently impossible from within Java.

Malwares have various variants; one of them is metamorphic malware. The malware uses semantics-preserving transformations (obfuscations) to change its own code as it progresses. It progresses by repeating the computing process and applying the result of previous stage so the next stage will be different from the last. Any signature-based antivirus program will find it difficult to detect the malicious malware. Despite the ongoing changes, the function stays the same. The longer the malware stays, the more it evolves, making it difficult for the antivirus to defend the system. Obfuscation is to make the information less clear and more difficult to understand. Software vendors use obfuscation technique to prevent the software from reversing the engineer. Intruders use obfuscation transformations so the malware may never reverse the engineer and the malicious intent cannot be comprehended.

Obfuscated Code for Dead Code Insertion and Code Reordering. Consider

```

mov eax, [edx + 0Ch],
    jmp +3,
    push ebx,
    dec eax,
    jmp +4,
    inc eax,
    jmp -3,
    call Release Lock,
    jmp +2,
    push [eax],
    jmp -2.
    
```

(1)

“Dead code” is semantically equivalent to a nil operation. Insertion of this type of code has no semantic impact on the malware. The insertion increases the size of the malware and modifies the byte and instruction level content of the malware. “Code reordering” changes the syntactic order of the code in the malware. The actual or semantic execution path of the program does not change but only the syntactic order as present in the malware image. Code reordering includes the techniques of branch obfuscation, branch inversion, and branch flipping and the use of opaque predicates.

5. Proposed Idea and Design

The idea of this paper is to construct a proper android environment. Figure 6 illustrates the flowchart of the MDTN system contains Monitoring, detecting, tracking, and notification (MDTN) which is interconnected in this proposed idea.

5.1. Monitoring. Scanning all application and activity in Smartphone: the engine must examine and monitor various locations of the computer such as the hard disk, registry, and main memory. If a change to a critical component is detected, it could be a sign of infection.

Third-party applications are entrusted with several types of privacy sensitive information. The monitoring system must distinguish multiple information types, which requires additional computation and storage.

System activities include any action of interest which may be taken by the system, typically utilizing system resources. When integrated with system resource monitoring, these features can be used to study how activities impact system resource usage. When integrated with user activity monitoring, these features can be used to study how user activity impacts the system.

Monitoring system can also be used to continuously monitor features but only issue callbacks when certain conditions are met. These monitors will be referred to as notifies. The monitoring module will continuously monitor these features at the requested frequency but will only initiate a call to the callback function when the specified criteria are met. The format for such a request is similar to the monitor request, but with the additional information to specify the notification conditions. Monitoring System includes application and screen activities which listed in Table 3.

Context-based privacy sensitive information is dynamic and can be difficult to identify even when sent in the clear. For example, geographic locations are pairs of floating point numbers that frequently change and are hard to predict [12].

5.2. Detecting. A malware detector is a system responsible to determine whether a program has malicious behavior. In other words malware detector D is defined as a function: $D : A\{\text{Malware, Normal}\}$ where D is set for detecting and A is set for application. Consider

$$D(A) \begin{cases} \text{Malware} \\ \text{Normal.} \end{cases} \quad (2)$$

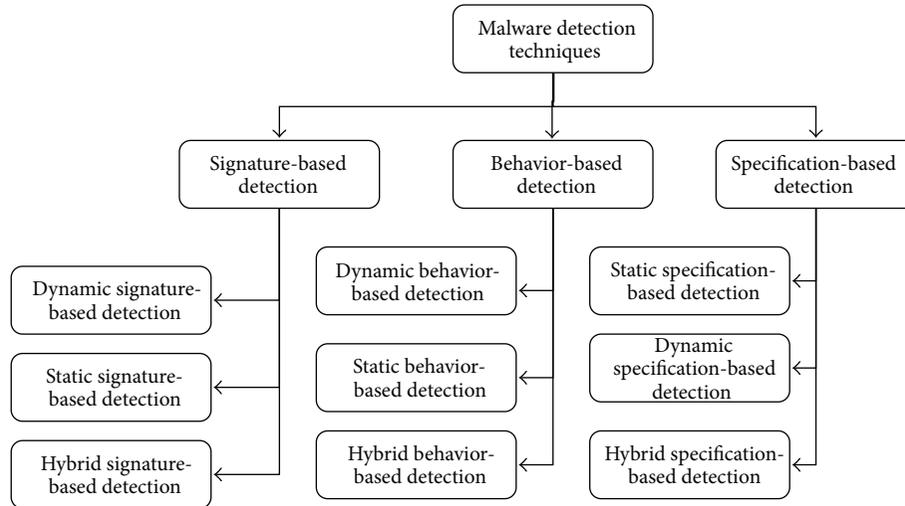


FIGURE 5: A classification of Smartphone malware detection techniques.

TABLE 3: Monitoring features for user activity.

Activity	Feature	Description
Application	appopen appclose	Open/close of application by user
Screen	screen	On/off screen

Detecting process will determine whether an application is a malware or legit through recorded behavior that is in line with library detector.

Generally, there are two techniques to detect malwares: anomaly-based technique and signature-based technique. Signature-based detection techniques define every known malwares by signature or particular patterns to identify malicious program. Anomaly-based detection techniques model normal behavior during a training phase and use this normal model to identify malicious programs.

Figure 5 illustrates the classification of malware detection techniques. In this classification, we followed the defined three rules. Reference behavior rule classified detection techniques broadly into two main categories: anomaly-based technique and signature-based technique. An anomaly-based detection technique constructs normal behavior model during the training phase. In detection phase any deviation from this model can be considered maliciousness [15].

This detection system is using behavior-based detection. This technique is a complex metastructure with dynamic concept and semantic interpretation. Behavior-based detection is effective and efficient to deal with complex techniques, such as polymorphic, binary packers, and encryption. This method is based on static code analysis which uses information embedded in a given executable file or code templates to capture the functionality of a specific malware. Behavior-based detection techniques assume that an intrusion can be detected by observing a deviation from normal or expected behavior of the system or the users. The model of normal or

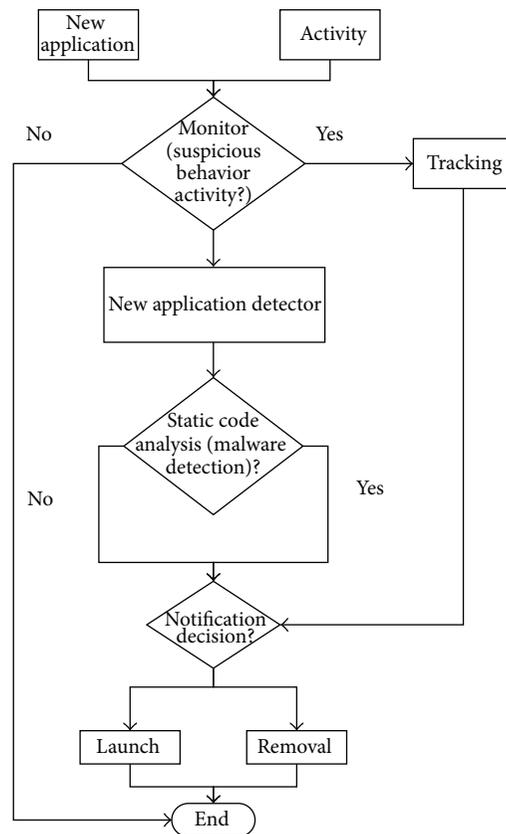


FIGURE 6: Flowchart of the system.

valid behavior is extracted from reference information collected by various means. The intrusion detection system later compares this model with the current activity. Advantages of behavior-based approaches are that they can detect attempts to exploit new and unforeseen vulnerabilities, the advantages and disadvantages of this method listed in Table 4.

TABLE 4: Advantages and disadvantages of behavior signature-based technique [15].

Advantages	Disadvantages
Fast and safe	Difficulty analyzing unknown malware
Low level of false positives	Cannot detect unknown malwares
Good in analyzing multipath malware	Not able to detect a lot of polymorphic viruses present (Packers)
Detect entire family of malware with one signature	
Detect malware before its execution	
Best results in detecting of polymorphic malware	

Once the engine has detected an item that requires further examination, the engine will refer to an updated list of known malware, called the “blacklist”. The blacklist contains “signatures” or identifiable patterns of know malware. The engine will be able to determine whether any file matches any of the known malware. If a match is identified, the file is classified according to the particular category: Malware’s integrity identification, eradication of the particular packages, publication of the list of packages to the remote server, and so forth [21]. One of the methods to perform detection is

$$\begin{aligned} & \text{PackageInfo class: signatures () ,} \\ & \text{getInstalledPackages () .} \end{aligned} \quad (3)$$

Removal. The final step for this engine is to take appropriate actions on files that are identified as malware. In most circumstances, the engine removes the program or file completely and restores the computer to its ore-infection state. Otherwise, a file can be disabled or quarantined, so that the user could enable it later.

For Metamorphic Malware There Is an Interpretation for Obfuscating Solution. Abstract interpretation declared in 1977 is a general model for the (static or dynamic) approximation of semantics of discrete dynamic systems. Obfuscating programs is making abstract interpreters incomplete. Modifying the simple self-interpreter so that all values in the store are obfuscated. Algorithm 1 illustrates Formal framework for malware detection itself are based on program semantics and abstract interpretation. This is as follows for obfuscation interpretation in metamorphic malware.

The following are the obfuscation techniques that are particularly used by metamorphic viruses [22]:

- (i) data flow obfuscation (instruction substitution, instruction permutation, *dead code* or *garbage code* insertion, variable substitution);
- (ii) control flow obfuscation (changing the control flow).

Modification of interpretation for obfuscation code is as follows.

- (i) Input values are obfuscated in the initial store.
- (ii) Variable values are obfuscated just before putting in the store.

(iii) Output values are deobfuscated in the program’s final store.

- (iv) Expression evaluation yields nonobfuscated values:
- (a) constant values are not obfuscated;
 - (b) variables’ values must be deobfuscated when got from the store.

Let $\rho \in uco(\Sigma)$ with Σ semantic objects (data, traces, etc.)

A program transformation $\tau : \mathbb{P} \rightarrow \mathbb{P}$ such that $\llbracket \mathbb{P} \rrbracket = \llbracket \tau(\mathbb{P}) \rrbracket$

ρ β -complete for $\llbracket \cdot \rrbracket$ if $\rho(\llbracket \mathbb{P} \rrbracket) = \llbracket \mathbb{P} \rrbracket^\rho$

$$\begin{aligned} & \tau \text{ obfuscates } P \text{ if } \llbracket P \rrbracket^\rho \sqsubset \llbracket \tau(P) \rrbracket^\rho \\ & \llbracket P \rrbracket^\rho \sqsubset \llbracket \tau(P) \rrbracket^\rho \Leftrightarrow \rho(\llbracket \tau(P) \rrbracket) \sqsubset \llbracket \tau(P) \rrbracket^\rho \end{aligned}$$

$$\mathcal{D}(P, M) = \begin{cases} \text{true,} & \text{if } \mathcal{D} \text{ determines that } P \\ & \text{is infected with } M, \\ \text{false,} & \text{otherwise.} \end{cases} \quad (4)$$

Consider a set \circ of obfuscating transformations ranged over by \circ .

Let $M \hookrightarrow P$ denote that program P is infected with malware M .

\mathcal{D} is sound for \circ if $\mathcal{D}(P, M) = \text{true} \Rightarrow \exists \circ \in \circ : \circ(M) \hookrightarrow P$.

\mathcal{D} is complete for \circ if $\forall \circ(M) \hookrightarrow P \Rightarrow \mathcal{D}(P, M) = \text{true}$,

where an ideal malware detector is sound and complete: sound means no false positives and complete means no false negatives.

Certifying Malware Detecting. We can characterize the most concrete property ϕ such that

$$MD\rho(M, P) = \text{true} \Leftrightarrow \exists T \in \text{Progr} : \llbracket \mathfrak{S}(M, T) \rrbracket^\rho = \llbracket P \rrbracket^\rho \text{ is sound and complete for } \circ_\phi.$$

Training Malware Detecting. Given \circ_ϕ we can characterize the most concrete property ρ such that MD is complete for \circ_ϕ .

Input P, d;	Program to be interpreted, and its data
pc := 2;	initialize program counter and obfuscated store:
store := [in ↦ obf(d), out ↦ obf(o), x ₁ ↦ obf(o), ...];	
While pc < length(P) do	
instruction := lookup(P, pc);	
case instruction of	Dispatch on syntax
skip : pc := pc + 1;	Obfuscate values when stored:
x := e : store := store[x ↦ obf(eval(e, store))]; pc := pc + 1;	
...	
Output dob(store[out]);	
obf(V) = 2 * V; dob(V) = V/2	Obfuscation/de-obfuscation
eval(e, store) = case e of	
constant : obf(e)	
variable : dob(store(e))	De-obfuscate variable values
e1 + e2 : eval(e1 + store) + eval(e2, store)	
e1 - e2 : eval(e1 + store) - eval(e2, store)	
...	

ALGORITHM 1

5.3. *Tracking.* Tracking is a phase where the application will track the source of the problem and perform the tracking activity in the Smartphone.

The following could be logged to represent a user:

- (i) Secure.ANDROID_ID (has limitations),
- (ii) TelephonyManager:
 - (a) getSimSerialNumber(),
 - (b) getDeviceID(),
- (iii) Build.Serial (good for tablets),
- (iv) Company-assigned ID.

The process for data tracking is started from detected tainted source or suspicious behavior. Tainted data comes from specific source; thus the contaminated data shall be tracked down specifically and dealt with. After the purging process is done, the result will be reported that the decontamination has been finalized.

To track the URL location of the intruder who remotely controls the malware inside the Smartphone, the following method will be used:

$$\begin{aligned}
 &Get_Document () \\
 &Get_LocationURL () \qquad (5) \\
 &Get_LocationName ().
 \end{aligned}$$

The device's IMEI was also exposed by applications. The IMEI uniquely identifies a specific mobile phone and is used to prevent a stolen handset from accessing the cellular network. TaintDroid flags indicated that nine applications transmitted the IMEI.

Seven out of the nine applications either do not represent an end user license agreement (EULA) or do not specify IMEI collection in the EULA [12]. From this result, tracking IMEI and IMSI activity should be made known to users to let them determine which activity is remotely controlled by intruders.

The method for IMEI and IMSI (personal information) is as follows:

Methods of the TelephonyManager class :

$$\begin{aligned}
 &get_DeviceId (), getSubscriberId (), \\
 &getNetworkOperator (), \\
 &getLine1Number (), getSimOperator (), \qquad (6) \\
 &getSimSerialNumber (), \\
 &getSimCountryIso ().
 \end{aligned}$$

5.4. *Notification.* The action is defined by a PendingIntent containing an Intent that starts an activity in your application. To associate the PendingIntent with a gesture, call the appropriate method of NotificationCompat.Builder.

A PendingIntent object helps to perform an action on the application's behalf, often at a later time, without caring about whether or not the application is running. After the action is performed, NotificationManager.notify() is called to pass the notification object to the system by sending the particular task.

The method for getting notification is as follows:

$$\begin{aligned}
 &NotificationCompact.Builder.build () \\
 &NotificationManager.notify () \qquad (7) \\
 &Android:name \\
 &= "android.support.PARENT_ACTIVITY".
 \end{aligned}$$

6. The Design System of MDTN and Discussion

The MDTN system is an interconnected process to monitor the downloading and installation progress of any file in a Smartphone. If in the case of suspicious behavior detected,

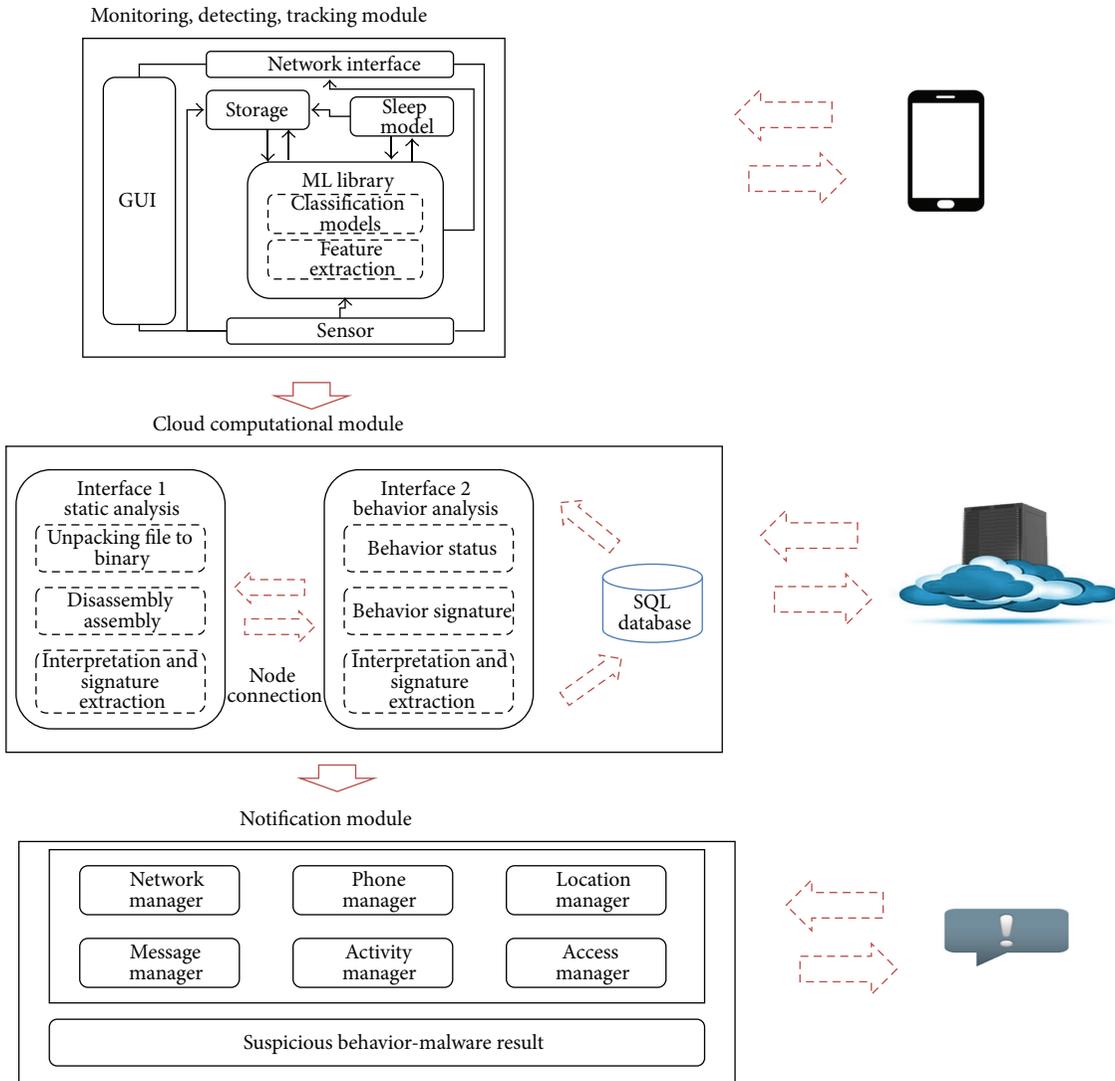


FIGURE 7: The MDTN system infrastructure.

the tracking module will be deployed to track the source of the behavior, the result will be forwarded to the notification module for the next decision. A new application which is going to be installed or web application with malware will be monitored and detected using static code analysis and signature database will determine whether the file contains any malware or not. The overall progress will be notified to the user, and the user may decide whether or not to install or to delete the given application. Figure 7 illustrates 3 modules inside MDTN system infrastructure.

There are 3 modules inside MDTN system. The first and the third module (module notification) are connected to the user, while the monitoring, detecting and tracking parts which consisted of classification models and extraction are featured inside the machine learning. They combine an internally developed platform-independent machine learning C library with specific components device—written in Java—which are responsible for communication, storage and user interface. The process is the classification pipeline which

is responsible for the inference of end user behavior. The pipeline continuously samples the phone sensors and extracts features used by classification models, which also run on the phone. The classification pipeline samples one sensor, GPS. All these processes are connected with the user’s smartphone.

The second module is in-between or middleware module that could reduce Smartphone’s performance. Though, it has its own disadvantage when it comes to delivery process into the cloud server where the server computes the data and consumes more time to redeliver back to the user. All data is stored within independent SQLite files. These files are transferred to the cloud infrastructure with an uploading policy that emphasizes energy efficiency to minimize the impact of using the phone’s batteries.

For this case we build our own Private Cloud Eucalyptus which was bundled with Ubuntu (UEC-Ubuntu is bundling OpenStack from 11.10). UEC/Eucalyptus is an on-premise private cloud OSS based platform, sponsored by Eucalyptus Systems, Linux based—RHEL, CentOS, Ubuntu, Support for

TABLE 5: Mobile malware for this research.

Name	Package's name	% of attacks
Trojan-SMS	Android SMS Trojan, jSMShider	33.5
Backdoor	Obad	20.6
Trojan	Crazyapps angry.birds, Beauty.Girl-1	19.4
Adware	Airpush-Minimob	7.1
RiskTool	Mobile Spy	6.0
Trojan-Downloader	Trojan.Extension	5.8
Trojan-Spy	Andr/PJApps	4.0
Others apps	DroidKungFu, Zitmo android, GoldDream	3.6

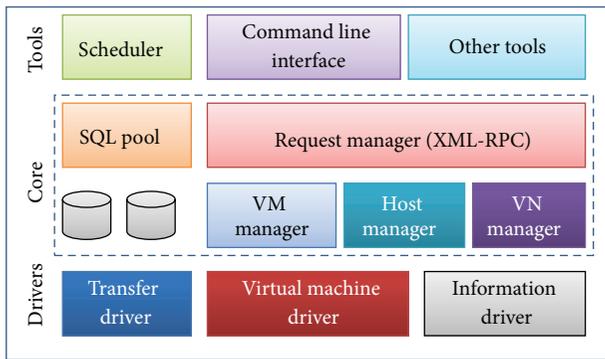


FIGURE 8: The three layers of the private cloud internal architecture.

VMware. Figure 8 illustrates the 3 layers inside the private cloud internal architecture that used in this work.

7. Performance Analysis

A standardized measurement should first be set to test and evaluate the performance in the two systems in which total time execution is used. Basically, throughput is a decent indicator of malware analyzer performance. Within a period of time, throughput will calculate the total number of completed analysis task. Thus, the total consumed time is used in which the defined samples affect the total number submitted in a fixed number. The time is calculated by summing the in-between period of the last analysis sample and the previous analysis time within the same group sampling.

Total execution time $T(n)$ for task $D(P, M)$ consists of three elements: setup time (t_s), execution time (t_e), and postprocessing time (t_p). Setup time is the total time to prepare and to deploy the required accessible malware sample. The time in the second system will become the connector as it requires longer time comparing to the first system. Nevertheless, the execution and postprocessing time itself will be the same in the first and the second systems. The setup time itself is a drawback to the second system, thus giving advantage to the first system which practically uses cloud computing. The required records from each tasks needed to calculate $T(n)$ are submit time, start analysis time, and finish time.

TABLE 6: System performance result.

Virus source	System performance		
	Detection	Memory remains (%)	Time (s)
Android SMS Trojan	Yes	85.69	5.24
jSMShider	Yes	92.63	3.50
Obad	Yes	79.61	6.26
crazyapps.angry.birds	Yes	87.38	4.25
com.Beauty.Girl-1	Yes	89.45	4.26
Airpush-Minimob	Yes	78.26	6.02
Mobile Spy	Yes	86.24	5.63
Trojan.Extension	Yes	78.86	6.24
Andr/PJApps	Yes	95.25	5.23
Zitmo android	Yes	84.12	4.35
DroidKungFu	Yes	93.14	4.58
GoldDream	Yes	80.21	4.53

7.1. Mobile Malware List (Table 5). This implementation is going to be executed on android OS platform using Samsung Galaxy S3, android version 4.3, IMEI number 352905053490342. This research uses 8 of biggest attack percentage mobile software, and this research uses 12 different types of mobile software.

7.2. System Performance Evaluation (Table 6). 12 types of mobile software (.APK) under trial are successfully detected as malware by the system on this system performance. Stabilized time (ms) is the time needed for the system to recognize the mobile software as a malware and the notification process via executable binary file. Total stabilized time $T(n)$ consists of three elements: setup time (t_s), execution time (t_e), and postprocessing time (t_p).

7.3. Tracking System Performance (Table 7). The tracking phase is not maximally carried out, as the result services of Zitmo android do not show and GoldDream's server is not detected. Under services, any service that is going to be used by the intruders can be tracked (data stealing). From the tracking result, user is able to know which of the many services on android system is under monitoring or modification, or under threat from the particular server. The tracking has not been maximally carried out by scrutinizing

TABLE 7: Tracking result.

Name	Tracked server	Services
Android SMS Trojan	✓	Phone Number, IMEI, IMSI
jSMShider	✓	SMS, MMS transaction
Obad	✓	MMS transaction, admin system
crazyapps.angry.birds	✓	Music MediaPlayer, MMS transaction
com.Beauty.Girl-1	✓	Music MediaPlayer, MMS transaction
Airpush-Minimob	✓	IMEI, MMS transaction
Mobile Spy	✓	MMS transaction, MediaPlayer
Trojan_Extension	✓	IMEI, IMSI, MMS transaction, MediaPlayer, CoreService, EmailService, CheckingService, bluetooth, SMSReceiver
Andr/PJApps	✓	Music MediaPlayer, system security
Zitmo android	✓	—
DroidKungFu	✓	Work service, MMS, email
GoldDream	—	MMS transaction

TABLE 8: Comparison performance.

Characteristic	TaintDroid [12]	CrowDroid [11]	RobotDroid [16]	MDTN
Detection techniques	Anomaly detection	Behavior based-dynamic analysis	Support vector machine active learning algorithm	Behavior based-static analysis
Operating system	TaintDroid	Android	Android	Android
Kernel level	Yes	No	No	No
Real time monitoring	Yes	No	No	Yes
Tracking system	Yes	No	No	Yes

any server that serves the intruders, and so far tracking depends on intruder's known website. In the future, the malware's server can be reported to the database system so that the particular application can be blocked.

7.4. Comparison System Performance (Table 8). On comparison with the 3 previous projects that have been carried out and developed—like TaintDroid, CrowDroid, and RobotDroid—real time monitoring and tracking system is provided by TaintDroid by which the kernel level is exercised. TaintDroid is an android operating system with added real time monitoring and tracking system.

8. Conclusion and Future Work

MDTN is an interconnected system process for Smartphone environment. For this research paper, the author uses Android OS because the operating system is frequently attacked by cybercriminals. Monitoring, detecting, tracking, and notification are used not only to check new application before being installed into the Smartphone, but also to detect suspicious behavior activity in real time. As for the detection method, behavior-based detection technique and database static code analysis are used to determine suspicious behavior and malware application. The tracking part can be developed

to later stage for the purpose of preventing future threat realization. In the case of reoccurrence, the system is able to block and recognizing the data as spam or threat.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the National Research Foundation of Korea under Grants 2011-0009349 and BK21.

References

- [1] Juniper research, Press Release: More than 90% of Smartphones Remain Unprotected from Malware and Attacks, Juniper Research Finds, 2014, <http://www.juniperresearch.com/view-pressrelease.php?pr=404>.
- [2] IDC Worldwide Quarterly Mobile Phone Tracker, *Android and iOS Combine for 92.3% of All Smartphone Operating System Shipments in the First Quarter While Windows Phone Leapfrogs BlackBerry*, According to IDC, 2013, <http://www.idc.com/getdoc.jsp?containerId=prUS24108913>.

- [3] M. Denis, "Mobile Malware Evolution: Part 6," 2014, http://www.securelist.com/en/analysis/204792283/Mobile_Malware_Evolution_Part_6?print_mode=1.
- [4] Cloud Security Alliance, *Top Threats to Mobile Computing*, Cloud Security Alliance, 2012.
- [5] V. B. Oliveira, Z. Abdelouahab, D. Lopes, M. H. Santos, and V. P. Fernandes, "HoneyPotLabsac: a virtual honeypot framework for android," *International Journal of Computer Networks & Communications*, vol. 5, no. 4, p. 159, 2013.
- [6] W. Enck, M. Ongtang, and P. McDaniel, "Understanding android security," *IEEE Security and Privacy*, vol. 7, no. 1, pp. 50–57, 2009.
- [7] W. Enck, D. Ocateau, P. McDaniel, and S. Chaudhuri, "A study of android application security," in *Proceedings of the 20th USENIX Security Symposium*, USENIX Association, August 2011.
- [8] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev, and C. Glezer, "Google android: a comprehensive security assessment," *IEEE Security and Privacy*, vol. 8, no. 2, pp. 35–44, 2010.
- [9] M. Ongtang, S. McLaughlin, W. Enck, and P. McDaniel, "Semantically rich application-centric security in android," in *Proceedings of the 25th Annual Computer Conference Security Applications (ACSAC '09)*, pp. 340–349, IEEE Computer Society, Los Alamitos, CA, USA, December 2009.
- [10] G. D. Bai, L. Gu, T. Feng, Y. Guo, and X. Q. Chen, "Context-aware usage control for android," in *Security and Privacy in Communication Networks*, vol. 50 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 326–343, Springer, Berlin, Germany, 2010.
- [11] I. Burguera, U. Zurutuza, and S. NadjmTehrani, "Crowdroid: behavior-based malware detection system for android," in *Proceedings of the ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '11)*, Chicago, Ill, USA, 2011.
- [12] W. Enck, P. Gilbert, B.-G. Chun et al., "TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones," in *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation (OSDI '10)*, pp. 1–6, USENIX Association, Berkeley, Calif, USA, 2010.
- [13] K. Lee, R. S. Tolentino, G.-C. Park, and Y.-T. Kim, "A study on architecture of malicious code blocking scheme with white list in smartphone environment," in *Communication and Networking*, vol. 119 of *Communications in Computer and Information Science*, pp. 155–163, Springer, Berlin, Germany, 2010.
- [14] C. Marforio, A. Francillon, and S. Capkun, "Application collusion attack on the permission-based security model and its implications for modern smartphone systems," Tech. Rep., Department of Computer Science, ETH Zurich, Zürich, Switzerland, 2011.
- [15] A. Amamra, C. Talhi, and J. Robert, "Smartphone malware detection: from a survey towards taxonomy," in *Proceeding of the 7th International Conference on Malicious and Unwanted Software (Malware '12)*, pp. 79–86, Fajardo, Puerto Rico, USA, October 2012.
- [16] M. Zhao, T. Zhang, F. Ge, and Z. Yuan, "Robotdroid: a lightweight malware detection framework on smartphones," *Journal of Networks*, vol. 7, no. 4, pp. 715–722, 2012.
- [17] I. You and K. Yim, "Malware obfuscation techniques: a brief survey," in *Proceedings of the 5th International Conference on Broadband Wireless Computing, Communication and Applications (BWCCA '10)*, pp. 297–300, November 2010.
- [18] V. Manjunath, "Reverse Engineering of Malware on Android," SANS Institute InfoSec Reading Room, 2011.
- [19] T. Vennon, "Android malware. A study of known and potential malware threats," SMobile Global Threat Centre, 2010.
- [20] A. Schmidt, H. Schmidt, L. Batyuk et al., "Smartphone malware evolution revisited: android next target?" in *Proceedings of the 4th International Conference on Malicious and Unwanted Software (MALWARE '09)*, pp. 1–7, October 2009.
- [21] A. Mujumdar, G. Masiwal, and B. B. Meshram, "Analysis of signature-based and behavior-based anti-malware approaches," *International Journal of Advanced Research in Computer Engineering and Technology*, vol. 2, no. 6, pp. 2037–2039, 2013.
- [22] J. Borello and L. Mé, "Code obfuscation techniques for metamorphic viruses," *Journal in Computer Virology*, vol. 4, no. 3, pp. 211–220, 2008.

Research Article

SmartMal: A Service-Oriented Behavioral Malware Detection Framework for Mobile Devices

Chao Wang,¹ Zhizhong Wu,¹ Xi Li,¹ Xuehai Zhou,¹ Aili Wang,² and Patrick C. K. Hung³

¹ Department of Computer Science, University of Science and Technology of China, Hefei 230027, China

² School of Software Engineering, University of Science and Technology of China, Suzhou 215123, China

³ Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON, Canada

Correspondence should be addressed to Xi Li; llxx@ustc.edu.cn

Received 14 March 2014; Accepted 2 June 2014; Published 5 August 2014

Academic Editor: Fei Yu

Copyright © 2014 Chao Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents SmartMal—a novel service-oriented behavioral malware detection framework for vehicular and mobile devices. The highlight of SmartMal is to introduce service-oriented architecture (SOA) concepts and behavior analysis into the malware detection paradigms. The proposed framework relies on client-server architecture, the client continuously extracts various features and transfers them to the server, and the server's main task is to detect anomalies using state-of-art detection algorithms. Multiple distributed servers simultaneously analyze the feature vector using various detectors and information fusion is used to concatenate the results of detectors. We also propose a cycle-based statistical approach for mobile device anomaly detection. We accomplish this by analyzing the users' regular usage patterns. Empirical results suggest that the proposed framework and novel anomaly detection algorithm are highly effective in detecting malware on Android devices.

1. Introduction

Personal digital assistants (PDAs), mobile phones, and recently smartphones have evolved from simple devices into sophisticated yet compact minicomputers which can connect to a wide spectrum of networks, including the Internet and corporate intranets. Designed as open, programmable, networked devices, smartphones are susceptible to various malware threats such as viruses, Trojan horses, and worms, all of which are well known from desktop platforms. These devices enable users to access and browse the Internet, receive and send emails, and short message service (SMS), connect to other devices for exchanging/synchronizing information, and install various applications, which make these devices ideal attack targets [1].

Above all, mobile devices have become popular companions in people's daily life, as is illustrated in Figure 1. It allows users to access news, entertainment, carry out research, or make purchases via e-businesses. Unfortunately, cyberspace is a double-edged sword; the new malware and viruses appearing on mobile devices have dramatically impacted the safety and security of users; this side effect of Internet access

has become a serious problem. According to the Internet Filter Reviews statistics [2], the amount of malware detected is each year the double. In particular, there are at least 7.12 million smartphones that have been infected by various malware and virus.

The challenges for smartphone security are becoming very similar to those that personal computers encounter and common desktop security solutions are often being downsized to mobile devices. Unfortunately, the increasing popularity smartphones and their ability to run third-party software have also attracted the attention of virus writers [3, 4]. Malware can make a smartphone partially or fully unusable, causing unwanted billing; stealing private information, and so on. If we have the ability to detect the attack as soon as it occurs, we can stop it from doing any damage to the system or personal data. This is where an intrusion detection system comes in, there are two types of intrusion detection systems: signature-based and anomaly-based systems. Signature-based approaches can only detect existing malwares and require frequent signature updates to keep the signature database up-to-date. Signature-based systems are often used for antivirus software on desktop

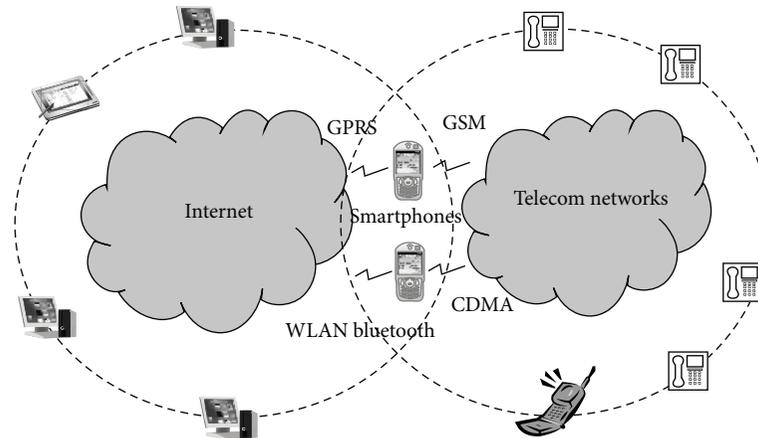


FIGURE 1: Mobile devices have become a common place for both Internet and telecom networks. They have been combined into a sound framework which allows different media to communicate with each other immediately and efficiently.

systems. Researchers are trying to develop anomaly based approaches which can detect unknown malwares.

Recently, behavior-based programming has been proved [5] to be an efficient way to detect abnormal utilizations to formalize requirements in the form of use cases and scenarios. It has also been introduced to the malware detection mechanism [1]. However, the behavior analysis technique is worth pursuing, it still poses significant challenge to clearly identify behaviors for distinct embedded applications.

In order to solve this problem, we will demonstrate the effectiveness of service-oriented architecture (SOA) in browser design. Traditionally, SOA provides effective measures with better flexibility and extensibility at lower cost by adopting reusable software modules. SOA can also reduce the complexity of integration and application development through uniform service description and integration interfaces. Therefore, SOA-based design is more convenient when building systems by providing a common way for interaction and communication.

By the exploration of benefits of SOA concepts, we can conclude that there are at least two significant advantages of integrating SOA concepts into malware detections. Firstly, it can greatly reduce the local workload of the detection algorithm. This feature allows users to run a light-weight client which works especially well for mobile devices, because all the processing threads will run on the servers. Secondly, the user behavior analyses, such as CPU/memory utilization, battery endurance, and network traffic flow, are located on central or distributed servers. This improves the load balancing status for global optimization. Finally, with the back-end management module (e.g., web pages), the malware information is easier to be kept up-to-date and be distributed to clients for real time synchronization.

This paper proposes service-oriented malware detection with distributed behavior analysis mechanisms for the first time, called SmartMal. The paper is extended from the previous publication at [6]. The contributions of this paper are listed as follows. The paper starts by describing the SOA-based malware with distributed detection algorithm.

The abnormal messages and irregular behaviors are provided through services. Secondly, this paper proposes and realizes a behavior analysis algorithm with SOA concepts. We integrate distributed components into a hierarchical kernel model. Diverse optimizing measures are taken to exploit the battery endurance, CPU/memory utilization and network traffic flow. Experimental results are presented to demonstrate the effectiveness of SmartMal.

The rest of the paper is organized as follows. Related work and motivation are summarized in Section 2. Section 3 discusses the architecture and main concepts of SmartMal, including architecture, client organization, and server hierarchical models. Section 4 describes the behavior analysis model. Section 5 demonstrates the SmartMal with a typical case: DoS attacks. Finally, conclusion and further work are presented in Section 6.

2. Related Work and Motivation

Safety and security problems on mobile devices have been a major focus in the past decade. In this section, we present a review on general malware detection techniques for mobile devices.

There has been a considerable amount of research into anomaly detection in computing systems and network traffic. These include statistics-based approaches, data-mining based methods, and machine learning based techniques. A wide set of anomaly detection approaches on smartphones are built from the above techniques.

Statistical-based approaches were originally used in anomaly detection on smartphones. Cheng et al. [7] propose a collaborative virus detection and alert system for smartphones where smartphones run a light-weight agent then collect and report information to a proxy. The proxy detects viruses through a statistical approach, and it keeps track of the average number of communications. Buennemeyer et al. [8] present a scheme that monitors abnormal changes of smartphones using smart batteries.

Bose et al. [9] propose a novel behavior-based detection framework for smartphones. The behavior signatures are constructed at run time by monitoring the events and API calls via Proxy DLL. They use support vector machines (SVMs) to train a classifier from normal and malicious data. The evaluation results show that the scheme can identify current mobile malwares with more than 96% accuracy. A distributed SVM algorithm is presented in [10]; with the distributed scheme, the participating clients perform their computation in parallel and update the support vectors simultaneously, so the overhead of machine learning algorithm is efficiently decreased.

Schmidt et al. [11] present programs that monitor smartphones running Symbian and Windows Mobile OS. They demonstrate that only a few features are needed to achieve acceptable detection performance. Machine learning methods, like artificial immune system (AIS) and self-organizing maps (SOM), are applied to detect abnormal behavior on remote server, and they proposed an algorithm called linear prediction to detect change by checking four predecessors of a chosen feature. In [12], they present a novel approach to detect malware, where function calls are extracted from binaries. The centroid machine classifies an executable via clustering, in which each cluster is defined by a centroid. That is, a binary is classified as malicious if it is closer to malicious cluster, naming the distance metric as Euclidean distance.

Game theory has been introduced into the anomaly detection area of mobile phone. Shabtai et al. [1] propose a light-weight malware detection system for Android smartphone, and they developed four malicious applications for experiment. Several usual classification algorithms and feature selection algorithms are evaluated to find the best performance in these detection systems. Alpcan et al. [13] present a novel probabilistic diffusion scheme for anomaly detection based on mobile device usage patterns. The scheme models the normal behavior and their features as a bipartite graph, which constitutes the basis for the stochastic diffusion process. In the stochastic diffusion algorithm, the Kullback-Leibler divergence is used to measure the distance between the distributions. uCLAVS [14] is a web service-oriented ontology framework for malware and intrusion detection. uCLAVS is based on the idea that the files analysis can improve their performance if they are moved to the network instead of running on every host. It enables each process to enter the system files, send them to the network, and then to decide whether they are executed according to the threat delivered report. Reference [15] proposes a model to reduce on-device CPU, memory, and power resources whereby mobile antivirus functionality is moved to an off-device network service employing multiple virtualized malware detection engines. TaintDroid [16] is a system-wide dynamic taint tracking and analysis system capable of simultaneously tracking multiple sources of sensitive data.

Meanwhile, programming behavior [5] is a new mechanism that has been integrated into the malware detection approaches [1, 9]. However, it has been widely used in business process [17], cache optimization [18], and operating systems [19]. However, the current research programs have serious common drawbacks: (1) most of the verification

operations as well as the databases are performed locally, which may cause significant security issues if the databases are hacked, and (2) the approaches of local browsers lack modularity, which will cause excessive and inefficient workloads for programmers.

This paper introduces a SOA concept for malware detection mechanisms, in order to construct a distributed malware detection framework with behavior analysis model. SOA is widely applied in software services, web services, operating systems, and so on. Various SOA frameworks have been proposed in many fields, such as chip design [20], mobile computing system [21], classroom scheduling [22], enterprise architecture [23], Internet browser [24], and electronic productions [25]. The advantages of SOA are to integrate various services and provide unified interfaces within different solutions.

In order to learn from the SOA concepts, we have also summarized the cutting-edge SOA researches. Alam et al. [17] present a behavioral attestation method for business processes. Zhang et al. [26] provide a presentation for proactively recommending services in a workflow composition process, based on service usage history. Haki and Forte [23] demonstrate that using the SOA concept into the enterprise architecture (EA) framework makes the best of the synergy existing between these two approaches. Zhou et al. [6] explore service composition and service dependency and propose an extended dependency-aware SOA model. A loosely coupled service-oriented implementation is presented in [27]. The architecture takes advantage of Octave models in creating and using prediction models. In this framework, every method is applied as an Octave script in a plug-in fashion. Achbany et al. [28] present an allocation method of services to tasks, but the algorithm is not applied in realistic systems. In conclusion, since SOA has the ability for software across organizations and network boundaries to collaborate efficiently, it has been widely employed in aspects of research areas to facilitate researchers.

Although there is a lot of research works related to SOA and malware detection for mobile devices platform, respectively, there is only a few studies on integrating SOA and malware detection in order to construct a service-oriented abnormal behavior analysis framework.

To utilize SOA architecture's benefits, this paper presents a services-oriented malware detection and authentication mechanisms at the server side. All clients send requests to the web servers at run time in order to obtain a list of malware behaviors. This paper is extended from previous work at [29]. The paper proposes a distributed malware detection framework with the following features:

- (1) a light-weight profiling and information collection application on mobile devices to record all the normal and irregular behaviors;
- (2) a malware detection and abnormal behavior mechanism as separate modules;
- (3) a set of system behavior analysis schemes which integrate CPU/memory utilization, battery endurance, and network traffic flow.

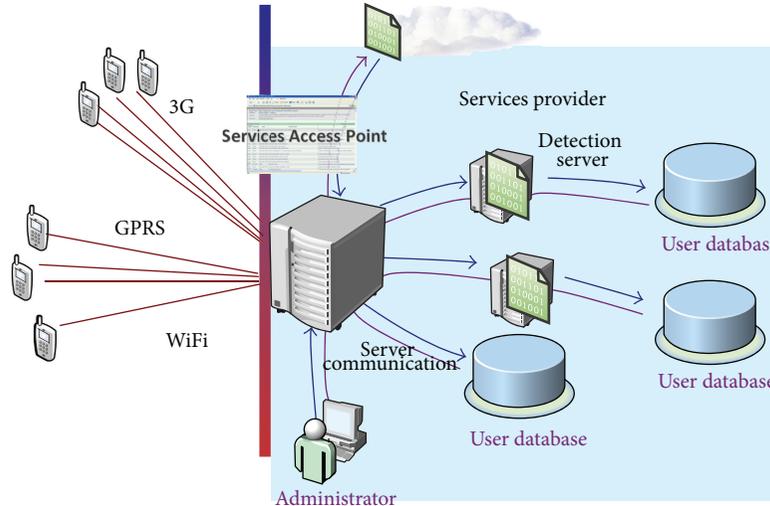


FIGURE 2: Architecture framework for SmartMal.

3. SOA Architecture Model

The concept of this paper is to apply SOA concepts into malware detection framework design. By integrating remote irregular and behavior analysis, the aim is to design an integrated client-server application with abnormal behavior maintenance. The architecture framework for SmartMal is illustrated in Figure 2. The system runs at client-server mode, of which the applications running on each client mobile device are in charge of keeping the record of the smartphones and collecting abnormal information. The selected abnormal information which is represented as vectors and extensible markup language (XML) messages are sent to the remote servers via general packet radio service (GPRS), 3G, or WiFi networks. Also the servers are distributed in one communication server and multiple malware detection servers. The communication server is responsible for exchanging messages with client users. The communication data are mainly through web services, in which data are packaged in certain data formats such as XML and JSON. After abnormal information is received, the communication server should forward the message to a specific server, in according to the client ID and system load balancing status. The detection algorithm running on the distributed servers will identify and return the results to major servers. The information will be stored in database and alerted to corresponding terminal devices when an attack or abnormal message occurs.

The SmartMal architecture provides a set of administrative control web pages. Once the data is updated, the new information will be pushed into communication servers and user clients simultaneously.

To manage the massive mobile data, the architecture maintains a *global status stable* that records the current server traffic amount data. Each server has a unique *tag*, while the major server has the smallest *tag*. The major server is elected from all the candidate servers, under the election algorithm illustrated in Algorithm 1. Denoted S indicates

```

INPUT: Server Set  $S$  and  $tag_j, j \in [1, n]$ .
OUTPUT: Major Server ID

(1) for each server  $tag_j$  in  $S$ 
(2)   send request with  $tag_s$  to all other servers
(3)   receive requests from all other servers
(4)   if  $tag_s < \text{Min}\{tag_j, j \in [1, n], j \neq i\}$  then
(5)     set server  $tag_s$  as the main server
(6)   else then
(7)     set server  $\text{Min}\{tag_j, j \in [1, n], j \neq i\}$  as main server
(8)   end if
(9) end for
    
```

ALGORITHM 1: Algorithm to elect the major server.

all the candidate server set, and the size of the set is n , represented as $tag_j, j \in [1, n]$.

3.1. *Client Architecture for Mobile Devices.* The Client's main function is to extract abnormal features as follows.

(1) *Feature Extractor.* This is the main module of the client. All the features are extracted through APIs provided by the Android application framework or information read from the Linux kernel. The collected features are clustered into three primary levels: Linux Kernel level (e.g., CPU, RAM, etc.), application level (e.g., messages, calling, etc.), and user behavior level. The user behavior level includes significant features that can reflect the user behavior, such as the screen on/off and the key pressed frequency. The feature extracting frequency is controlled by a timer whose value can be changed by user with the default value of 30 seconds. A total of 29 features are collected during every extracting, and the vector data structure is used to store features. As the data size of each transmission is very small (less than 200 bytes), compression mechanisms may not be able to achieve efficient performance.

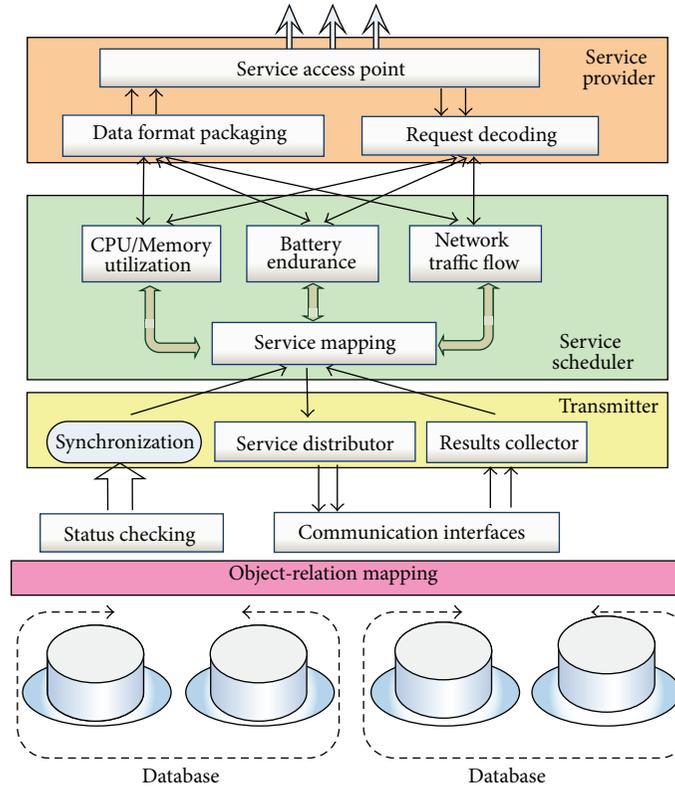


FIGURE 3: Hierarchical layer of servers.

(2) *Communication Module*. This module sends feature vectors to remote servers and receives anomaly alerts from the servers if the features are detected as anomaly. If the client is connected to the server for the first time, the communication module will request registration with the unique international mobile equipment identity (IMEI) of the smartphones.

(3) *Graphical User Interface (GUI)*. This module provides the users with a mean to configure client parameters, such as the value of extractor timer and server IP address.

3.2. *Anomaly Detection Server*. The anomaly detection server’s major task is to classify the feature vectors as normal or abnormal. The components include the following.

(1) *Database*. MySQL is used to store massive feature vectors with classValue (normal or anomaly). Database interfaces are provided for various operations. In the database, a total_table relation includes all vector information, and each detector corresponds to a detector_table, while the primary key is extract_time and phone_tag. All newly received feature vectors were stored in total_table. For each feature vector, it was assigned to the corresponding detection server according to the phone_tag and processing history. If the client was newly registered, the vector was assigned to the lowest load server.

(2) *Detecting Module*. This is the major module of the detection server, and complex detecting algorithms are

implemented here. It consists of several detectors with a detector manager. Each detector is corresponding to a classification algorithm which distinguishes between normal and abnormal feature vectors, such as J48 Detector implemented with the C4.5 decision-tree algorithm. When new feature vectors come, each detector fetches the set of vectors from detector_table, builds the classifier (if it does not exist), and classifies the feature vectors. Then, the detector manager gives out the final results by integrating all detectors’ results and stores the results into total_table and detector_table. The detector manager can also configure the parameters of detectors.

(3) *Communication Module*. This module communicates with the client and deals with various requests and messages. The module passes received feature vectors to detecting module, and if the vector is detected as anomaly the module will send an anomaly alert to the clients.

(4) *Client Manager*. When a new client requests for registration, this module will register the client with the IMEI.

(5) *GUI*. The server’s GUI configures database and visualizes current detectors and connected clients.

3.3. *Service-Oriented Hierarchical Model*. Figure 3 presents the hierarchical model of distributed servers, which consist of three layers: services, service scheduler, and transmitter. The functionalities of each layer are introduced as follows.

First of all, services provider provides service access points (SAPs) to clients. Each SAP is in charge of one specific kind of service. All the SAPs are provided with a data format packaging mechanism. When a request arrives, the SAP first decodes the target request and then identifies which service is requested. Then, the specified request will be sent to service scheduler.

Second, services scheduler is in charge of service scheduling and mapping. Each Internet request may include several service requests. Therefore, if more than one servant is available, then each service request must be mapped and scheduled to a certain servant according to the system's load balancing status.

Finally, transmitter dispatches the subtasks to different servants for execution. After the task is completed, the results are collected by transmitter.

With respect to the period-like features of 3G/GPRS/WiFi client modules, SmartMal server provides three services for demonstration: CPU/memory utilization, battery endurance, and network traffic flow. From the exploration of the state-of-the-art studies, it is quite common that the malware applications will either drag down the network flow performance, resulting in the congestion, or illegally waste the CPU/memory utilization, or the energy.

The high level services are mapped to different servants. In order to provide a feasible system for services, at least one servant for each service is integrated into the system. Each service request is transmitted to a specific servant. All the servants are managed for efficiency use. The data transmission between servant and service layer is through communication interfaces and status checking interface.

Status checking interface is responsible for providing synchronization information of diverse servants for services mapping and scheduling, such as load balancing and services bottleneck exploration.

The physical layer consists of database and object-relation (O-R) mapping mechanisms. Generally, all the analyzed irregular behaviors are stored in databases, which may be located at distributed areas. Dealing with the current relation based database models, O-R mapping methods are widely employed for object-oriented abstractions, such as Hibernate and Toplink. Benefiting from these approaches, the high-level objects are mapped to relational databases. We hereby utilize TopLink for demonstration to map the database tuples to the standard C++ classes.

3.4. Remarkd Features. It is quite true that spelling out the requirements for a software system under development is not an easy task, and translating captured requirements into correct operational software can be even harder [5]. Many technologies (languages, modeling tools, and programming paradigms) and methodologies (agile, test-driven, and model-driven) are designed, among other things, to help address these challenges. One widely accepted practice is to formalize requirements by behavioral programming skills in the form of use cases and scenarios.

However, in realizing abnormal detection based analysis method, not all the collected behaviors are reflecting

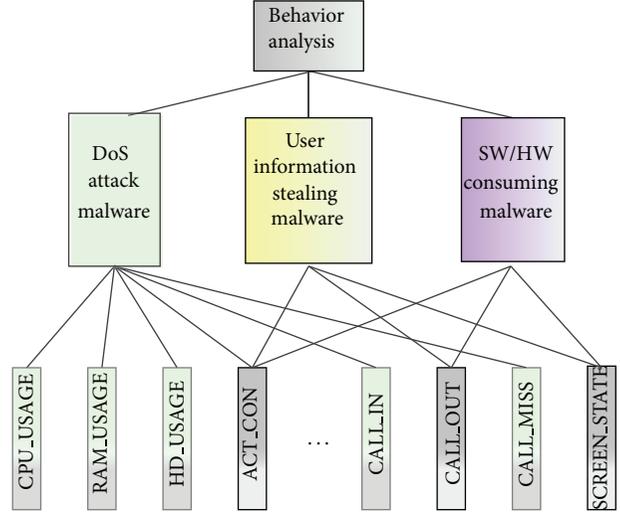


FIGURE 4: Hierarchical layers for the remarked features.

abnormal message. Therefore, it is challenging to choose the required information among the extracted set of phases or behaviors. It has been proved that the detection efficiency can be significantly improved by the refinements of degrading the dimensions and eliminating the superfluous information [11].

In this paper, we model the weight for each feature as a synthesized combination of subjective and objective weight to identify the behavior and characteristics. The weight of the synthesized weight is represented as follows:

$$w_i = uw_{si} + (1 - u)w_{oi}. \quad (1)$$

For each feature i , w_i represents the synthesized weight, w_{si} denotes the subjective weight, w_{oi} refers to objective weight, and u indicates the proportion of subjective weight. The major contribution of this method is to introduce the subjective weight that can leverage the default analyzed results obtained from the behaviors. We use an analytic hierarchy process (AHP) algorithm to construct a three-layer model to divide the complex strategic decision problem into different subjects aiming at multiple targets. For each subject, a fuzzy quantitative approach is employed to calculate the weight for each feature and then merge it hierarchically.

Figure 4 illustrates the hierarchical model for the remarked features that is composed of three layers. The top layer denotes the final target of behavioral analysis to identify the optimal features. In the middle layer, three classifications are listed according to different abnormal behaviors such as DoS attack malware, user information stealing malware, and irregular software/hardware resource consuming malware. Finally, all the behaviors on operating system application interfaces are reflecting the three abnormal behaviors in bottom level.

In the hierarchical analysis method, the relative weight a_{ij} represents the correlation between the i th element and the j th element. Assume that there are $n \times m$ elements in total, and then $A = (a_{ij})_{n \times n}$ is denoted as the correlation matrix. For the elements in the matrix, we have $a_{ji} = 1/a_{ij}$, $a_{ij} = a_{ij} \cdot a_{ij}$ and

TABLE 1: Configurations in the correlation matrix.

Value	Representation
$a_{ij} = 1$	i th element and j th element have the same effect
$a_{ij} = 3$	i th element is a little more important than j th element
$a_{ij} = 5$	i th element is important than j th element
$a_{ij} = 7$	i th element is much more important than j th element
$a_{ij} = 9$	i th element is extremely more important than j th element
$a_{ij} = 2n$	Superior of i th than j th element between $2n - 1$ and $2n + 1$

$a_{ii} = 1$. The values and representation of different parameters are described in Table 1.

Then, we normalize the matrix A to matrix Q :

$$Q = (q_{ij})_{n \times n} \quad (2)$$

$$q_{ij} = \frac{a_{ij}}{\sum_{k=0}^n a_{kj}}$$

Add the elements in matrix Q by rows to get the vector α :

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)^T \quad \text{in which } \alpha_i = \sum_{j=1}^n q_{ij}. \quad (3)$$

The vector α is normalized to the weight vector W :

$$W = (w_1, w_1, \dots, w_n)^T \quad \text{in which } w_i = \frac{\alpha_i}{\sum_{j=1}^n \alpha_j}. \quad (4)$$

After W has been calculated, we need to maintain the consistency with respect to the subjective perceptive. Consistency Index (CI) is utilized as the evaluation metrics:

$$CI = \frac{\lambda_{\max} - n}{n - 1}. \quad (5)$$

In (5), λ_{\max} refers to the peak value of the feature, which is derived in the following equation:

$$\lambda_{\max} = \frac{1}{n} \sum_{i=1}^n \frac{(AW)_i}{w_i}. \quad (6)$$

Moreover, we also utilize consistency rate (CR) as to characterize and to model the proportion of the consistency $CR = CI/RI$. RI refers to random index that is the maximum value that a_{ij} is selected completely at random. Obviously, the value of CR depends on the order of the matrix n . The consistency is accepted only if $CR < 0.1$, and otherwise the correlation matrix should be leveraged until the condition is met.

After the weight for each target has been calculated, they could be moved forward to the next step, where different weight vectors are systematized as a combination. Specially, the steps to combine all the vectors are as follows.

- (1) Calculate the importance for every level to the top level. This calculation process is carried out from top to bottom.

TABLE 2: Applications for profiling.

Game applications		
Fruit Ninja	Angry Birds	Can Knockdown
X Construction	Cut String	Gold Miner
Bubble Ball	Shift	Flight Chess
Sudoku	Talking Tom	
Software tool applications		
Office Suite	360 Guard	Root Explorer
King Soft	iReader	PowerAMP
Mobo Player	UCWeb	Fetion
Task Manager	MSN	Google Map
Google Music	King Reader	Mobile TV
Storm Player	Tencent QQ	Shang Mail
Sina Weibo	RenRen	Adobe Reader

- (2) Assume that there are n_{k-1} elements resided in the $(k-1)$ th level and the weight vector is calculated as

$$W^{(k-1)} = (w_1^{(k-1)}, w_2^{(k-1)}, \dots, w_n^{(k-1)})^T. \quad (7)$$

- (3) Assume that there are n_k elements resided in the k th level and the weight vector of the impact on $(k-1)$ th level refers to

$$P^{(k)} = (w_{1j}^{(k)}, w_{2j}^{(k)}, \dots, w_{nj}^{(k)})^T. \quad (8)$$

If the i th element is independent with the j th element, then $w_{ij} = 0$.

- (4) From (2) and (3), we can get that the weight vector of the impact on k th level is

$$W^{(k)} = (P_1^{(k)}, P_2^{(k)}, \dots, P_n^{(k)}) W^{(k-1)}. \quad (9)$$

After both the subjective and objective weights are evaluated, the proportion coefficient u can be calculated in step 1.

4. Behavior Analysis

To demonstrate the effectiveness of the SmartMal architecture, we have implemented a prototype application for both behavioral analysis and abnormal malware attack detection. Due to the abnormal malware detection for mobile devices, there is no acknowledged data set. For behavioral analysis, it is extremely important to select a fair and reasonable benchmark set for behavior analysis. We have selected 32 most highly ranked applications in the Android market, including 11 game applications and 21 software tools, presented in Table 2. All the software programs are installed in three mobile devices for malware detection (1 Moto Me722 handset and 2 Samsung S5830 handsets).

All the 32 applications are installed and run on the smartphones for at least one hour, during which the malware detection engines are running at back stage. The back stage engine is configured to sample the mobile device application

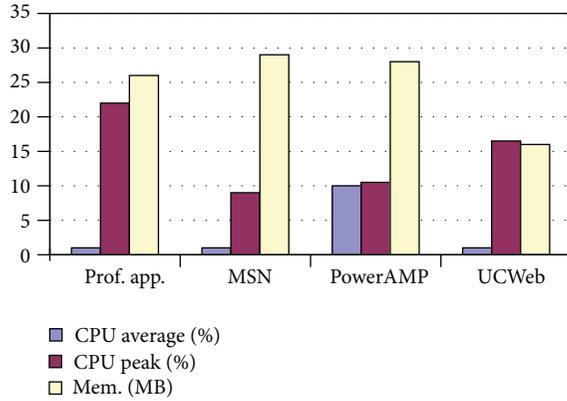


FIGURE 5: CPU/memory utilization.

running status every 30 seconds. After the execution is finished, there should be up to 120 items of the characterized features. For some special purposed behavior, such as battery consumption, text messages, and incoming/outgoing phone calls, it is not fair to gather the fingerprints every 30 seconds. In these application-specific scenarios, we use an accumulative value for the recent 9 intervals, each with 30 seconds. Finally, all these items are marked with normal behaviors for further comparison and detection schemes.

Alternatively, these three Android smartphones are delivered to three persons for regular daily use. Meanwhile, the back stage engine keeps on tracking the featured behavior. This period has lasted for more than 90 days.

In this paper, we focus on the CPU/memory utilization rate, battery consumption, and network traffic flow to analyze the workload behavior. The experimental results are illustrated in Figure 5.

4.1. CPU/Memory Utilization. To evaluate the CPU and memory utilization, we use *top* command in Linux operating system to observe the CPU and memory utilization of the client. Meanwhile, we have also compared the statistics for typical software applications including IM software MSN, media player PowerAMP, and UCWeb, which is illustrated in Table 3. Taking CPU average utilization rate into consideration, the profiling account only occupies less than 1% of CPU, which is ignorable. Alternatively, as our profiling application explores the feature every 30 seconds, thereby the peak utilization achieves 20%–24%, but the duration of the sample is too short to be noticed. Therefore, it does not cause any influence for user experiences. Finally, for the sake of the memory utilization, our profiling application takes up to 26 MB, which is smaller than MSN and PowerAMP applications, and, consequently, the overhead is affordable for less than 5%. Note that a general smartphone can integrate more than 512 MB internal memory.

4.2. Battery Endurance. One of the most serious challenges for smartphones is the power consumption and battery endurance. To analyze the behavior of how our profiling application has an impact on the battery endurance, we have

TABLE 3: Battery endurance statistics.

	100% to 15%	100% to 5%
Profiling offline	525 minutes	590 minutes
Profiling online	480 minutes	540 minutes

evaluated the endurance trial on both scenarios, profiling online and offline, described in Table 3. On one hand, when the client application is offline, it takes approximately 525 minutes from the full battery charge 100% to 15%, while it takes 590 minutes when only 5% battery is left. On the other hand, with respect to the online profiling application, the duration only lasts for 480 minutes from 100% to 15%, while it lasts for 540 minutes from 100% to 5%.

4.3. Network Traffic Flow. Due to the fact that our profiling application requires Internet to transfer extracted behaviors to the major server, consequently, we also need to analyze the behavior for the network traffic flow to evaluate how our approach has an impact on the Internet utilization. We employ TrafficStats toolset provided in Android operating system to monitor the traffic flow for both single upload transaction and batch uploading process. For the sake of single uploading procedure, the packaged message should be delivered once the abnormal behavior is detected, and the uploading flow speed is 9 Kb/10 min, while the downloading speed is 4 Kb/10 min. With respect to the batch procedure, up to 10 featured messages will be batched together into a package for uploading operation. However, from the experiment, we can get a result which is exactly the same as the single uploading transaction; that is, the uploading flow speed is 9 Kb/10 min, while the downloading speed is 4 Kb/10 min. Due to the small volume for each packet, multiple messages can reside in the same package to reach a minimum size for one pack. Considering the 3G and WiFi network bandwidth, the 13 Kb/10 min network traffic flow is acceptable.

5. Malware Detection: A DoS Attack Case

In this section, we introduce abnormal malware detection in real DoS attack case. It is quite common that the network flow of the mobile devices statistics could be periodical. For example, the TCP SYN packages and RAB establishing/release procedures can be repeated when in a period-like manner. In particular, the mathematic idioms used in malware detection are denoted in Table 4.

5.1. Detection Algorithm. Generally speaking, τ decides the aggregation degree for the accumulated data, while distinct DoS attack behaviors on mobile networks perform different aggressive abnormal data degree. We set the minimum interval $\tau_0 = 30$ seconds, which is used to accumulate more coarse grained information, such as 1 minute, 5 minutes, 10 minutes, and 30 minutes.

Given two probability distribution for distinct time slices with same feature and intervals, the formal probability distribution can be normalized to $X^\tau(k_1)$ and $X^\tau(k_2)$.

TABLE 4: Idioms and terms defined in DoS attack cases.

τ	Statistics intervals
$v_i^\tau(k)$	i th devices behavior in the interval k
$N^\tau(k)$	Total amount of devices in the interval k
$v^\tau(k)$	$\{v_i^\tau(k), i = 1, 2, \dots, N^\tau(k)\}$ the vector in interval k
$X^\tau(k)$	Probability distribution for normalized $v^\tau(k)$
$L(k_1, k_2)$	Similarity for $X^\tau(k_1)$ and $X^\tau(k_2)$
$W_0(k)$	The observation window for $X(k)$
$W_1(k)$	The sampling observation window for $X(k)$
$D_I(k)$	Internal distance for probability correlation $X(k)$
$D_E(k)$	External distance for probability correlation $X(k)$
M	Abnormal probability distribution set

INITIATION: Set values of t, N, M
INPUT: $V(k), t, N, M$
OUTPUT: $X(k)$

Begin

- (1) Obtain $X(k)$ and $N(k)$ from $V(k)$
- (2) Define the observation window $W_0(k)$
- (3) Select $W_1(k)$ from $W_0(k)/M$ by sampling
- (4) Calculate the distance $D_I(k)$ and $D_E(k)$
- (5) **If** $D_E(k) > D_I(k)$
- (6) $X(k)$ is detected as anomaly
- (7) Set $M = M \cup \{k\}$
- (8) **else** $X(k)$ is normal
- (9) Increase k by one and go-back to (1)

End

ALGORITHM 2: Pseudocode of abnormal behavior analysis for mobile networks.

First, we define $L(k_1, k_2)$ as the distance for the two probability distributions k_1 and k_2 . Then, we need to sample the observation windows in the following two phases.

First, for the sake of a given interval k and observation window $W_0(k)$, we need to select the time slice which is most relevant to the amount of total registered mobile devices in (10) with an experimental value from 5% to 15%. Consider

$$W'_0(k) = k_i \in W_0(k), \quad \frac{|N(k) - N(k_i)|}{N(k)} \leq S. \quad (10)$$

Second, choose probability distribution $X^\tau(k)$ that is most relevant to $X(k)$ from the observation window $W_0(k)$ to organize a sampling observation window $W_1(k)$.

After $W_1(k)$ has been identified, both internal distance $D_I(k)$ between $X(k)$ and $W_1(k)$ and external distance $D_E(k)$ is calculated, respectively, and presented in (11) and (12). Then, the processing flow is described in Algorithm 2. Consider

$$D_I(k) = \{L(k_i, k_j), k_i, k_j \in W_1(k), k_i \neq k_j\}, \quad (11)$$

$$D_E(k) = \{L(k, k_i), k_i \in W_1(k)\}. \quad (12)$$

5.2. *Similarity Evaluation.* To analyze whether the behavior is a normal operation or the malware attack operation,

we use a Kullback-Leibler (KL) divergence based approach. Assuming that parameters p and q represent the probability distribution of two data sets, therefore, the KL divergence can be used to measure the relative entropy between the two probability distributions described in the following equation:

$$D(p \parallel q) = E \left[\log \left(\frac{p(\omega)}{q(\omega)} \right) \right] = \sum_{\omega \in \Omega} p(\omega) \log \left(\frac{p(\omega)}{q(\omega)} \right), \quad (13)$$

in which $0 \log(0/q) = 0$ and $p \log(p/0) = \infty$. Moreover, the KL divergence is 0 only when p is equal to q . Since KL divergence is not a metric, we propose a revised metric to measure the distance. Consider

$$L(p, q) = \frac{1}{2} \left(\frac{D(p \parallel q)}{H_p} + \frac{D(q \parallel p)}{H_q} \right), \quad (14)$$

where $D(p \parallel q)$ and $D(q \parallel p)$ represent the KL divergence, while H_p and H_q are the entropy for p and q , respectively. In particular, the calculation of the entropy is introduced in the following equation:

$$H(X) = (P_1, P_2, \dots, P_n) = P(x_i) \log P(x_i). \quad (15)$$

Referring to (15), X represents the probability distribution. $P(x_i)$ indicates the probability that the source fetches i th signal, and we have $\sum_i P(x_i) = 1$. Due to the calculation of $D(p \parallel q)$ which requires the additional information, $D(q \parallel p)/H_q$ denotes the extra workload for the calculation. To maintain the accuracy, the final distance L is set to the average distance of $L(p, q)$ and $L(q, p)$.

Referring to a probability distribution, the dimension for the two distributions can be different and an occasional case of $p \log(p/0) = \infty$ may happen. In order to avoid this situation, we can choose the maximum mobile device volume as the uniform-dimensional degree, while for the insufficient distribution, we can use a signal ϵ representing 0; therefore, the situation of $p \log(p/0) = \infty$ can be avoided. In this paper, we set $\epsilon = 10^{-10}$.

5.3. *Experimental Results and Analysis.* We setup a simulation platform to verify the DoS behavior detection for periodic probability distribution. The NET_SEND behavior is implemented as the TCP SYN simulation, while the back stage servers will send an abnormal malware messages demonstrating the SYN flooding attack. In order to simulate a relatively large scale experimental platform, we combine the message from 3 smartphones every 5 minutes into a 10-length chain with a 30-dimensional NET_SEND vector; then, it is normalized into the probability distribution for TCP SYN behaviors.

We run the applications on smartphones for two months continually. As one probability distribution vector includes the information collected every 5 minutes, we totally get $2 \times 30 \times 24 \times 12 = 15480$ normalized vectors.

Figure 6 illustrates the detection accuracy for DoS attacking malwares. The X-axis represents the amount of attaching smartphones, while the Y-axis is the detected accuracy rate

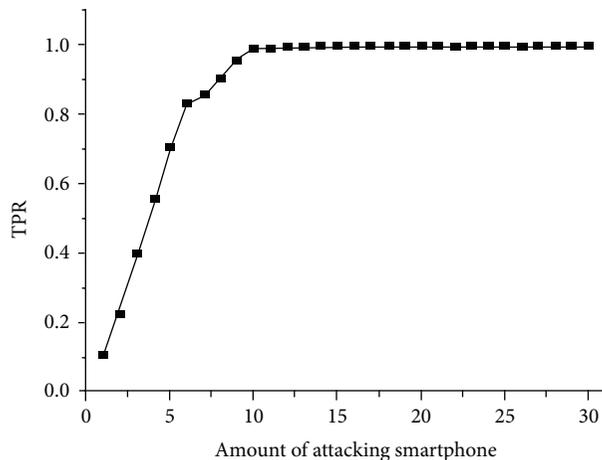


FIGURE 6: The detection accuracy for malware applications under different amounts of attacks.

for malware attacks, denoted as true positive rate (TPR). It can be easily derived that the TPR is increased with the amount of attacking smartphones. When there is only one device, the TPR is only 10%, which means more than 90% attacks failure. When the device amount is increased to 5, the TPR is also grown up to 50%. When there are more than 10 devices in total, the TPR is stable as high as 99.1%. Our approach can obtain both highly efficient and accurate results to detect all the abnormal behaviors and malware attacks.

6. Conclusions

This paper proposed a service-oriented malware detection framework “SmartMal,” which is the first work to combine SOA concepts with state-of-the-art behavior-based malware detection methodologies. By applying SOA into the framework, irregular behaviors can be processed in parallel servers instead of operating locally. Utilizing the distributed operation of irregular behavior analysis, SmartMal can largely reduce clients’ computational complexity with great flexibility and modularity.

Moreover, as a test case, we have proposed a randomization method to defend against signaling DoS attacks on 30 cellular networks from the system-design perspective. By setting the parameter that is crucial for attacking efficiency as random distribution, the parameter is more difficult to be measured and the measured value is the maximum of the random value. The cost of launching an attack is increased enormously. Our simulation of signaling attack via RAB establishment release shows that our randomization method can achieve as high as 99.1% of the malware and irregular behaviors. The randomization method is easy and effective towards this kind of signaling attacks including paging attacks.

The initial results are promising, but there is a lot of work worth pursuing. Future works include extending the server to distributed cloud systems to achieve high throughput and integration of services. Meanwhile, we also plan to integrate the design-space exploration into the malware detection

methods and behavior analysis to improve the accuracy and flexibility.

Conflict of Interests

The authors of the paper do not have a direct financial relation that might lead to a conflict of interests.

Acknowledgments

This work was supported by the National Science Foundation of China under Grants no. 61272131, no. 61379040, and no. 61202053, Fundamental Research Funds for the Central Universities no. WK0110000034, and Jiangsu Provincial Natural Science Foundation Grant no. SBK201240198. The authors deeply appreciate many reviewers for their insightful comments and suggestions.

References

- [1] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, ““Andromaly”: a behavioral malware detection framework for android devices,” *Journal of Intelligent Information Systems*, vol. 38, no. 1, pp. 161–190, 2012.
- [2] Internet-Filter-Review, *Top Ten Reviews*, 2010, <http://internet-filter-review.toptenreviews.com/>.
- [3] L. Davi, A. Dmitrienko, A.-R. Sadeghi, and M. Winandy, “Privilege escalation attacks on android,” in *Proceedings of the 13th International Conference on Information Security*, Springer, Boca Raton, Fla, USA, 2010.
- [4] A. D. Schmidt, H. G. Schmidt, L. Batyuk et al., “Smartphone malware evolution revisited: android next target?” in *Proceeding of the 4th International Conference on Malicious and Unwanted Software (MALWARE '09)*, pp. 1–7, canMontreal, Canada, October 2009.
- [5] D. Harel, A. Marron, and G. Weiss, “Behavioral programming,” *Communications of the ACM*, vol. 55, no. 7, pp. 90–100, 2012.
- [6] J. Zhou, D. Pakkala, J. Perälä, E. Niemelä, J. Rieki, and M. Ylianttila, “Dependency-aware service oriented architecture and service composition,” in *Proceedings of the IEEE International Conference on Web Services (ICWS '07)*, pp. 1146–1149, July 2007.
- [7] J. Cheng, S. H. Y. Wong, H. Yang, and S. Lu, “SmartSiren: virus detection and alert for smartphones,” in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys '07)*, pp. 258–271, ACM, San Juan, Puerto Rico, June 2007.
- [8] T. K. Buennemeyer, T. M. Nelson, L. M. Claggett, J. P. Dunning, R. C. Marchany, and J. G. Tront, “Mobile device profiling and intrusion detection using smart batteries,” in *proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS '08)*, p. 296, Waikoloa, Hawaii, USA, January 2008.
- [9] A. Bose, X. Hu, K. G. Shin, and T. Park, “Behavioral detection of malware on mobile handsets,” in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*, pp. 225–238, Breckenridge, Colo, USA, June 2008.
- [10] A. S. Shamili, C. Bauchhage, and T. Alpcan, “Malware detection on mobile devices using distributed machine learning,” in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 4348–4351, August 2010.

- [11] A. D. Schmidt, R. Bye, H. G. Schmidt et al., "Static analysis of executables for collaborative malware detection on android," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, 2009.
- [12] A. Schmidt, J. H. Clausen, A. Camtepe, and S. Albayrak, "Detecting symbian OS malware through static function call analysis," in *Proceedings of the 4th International Conference on Malicious and Unwanted Software (MALWARE '09)*, pp. 15–22, Montreal, Canada, October 2009.
- [13] T. Alpcan, C. Bauckhage, and A.-D. Schmidt, "A probabilistic diffusion scheme for anomaly detection on smartphones," in *Proceedings of the 4th IFIP WG 11.2 International Conference on Information Security Theory and Practices: Security and Privacy of Pervasive Systems and Smart Devices (WISTP '10)*, pp. 31–46, Springer, Passau, Germany, April 2010.
- [14] C. A. Martínez, G. I. Echeverri, and A. G. Castillo Sanz, "Malware detection based on cloud computing integrating intrusion ontology representation," in *IEEE Latin-American Conference on Communications (LATINCOM '10)*, September 2010.
- [15] J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian, "Virtualized in-cloud security services for mobile devices," in *Proceedings of the 1st Workshop on Virtualization in Mobile Computing*, ACM, Breckenridge, Colo, USA, 2008.
- [16] W. Enck, P. Gilbert, B.-G. Chun et al., "TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones," in *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, USENIX Association, Vancouver, Canada, 2010.
- [17] M. Alam, M. Nauman, X. Zhang, T. Ali, and P. C. K. Hung, "Behavioral attestation for business processes," in *Proceedings of the IEEE International Conference on Web Services (ICWS '09)*, pp. 343–350, July 2009.
- [18] N. Fabian, B. Rainer, and K. Wolfgang, "Adaptive cache infrastructure: supporting dynamic program changes following dynamic program behavior," in *Proceedings of 9th Workshop on Parallel Systems and Algorithms (PASA '08)*, Dresden, Germany, 2008.
- [19] C. Krintz and R. Wolski, "Using phase behavior in scientific application to guide linux operating system customization," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, IEEE Computer Society, April 2005.
- [20] C. Wang, X. Li, P. Chen, J. Zhang, X. Feng, and X. Zhou, "Regarding processors and reconfigurable IP cores as services," in *Proceedings of the IEEE 9th International Conference on Services Computing (SCC '12)*, pp. 668–669, Honolulu, Hawaii, USA, June 2012.
- [21] D. Van Thanh and I. Jørstad, "A service-oriented architecture framework for mobile services," in *Proceedings of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop*, pp. 65–71, IEEE, July 2005.
- [22] A. Wang, C. Wang, X. Li, and X. Zhou, "SmartClass: a services-oriented approach for university resource scheduling," in *Proceedings of the 10th International Conference on Services Computing*, 2013.
- [23] M. K. Haki and M. W. Forte, "Service oriented enterprise architecture framework," in *Proceedings of the 6th World Congress on Services (SERVICES '10)*, pp. 391–398, Miami, Fla, USA, July 2010.
- [24] A. Wang, C. Wang, X. Li, and Z. Xuehai, "SOBA: a services oriented browser architecture with distributed URL-filtering mechanisms for teenagers," in *Proceedings of the IEEE World Congress on Services*, 2013.
- [25] I. M. Delamer and J. L. M. Lastra, "Service-oriented architecture for distributed publish/subscribe middleware in electronics production," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 4, pp. 281–294, 2006.
- [26] J. Zhang, W. Tan, A. John, I. Foster, and R. Madduri, "Recommend-as-you-go: a novel approach supporting services-oriented scientific workflow reuse," in *Proceedings of the IEEE International Conference on Services Computing (SCC '11)*, pp. 48–55, IEEE Computer Society, July 2011.
- [27] G. Kousiouris, D. Kyriazis, K. Konstanteli, S. Gogouvitis, G. Katsaros, and T. Varvarigou, "A service-oriented framework for GNU Octave-based performance prediction," in *Proceedings of the IEEE International Conference on Services Computing (SCC '10)*, pp. 114–121, Miami, Fla, USA, July 2010.
- [28] Y. Achbany, I. J. Jureta, S. Faulkner, and F. Fouss, "Continually learning optimal allocations of services to tasks," *IEEE Transactions on Services Computing*, vol. 1, no. 3, pp. 141–154, 2008.
- [29] C. Wang, Z. Wu, A. Wang, X. Li, F. Yang, and X. Zhou, "SmartMal: a service-oriented behavioral malware detection framework for smartphones," in *Proceedings of the IEEE 10th International Conference on High Performance Computing and Communications & IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC '13)*, pp. 329–336, Zhangjiajie, China, November 2013.

Research Article

An Action-Based Fine-Grained Access Control Mechanism for Structured Documents and Its Application

Mang Su,¹ Fenghua Li,² Zhi Tang,³ Yinyan Yu,³ and Bo Zhou⁴

¹ State Key Laboratory of Integrated Services Network, Xidian University, Xi'an 710071, China

² State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

³ Institute of Computer Science & Technology, Peking University, Beijing 100080, China

⁴ Liverpool John Moores University, Liverpool L3 3AF, UK

Correspondence should be addressed to Fenghua Li; fhli@iie.ac.cn

Received 8 March 2014; Revised 10 June 2014; Accepted 11 June 2014; Published 17 July 2014

Academic Editor: Fei Yu

Copyright © 2014 Mang Su et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an action-based fine-grained access control mechanism for structured documents. Firstly, we define a describing model for structured documents and analyze the application scenarios. The describing model could support the permission management on chapters, pages, sections, words, and pictures of structured documents. Secondly, based on the action-based access control (ABAC) model, we propose a fine-grained control protocol for structured documents by introducing temporal state and environmental state. The protocol covering different stages from document creation, to permission specification and usage control are given by using the Z-notation. Finally, we give the implementation of our mechanism and make the comparisons between the existing methods and our mechanism. The result shows that our mechanism could provide the better solution of fine-grained access control for structured documents in complicated networks. Moreover, it is more flexible and practical.

1. Introduction

The various handheld devices, represented by tablets and smart phones, have brought a novel and collaborative way to access information. It satisfies the increasing requirement for individualized service and makes the information multidimensional in such open networked environment. As a result structured documents, which integrate both content and layout, have become a key information representation in the networked environment of cloud computing, mobile computing, and pervasive computing. While providing the original feeling and the same user experience as paper-based publications, it has speeded up the information transmission significantly. Research on structured documents contains document modeling, analysis, and recognition. At present, the readers and digital contents become more and more various. We expect that the structured document is stable in layout and flexible in content. The content of structured document not only could be texts and pictures, but also

integrate both 2D and 3D elements. Moreover, it could interact with the users as well. With the rapid increasing of data represented by structured documents, the granularity of the access control is changing from the whole document to its objects, and the requirement for interaction and permission specification has emerged. However, the current access control model and permission management are mainly realized based on operation system, database, and large-scale information system. The existing access control mechanisms are implemented based on the whole document, so they cannot be adapted to the current situation. Therefore, the description and model of the access control for structured documents have to be changed from the entire document to specific parts of it and from coarse-grained to fine-grained.

Access control is mainly used to solve the problem of data sharing and permission assignment. In the initial stages of access control, there were models such as discretionary access control (DAC) and mandatory access control (MAC). These models have mapped users and permissions directly. Later,

role based access control (RBAC) was proposed by introducing the role concept, which has assigned permission to roles instead of users. With the development of the network, the access control models based on temporal state and location have been presented for the purpose of privacy protection. The models could activate and inactivate periodic role or describe how to use the location information in permission determination [1–4]. The wide application of digital communications makes the network become an open, distributed, and complicated system, which supports mobile computing and cloud computing as well. In order to accommodate the distributed environment, UCON and RBAC for multidomain [5, 6] have been proposed. The authors in [7–9] have given the concepts of “attribute” and “action,” respectively, for the future considerations of temporal state and environmental state. At present, the latest challenges for access control are not only the different requirements of security on different devices in various environments, but also the problems of fine-grained management when the users are crossing multiple domains.

The access control for structured document, represented by XML, usually is implemented based on DAC, MAC, and RBAC, without considering temporal state and environmental state. Reference [10] has presented a fine-grained MAC model for XML documents. Its management and control are based on properties of XML. However, because it is based on MAC only, without considerations of temporal and contextual information, it cannot accommodate the demand of structured document access control at anytime and anywhere.

To solve the aforementioned problems, we have done research on structured document description and access control models. In this paper, we present an action-based fine-grained access control mechanism for structured documents by extending the model of [9]. Based on the analysis of application scenarios, we also provide the corresponding security protocols. The protocols take advantage of the current signature encryption technology to ensure the integrity of access control mechanism and provide basis for its implementation. Therefore we can skip the security proof in detail. Finally, we justify our method by comparing it with existing solutions.

The rest of the paper is structured as follows. Section 2 analyzes the common application scenarios of structured documents. The new access control mechanism of structured documents, including model, protocols, and its implementation, are presented in Section 3. The implementation of our mechanism will give in Sections 4 and 5 which compares the properties of the new mechanism with current ones. And finally we conclude the paper in Section 6.

2. Application Scenario Analysis

Users always expect to access any digital contents of structured documents anytime and anywhere. Moreover, they would like to specify the permissions according their own requirement. However, both the creators of the documents and administrators of content servers not only do hope to provide users with convenient and efficient service, but also expect to implement the fine-grained access control

of structured documents simultaneously. We analyze three common application scenarios of structured documents in the following subsections.

(1) Document preview: content providers (CP) expect to attract customers to buy the whole digital work by offering a few free contents, while customers often hope to determine the purchase intentions based on their experience of preview.

(2) Individual specification of document permissions: if user B has already purchased the digital contents needed successfully, he could expect to specify permissions and constraints associated with the usage of the contents individually. For instance, he would choose some sections to be a new section and then specify their permission for future usage. Therefore, the permission description of structured documents should be fine-grained and the permissions specification of new document should be flexible.

(3) Multielement restrictions on access to documents: there are three more aspects of the issue. Firstly, the authorization of structured documents usually has a lifetime. After the time, the documents will not be used. Secondly, the contents servers also have restrictions on the number of current online users to ensure the performance. Thirdly, the users' environmental states will play an important role in permission assignment.

3. Action-Based Fine-Grained Access Control Mechanism for Structured Documents

Based on the scenario analysis in Section 2, we propose an action-based access control mechanism in this section. In the first subsection, we present a fine-grained description for the structured documents, as our mechanism is built on top of it. In the second subsection, we introduce the action-based access control model for the structured documents based on ABAC. Finally, we describe the corresponding protocols in the last subsection.

3.1. A Fine-Grained Description. Structured documents could be described as a 5-tuple $S_{Doc} = (S_o, o_r, S_a, \partial, sattr)$.

- (i) S_o denotes all the objects sets of structured document S_{Doc} . The objects could be anything users want to describe, such as sections, chapters, pages, sentences, and even words.
- (ii) o_r denotes the root of objects, where $o_r \in S_o$, and it is defined to implement traversal and search for other objects.
- (iii) S_a denotes all the objects' attributes sets of structured document. The attributes of objects could be security level, time, location, and so on.
- (iv) $sattr$ is a binary relationship, where $sattr \subseteq S_a \times S_o$. If $a_1 \in S_a, o_1 \in S_o$, and $(a_1, o_1) \in sattr$, then a_1 is the security attribute of o_1 .
- (v) Function ∂ ($\partial : S_o \rightarrow S_o$) denotes the nesting relationship between objects. If $o_1, o_2 \in S_o$ and $\partial(o_1) = o_2$, then o_1 is nested by o_2 . Assuming object o_k is included by document d and the object nested by o_k

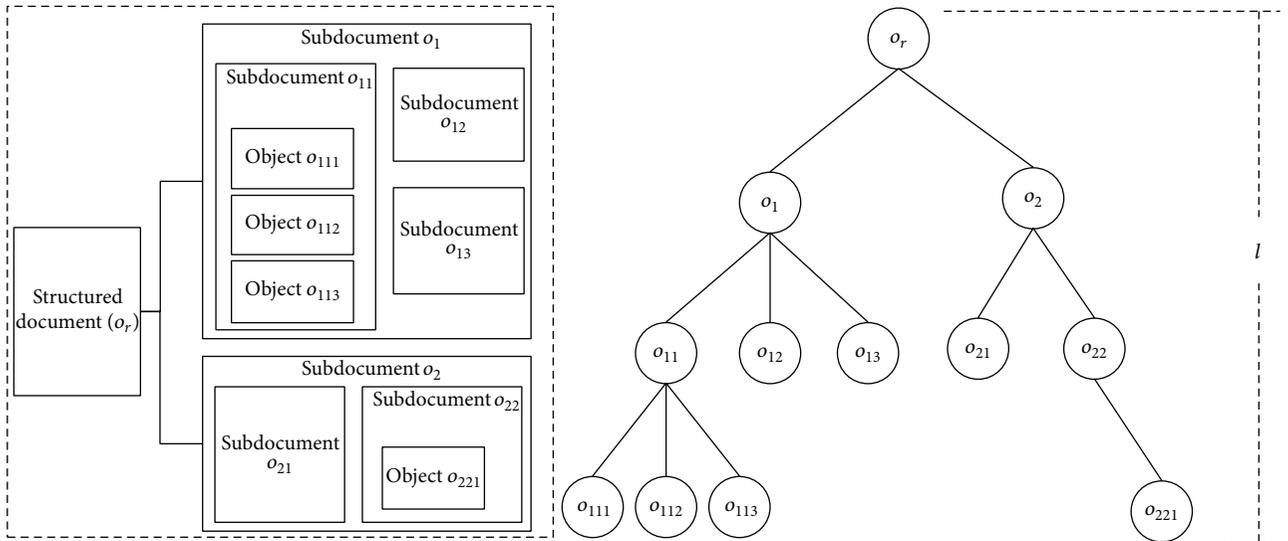


FIGURE 1: The nesting relationship between objects.

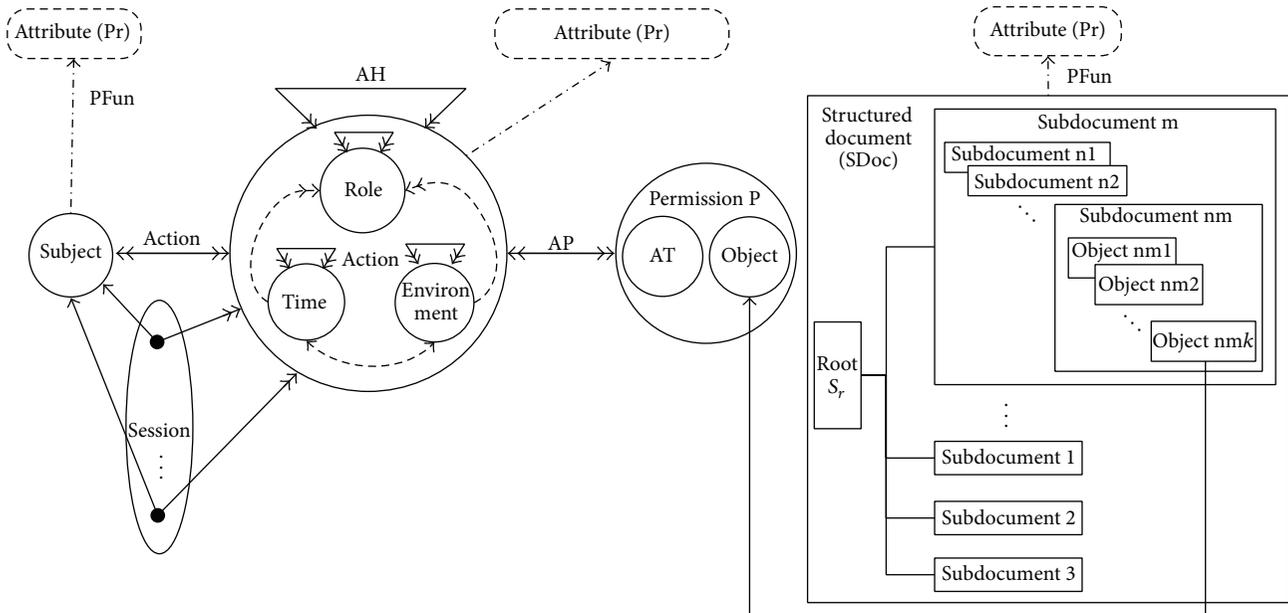


FIGURE 2: The action-based fine-grained access control model for structured document.

is represented as o_{k-l} , l is the nesting depth of o_k in d when $o_{k-l} = d$. So let $\partial_l = (\partial)^l$; then $\partial(o_k) = o_{k-l}$, $(\partial(o_k))^2 = \partial(\partial(o_k)) = o_{k-2}, \dots$, and $(\partial(o_k))^l = \partial(\dots \partial(o_k)) = o_{k-l}$. The nesting relationship of objects, subdocuments, and documents could be described by multibranches tree. The structured document is the root of the tree, and the nested objects and subdocuments are nodes or leaves. For instance, in Figure 1, $(\partial(o_{111}))^l = o_r$, $l = 3$. A sensible design of nesting depth could make the fine-grained access control more effective.

3.2. Action-Based Access Control Model. The action-based access control model for structured document is shown in

Figure 2. The action hierarchy (AH) and action-permission assignment (AP) are described in [9]. The basic concepts are defined as follows:

- (i) *Subject*: a person or a nature agent that applies to access the resource;
- (ii) *Session*: a mapping between a subject and a subset of the roles that the subject is assigned to;
- (iii) *Role*: the description of the permission set of subject who wants to access the object at special time and in special place;
- (iv) *Time*: the temporal constraint set of subject;
- (v) *Environment*: the environmental constraint set of subject. For instance, location, platform, and other

objective conditions related with access control. The system will restrict a subject's permission according to environment when it applies to access;

- (vi) *Action*: the set of role, time, and environment of subject when it applies to access;
- (vii) *Structured document (SDoc)*: the document is consisted of chapters, sections, fonts, tables, and other elements, including the syntax and semantic of layout;
- (viii) *Operation type (AT)*: what a subject could operate on an object. Here $\forall at \in AT$, and $at = (r, e, a, w)$. The four symbols mean read, execute, append, and write, respectively. The actual operations such as create, delete, and view could be abstracted to them;
- (ix) *Permission (P)*: describes the authorized operation of a subject on a specific object, including the objects and operation type. Here $P = AT, Object$;
- (x) *Action (Subject \rightarrow Action)*: a mapping between subject and action, by which a subject's action information could be obtained.

Definition. object is the finest granularity that a subject could access. It is atomic and unified in security requirement. One or more objects could combine a subdocument, and one or more subdocuments could combine a document. We denote the set of objects as *Object*, and in our model it is equal to S_o of the structured document.

In order to solve the fine-grained access control problem of structured document, we extend the model in [9] to 6-tuple (*Subject, Action, SDoc, AT, P, and Object*). The subject's accessing requirement on object of structured document *SDoc* for operation type *AT* is denoted as $Req = (Subject, AT, SDoc, and Object)$.

3.3. The Access Control Protocol. The structured documents should be described in sections, pages, or even elements to satisfy the requirement for fine-grained access control. In order to implement our model, we define the corresponding protocols, including document creation, permission specification, and authority management. The symbols details are shown as follows.

Detailed Descriptions of Symbols

- Sid: subject's/user's ID,
- Oid: object's ID,
- t_s : subject's temporal state,
- e_s : subject's environmental state,
- T_s : starting time of policy,
- T_e : end time of policy,
- k_o : the object's key symmetry encryption, it is generally encrypted under the subject's private key,
- at: operation type,
- pd: the policy's description, it could be described in XML. Our paper will not discuss it in detail due to the paper length,
- Sk_A : A's private key.

The *NAME* denotes the abstract data types, including the action, role, time, environment, and structured document. The *ROLES* denotes the set of roles. *TSTATES* denotes the set of temporal states. *ESTATE* denotes the set of environmental states. *OBJECTS* denotes the set of objects. *POLICYS* denotes the set of policies. We give the functions for protocol in λ -notation as follows.

- (i) GenSDoc: content provider generates the structured document *SDoc* in specified format (format) by document-packaged tools.

$GenSDoc (contents: NAME; out SDoc: NAME) \triangleleft$
 $SDoc = format (contents); \triangleright$

- (ii) Define Object: user divides the *SDoc* into sections according to their requirement for access control.

$Define Object (SDoc: NAME; out OBJECTS: NAME) \triangleleft$
 $if object \notin OBJECTS then OBJECTS' = OBJECTS \cup \{ object \} \triangleright$

- (iii) Assign Policy: user assigns the policy to the divided *SDoc* and submits the result to the policy server, where $Policy = \{P_1, P_2, \dots, P_k\}$.

$Assign Policy (Object, environment, temporal, at, key, Ts, Te, pd: NAME; out Policy: NAME) \triangleleft$
 $if policy \notin Policy$
 $then policy.role = role$
 $policy.temporal = temporal$
 $policy.environment = environment$
 $policy.object = Object$
 $policy.at = at$
 $policy.k = key$
 $policy.Ts = Ts$
 $policy.Te = Te$
 $policy.pd = pd$
 $POLICYS' = POLICYS \cup \{ policy \}$
 $POLICYS = POLICYS' \triangleright$

- (iv) Verifyid: verify the user's identity and certificate; if legal, then return "True," otherwise return "False."

$Verifyid (user, certification: NAME; out result: BOOLEAN) \triangleleft$
 $\triangleleft result = (user \in U) (isvalid (certification)) \triangleright$

- (v) GetReq: obtain the user's requirement *req*.

$GetReq (user, object, SDoc, at:NAME; out req: NAME) \triangleleft$
 $req.subject = user$
 $req.at = at$
 $req.Object = object \triangleright$

- (vi) GetAction: obtain the information including role R , environment E , and time T for the policy server's further judgment. The information received is $(Role, E, T, \text{and Object})$.

GetAction (*req, temporal, environment: NAME; out action: NAME*) \triangleleft
action.role = *req.subject*
action.temporal = *temporal*
action.environment = *environment* \triangleright

- (vii) Verifyt: when the user's requirement req and action information $action$ are obtained, the policy server will validate the user's temporal information by this function. If valid, then return "True," otherwise return "False."

Verifyt (*action, validt: NAME; out result: BOOLEAN*) \triangleleft *action.temporal* \in *TSTATES; validt* \subseteq *TSTATES*
result = $(\exists t_1, t_2 \in \text{validt} \cdot t_1 \leq \text{action.temporal} \leq t_2)$ \triangleright

- (viii) Verifye: when the user's requirement req and action information $action$ are obtained, the policy server will validate the user's environmental information by this function. If valid, then return "True," otherwise return "False."

Verifye (*action, valide: NAME; out result: BOOLEAN*) \triangleleft
action.environment \in *ESTATES; valide* \subseteq *ESTATES*
result = $(\exists e_1, e_2 \in \text{valide} \cdot e_1 \leq \text{action.environment} \leq e_2)$ \triangleright

- (ix) Verifyr: when the user's requirement req and action information $action$ are obtained, the policy server will validate the user's role by this function. If valid, then return "True," otherwise return "False."

Verifyr (*action, validr: NAME; out result: BOOLEAN*) \triangleleft
action.role \in *ROLES; validr* \subseteq *ROLES*
result = $(\exists r_1, r_2 \in \text{validr} \cdot r_1 \leq \text{action.role} \leq r_2)$ \triangleright

- (x) JusUsage: the policy server will search the corresponding policy for user according to his action and requirement.

JusUsage (*action, req: NAME; out poicy: NAME*) \triangleleft
policy' = $(\exists \text{policy}' \in \text{POLICY} \cdot (\text{policy}'.\text{object} = \text{req.object} \wedge \text{policy}'.\text{at} = \text{req.at}))$
if verifyr (*action, policy'.role*) *then*
if verify (*action, policy'.temporal*) *then*
if verifye (*action, policy'.environment*) *then*
policy = *policy'* \triangleright

(1) *The Protocol for Structured Documents Creation.* By using corresponding tools, the owner of the resource will package his text, video, audio, picture, and 3D objects into one digital work and assign policies according to their own requirement (see Figure 3). The steps of the protocol are explained as follows.

Step 1. CP \rightarrow CS the content provider CP creates the digital work and submits it to the content server.

Step 1-1 Generate the digital content EC.

Step 1-1-1 Call the function GenSDoc() to generate the digital content C in corresponding format.

Step 1-1-2 Call the function DefineObject() to divide the digital content into objects $\{o_1, \dots, o_m\}$ according to the requirement.

Step 1-1-3 Generate the random number N_{CP} .

Step 1-1-4 Encrypt the objects o_1, \dots, o_m obtained from Step1-1-2 under the session key k_1, \dots, k_m exchanged before. The cipher text is $C' = E_{k_1, \dots, k_m}(o_1, \dots, o_m)$.

Step 1-1-5 Generate the signature $\text{Sig}_{\text{Sk}_{CP}}$ ($\text{Hash}(C' \parallel N_{CP})$).

Step 1-1-6 Generate the final digital content $EC = \{C', \text{Sig}_{\text{Sk}_{CP}}(\text{Hash}(C' \parallel N_{CP})), N_{CP}\}$.

Step 1-2 Submit the digital content EC to the content server CS.

Step 2. CS \rightarrow CP content server returns the result of new content submission to content provider.

Step 2-1 Generate the data for result.

Step 2-1-1 Verify the information by using the CP's public key. If failed, go to Step 5.

Step 2-1-2 Save the digital content EC, and add the new content information to the publishing list. Generate the basic data Re for result.

Step 2-1-3 Generate random number N_C .

Step 2-1-4 Generate the signature Sig_{Sk_C} ($\text{Hash}(Re \parallel N_C \parallel N_{CP})$).

Step 2-1-5 Generate the final data for returning result $Result = (Re, \text{Sig}_{\text{Sk}_C}(\text{Hash}(Re \parallel N_C \parallel N_{CP})), N_C, N_{CP})$.

Step 2-2 Send *Result* to content provider CP.

Step 3. CP \rightarrow PS the content provider CP creates the policies and submits it to the policy server.

Step 3-1 Generate the policy requiring data.

Step 3-1-1 Generate the policy descriptions Pd_1, Pd_2, \dots, Pd_m corresponding with the objects of content C .

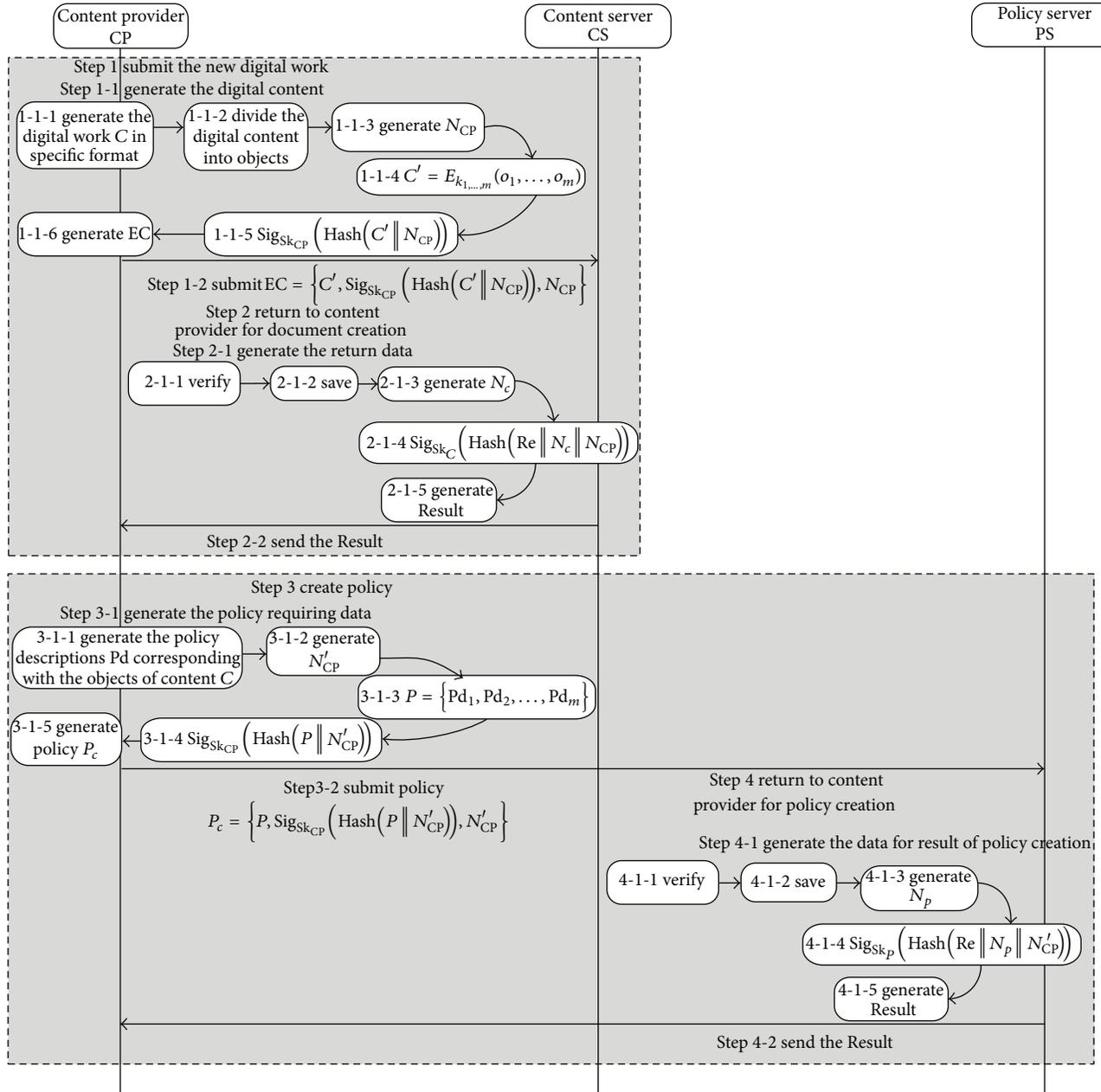


FIGURE 3: The process of structured documents creation.

Step 3-1-2 Generate the random number N'_{CP} .

Step 3-1-3 Policy $P = \{Pd_1, Pd_2, \dots, Pd_m\}$.

Step 3-1-4 Generate the signature $\text{Sig}_{\text{Sk}_{CP}}(\text{Hash}(P \parallel N'_{CP}))$.

Step 3-1-5 Generate the user's policy $P_c = \{P, \text{Sig}_{\text{Sk}_{CP}}(\text{Hash}(P \parallel N'_{CP})), N'_{CP}\}$.

Step 3-2 Submit the P_c to policy server.

Step 4. PS \rightarrow CP policy server returns result to content provider for policy creation.

Step 4-1 Generate the data for result of policy creation.

Step 4-1-1 Verify the information by using the CP's public key. If failed, go to Step 5.

Step 4-1-2 Save the policy, and generate the basic data Re for result.

Step 4-1-3 Generate the random number N_p .

Step 4-1-4 Generate the signature $\text{Sig}_{\text{Sk}_p}(\text{Hash}(\text{Re} \parallel N_p \parallel N'_{CP}))$.

Step 4-1-5 Generate the final data for result $\text{Result} = (\text{re}, \text{Sig}_{\text{Sk}_p}(\text{Hash}(\text{Re} \parallel N_p \parallel N'_{CP})), N_p, N'_{CP})$.

Step 4-2 Send Result to content provider CP.

Step 5. End of Creation.

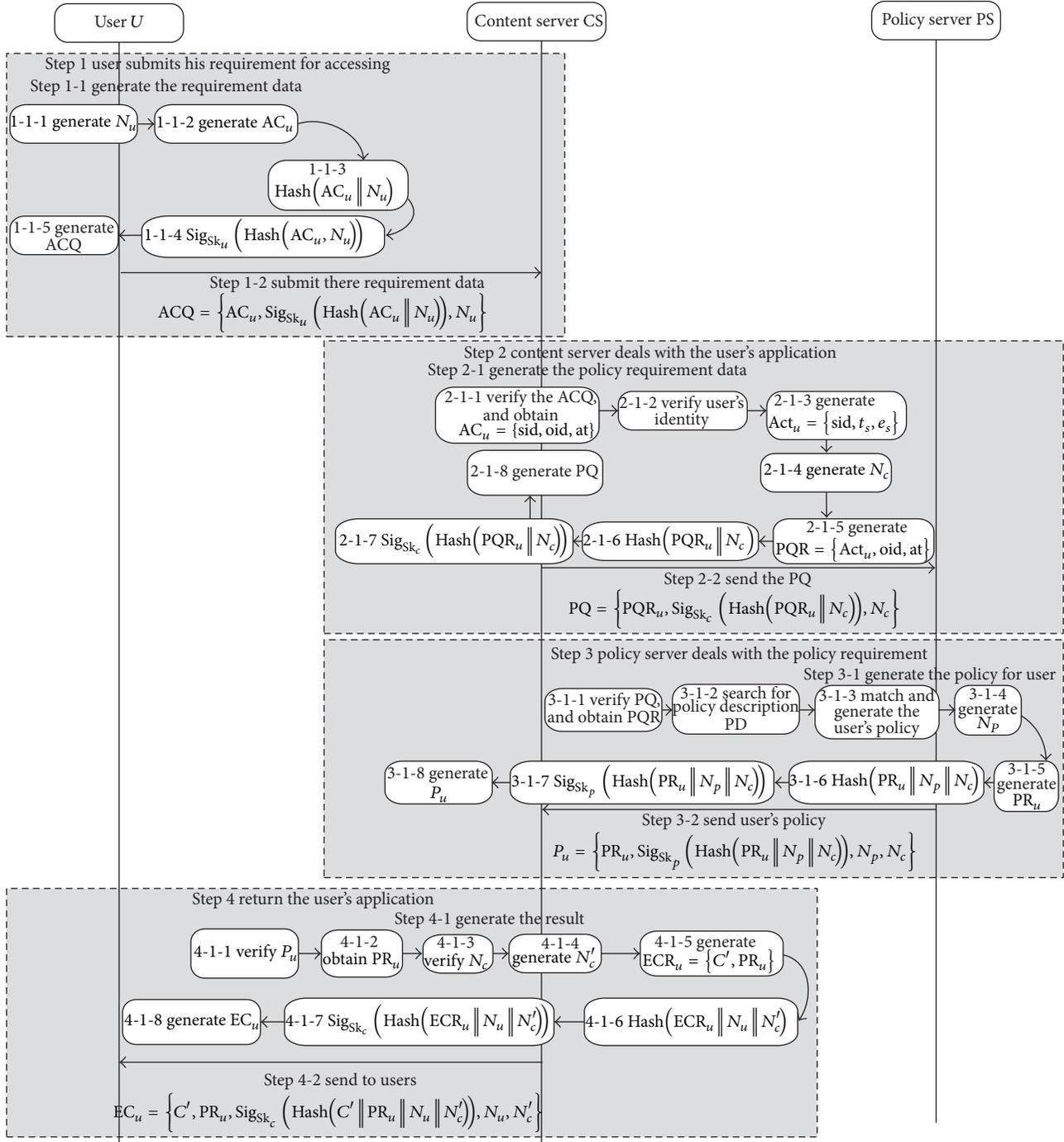


FIGURE 4: The process of users' authorization.

(2) *The Protocol of Authorization.* User U registers the information to the system and obtains a role that can access the system. When U needs to access the resource, content server and policy server will return the corresponding digital content and policy according to their role, environment, and temporal states (see Figure 4). The steps of the protocol are explained below.

Step 1. $U \rightarrow CS$ user U submits the requirement for accessing a digital content to content server according to the resource list of the system.

Step 1-1 Generate the requirement data.

Step 1-1-1 Generate the random number N_u .

Step 1-2-1 Generate the requirement basic data $AC_u = \{sid, oid, op\}$.
 Step 1-3-1 Generate $Hash(sid || oid || op || N_u)$.
 Step 1-4-1 Generate the signature Sig_{Sk_u} ($Hash(sid || oid || op || N_u)$).
 Step 1-5-1 Generate the $ACQ = \{AC_u, Sig_{Sk_u}$ ($Hash(sid || oid || op || N_u)$), $N_u\}$.

Step 1-2 Submit the ACQ to content server.

Step 2. CS \rightarrow PS content server in data center deals with the user's requirement.

Step 2-1 Generate the policy requirement data.

Step 2-1-1 Analyze the user's requirement data ACQ , and verify the data by using the user's public key. If failed, go to Step 5.
 Step 2-1-2 Call function $GetReq()$ to obtain the user's requirement req , and verify the legitimacy of user's identity; if failed, go to Step 5.
 Step 2-1-3 Call function $GetAction()$ to generate the user's action information $Act_u = \{sid, t_s, e_s\}$.
 Step 2-1-4 Generate the random number N_c .
 Step 2-1-5 Generate the user's policy requirement basic data $PQR = \{Act_u, oid, op\}$.
 Step 2-1-6 Generate $PQ_Hash = Hash(sid || t_s || e_s || oid || at || N_c)$.
 Step 2-1-7 Generate signature Sig_{Sk_c} (PQ_Hash).
 Step 2-1-8 Generate the policy requirement data $PQ = \{PQR_u, Sig_{Sk_c}$ (PQ_Hash), $N_c\}$.

Step 2-2 Send PQ to the policy server.

Step 3. PS \rightarrow CS the policy server in data center deals with the user's policy requirement.

Step 3-1 Generate the policy for user.

Step 3-1-1 Analyze the policy requirement data PQR , and verify the data by using the CP's public key. If failed go to Step 5.
 Step 3-1-2 Obtain the information sid, oid, at, t_s, e_s from the requirement, and search in the policy database for the corresponding policy description PD according to the information.
 Step 3-1-3 Call function $JusUsage()$ to match the user's requirement and generate the user's policy.
 Step 3-1-4 Generate random number N_p .
 Step 3-1-5 Generate the user's policy basic data $PR_u = \{sid, oid, at, T_s, T_e, k_o, Pd\}$.
 Step 3-1-6 Generate $PR_Hash = Hash(PR_u || N_p || N_c)$.
 Step 3-1-7 Generate signature Sig_{Sk_p} (PR_Hash).
 Step 3-1-8 Generate the policy data $P_u = \{PR_u, Sig_{Sk_p}$ (PR_Hash), $N_p, N_c\}$.

Step 3-2 Send P_u to content server.

Step 4. CS \rightarrow U data center returns the result to user.

Step 4-1 Generate the result for return

Step 4-1-1 Analyze the policy data P_u , and verify the data by using the PS's public key. If failed, go to Step 5.

Step 4-1-2 Obtain the policy basic data PR_u .

Step 4-1-3 Verify the random number N_c . If failed, go to Step 5.

Step 4-1-4 Generate the random data N'_c .

Step 4-1-5 Generate the basic data $ECR = \{C', PR_u\}$ for return.

Step 4-1-6 Generate $ECR_Hash = Hash(C' || PR_u || N'_c)$.

Step 4-1-7 Generate signature Sig_{Sk_c} (ECR_Hash).

Step 4-1-8 Generate $EC_u = \{ECR, Sig_{Sk_c}$ (ECR_Hash), $N'_c, N_u\}$.

Step 4-2 Return the EC_u to the user.

Step 5. End of authorization.

(3) *The Protocol for User to Read Structured Document.* Generally, the user could read the digital content by using the specific readers. When the user obtains the final return data ECR , his reader will verify the signature and random number. If succeeded, it will analyze the data in three steps. Firstly, the reader gets the T_s and T_e from PR_u and verifies this policy. Secondly, it obtains the session key k and cipher text C' . Policy server could encrypt the key k with the user's public key, and the reader can decrypt it with the user's private key. Finally, the reader will analyze the permission description Pd and compute $D_{k'}(C')$ to obtain the corresponding permission. After obtaining the permission description, the reader will present the corresponding digital content to the user.

(4) *Security Analysis.* Because the digital content has been encrypted before submission, our protocol ensures the confidentiality of the data. The requirement and data packages transferred among the users, content providers, content servers, and policy servers have been signed. Therefore the integrity and nonrepudiation can be ensured. In addition, the random numbers have been used during the transmission to withstand the replaying attack and man-in-the-middle attack.

4. Implementation of Our Mechanism

Our system is under Windows based on C/S model, and the framework for implementation is shown in Figure 5. The server side consisted of the content server and policy server and the client side includes user interface and creator interface. The server side consists of user's requirement analysis module, verification module, action obtaining module, permission assignment module, and the other universal modules like cryptography and data transforming modules. The user's requirement analysis module could analyze the data

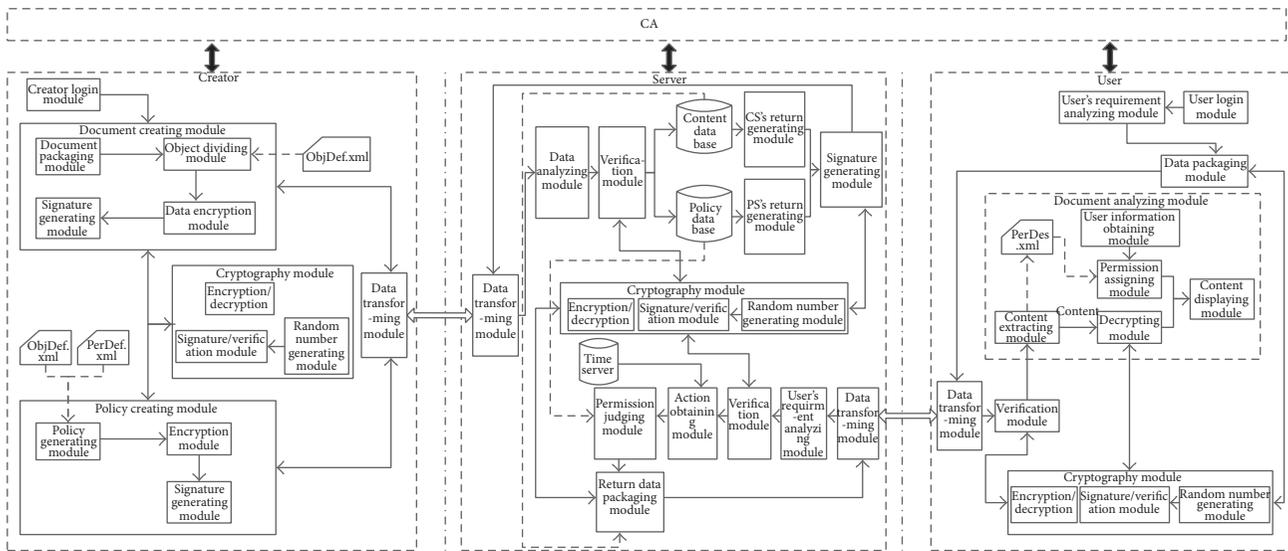


FIGURE 5: Framework for implementation.

packages from the users and divide them into the data stream for document creating, policy creating, or data accessing. The verification module will verify the integrity and the random number by calling the cryptography module. The action obtaining module will get the user's role, temporal state, and environmental state together to generate the action. The user's role is obtained according to *sid*. The temporal state from the time server and the environmental state includes the network, physical location, platform, or the software. In our system, we take the network and software for experiment, and we will improve the other factors in the future. The network information is obtained from the user's data packages and the software information is initialized at the beginning. The client will send the Hash of the client software to server when accessing the resource. The permission assignment module makes the decision according to the policy data base. The format of this data base is (sid, oid, at, t_s , e_s , *perDes.xml*). Besides, the common modules like data based managing and storing will not be described here.

The client side includes two types, creator and user. The creator mode includes the document packaging module, object dividing module, encryption module, signature generating module, policy generating module, and some general modules for cryptography and data transforming. Document packaging module will compress and package the resource uploaded by the creator and generate the structured file in certain format. The object dividing and encrypting module will define the objects and encrypt them according to the file *objDef.xml*. The policy generating module will return the permission describing file *perDes.xml* based on the files *objDef.xml* and *perDef.xml*. The user mode consists of user's requirement analyzing module, document analyzing module, user login module, and some general modules for cryptography and data transforming. The user login module will get the username and password. And, then, it will decide the user's role information and pass it the system.

The user's requirement analyzing module will analyze the user's requirement and abstract the sid, oid, at to the requirement basic data for system. Document analyzing module includes the submodules like content extracting, decrypting, user information obtaining, and permission assigning. Firstly, this module extracts the content and policy and then gets the user's corresponding information, for example, role and time network address, according to the policy description. Finally, it will display the right data and permission for user.

The XML files mentioned above will be introduced as follows and the analysis of them is implemented based on the ParseXML.

(1) *objDef.xml* describes the division of objects, including the object name, object ID, parent object, and content. For example, an object named "Introduction," whose ID is "O11" and parent object is "O1," could be described in Algorithm 1.

(2) *perDef.xml* describes the permission definition of the certain object, including the object ID, operation type, and action description. For example, the situation that role A can read and write the object O11 with the IP from 172.16.1.5 to 172.16.1.35 between 8:00 am and 10:00 am could be described in Algorithm 2.

(3) *perDes.xml* describes the permission information for analysis of client, including the sid, oid, the begin and end time, key information, and the description of permission. According to (1) and (2), the *perDes.xml* for "User_A" to access "O11" is as Algorithm 3.

The tag `<ko>` is defined to describe the key information. This key is encrypted by the key embedded in the client and the user cannot obtain it, so the user can obtain the right data only by the certain client.

By testing, the system has realized the functions of data encryption, data decryption, and access control according to the temporal state and network address. Also, the users can

```

<Obj>
  <ObjName>Introduction</ObjName>
  <ObjID>O11</ObjID>
  <ObjFather>O1</ObjFather>
  <ObjCon> Introduction The paper XXXXX</ObjCon>
</Obj>

```

ALGORITHM 1

```

<Permission>
  <Obj>
    <ObjID>O11</ObjID>
  </Obj>
  <Action>
    <Role>A</Role>
    <Time>
      <after>8:00am</after>
      <before>10:00am</before>
    </Time>
    <Environment>
      <IP>
        <from>172.16.1.5</from>
        <to>172.16.1.35</to>
      </IP>
    </Environment>
  </Action>
  <PerDes>read&write</PerDes>
</Permission>

```

ALGORITHM 2

```

<sid>User_A</sid>
<oid> O11</oid>
<Ts>2014-1-1</Ts>
<Te>2014-12-31</Te>
<ko> * * * * * </ko>
<Permission>
  <Action>
    <Role>A</Role>
    <Time>
      <after>8:00am</after>
      <before>10:00am</before>
    </Time>
    <Environment>
      <IP>
        <from>172.16.1.5</from>
        <to>172.16.1.35</to>
      </IP>
    </Environment>
  </Action>
  <PerDes>read&write</PerDes>
</Permission>

```

ALGORITHM 3

only use the data on the certain client equipment. Because the client will do the integrity authentication by communicating with the server, the user cannot access by tampering the client and cannot decrypt the data without the client. The keys of our system are managed by the certificate authentication (CA) in Figure 5. The CA realizes the functions of certificate creating, distributing, and managing. The symmetric keys will be generated by the creating clients and managed by the servers. Our main focus, however, is on the designing and implementation of the access control mechanism for structured document based on ABAC (action-based access control); the new technology to manage the keys, such as key derivation functions or key hierarchy, will be discussed in the future.

The implemented results of the system are shown in Figure 6. The user should login the system (see Figure 6(a)) and choose the role (see Figure 6(b)). After login and role assignment, the user can create the resource by "Document Creation" (see Figure 6(c)) or access the resource by "User Reader" (see Figure 6(d)). When creating a new resource, the creator can choose the resources for packaged, decide the algorithm for encryption, and upload the XML files for definitions of object and permission.

Let us make an assumption that a creator has created the resource called "Exercise 1" including the object O1 and object O2. O1 is the questions parts and O2 is the answers parts. User A could apply for permission to access in the classroom (network address: 172.16.66.5–172.16.66.90) and they can read O1 and O2 between 8:00 am and 10:00 am; otherwise, he can only read O1. If the user is out of the classroom, the accessing requirements are always illegal. Results of some experiments are shown in Figure 7. The four situations of the illegal time and legal network address, the legal time and illegal network address, the illegal time and network address, and the legal time and network address are shown in Figures 7(a), 7(b), 7(c), and 7(d), respectively.

5. The Properties of Action-Based Access Control for Structured Documents

In this section, we will make the comparison between the current models and ours. The result is shown in Table 1. GB-RBAC [11] and H-RBAC [12] give the group-based and attribute-based access control for distributed computing and cloud computing, respectively. Nevertheless, those models are insufficient for the complicated network without temporal and environmental constraints. GTRBAC [1] extends the temporal constraint of users and allows expressing periodic as well as duration constraints on roles, user-role assignments, and role-permission assignments. LRBAC (location-aware RBAC) [2] has extended RBAC to incorporate the notion of location, but the temporal constraint was ignored. TLRBAC [3] and RBAC model based on space, time, and scale have introduced the temporal state and simple environmental state, but the environmental state only can describe the single physical location which is not enough. In order to describe the permission in more details, paper [7] presents

the attribute-based access control and its formal definition. It abstracts temporal state and environmental state as attributes. The permission assignment is based on attributes. However, the description of attributes is too complex, and the concept role is weakened. Action-based access control [9] integrates the temporal state, environmental state, and role and discusses the description of environment in detail. Therefore, it could have wide application in the MLS systems. In the complicated network environment, flexibility and extensibility have become the vital requirements of research on access control. Paper [13] proposes a trust and role based access control framework in infrastructure-centric environment. The scheme is flexible and scalable. Paper [14] presents a dynamic access control model based on description logic (DL). It could assign permission to a user according to his role in a given context. Moreover, composed context is supported. However, all the models mentioned above have not paid enough attention to the fine-grained access control, so they could not accommodate the requirement for structured document. According to the description of XML, paper [10] gives a fine-grained mandatory access control model and its rules for XML documents. Nevertheless, the model's only authority is based on the users' security level and that of XML elements or attributes. Thus it is not suitable for the multielement access control in the current complicated network.

We propose an action-based fine-grained access control for structured document by analyzing the advantages and disadvantages of the models and mechanisms above. Our mechanism is extended from ABAC [9]. It could implement fine-grained description of the structured document and the authority is based on action in the complicated network.

6. Conclusions

To solve the problem of fine-grained access control for structured documents in the current complicated network, this paper proposes an action-based access control mechanism. By defining the objective describing model, it could support the permission management on chapters, pages, sections, words, and pictures of structured documents. The corresponding protocols are given to cover different stages in the lifecycle of structured document, including document creation, users' authorization, and user's usage. Other requirements such as preview, combination, and permission specification are accommodated as well. Meanwhile, the confidentiality, integrity, and nonrepudiation are ensured by our mechanism too. In the future research, we intend to implement the access control mechanism with considerations of more factors and address the problem of dynamic adjustment and combination for policy.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

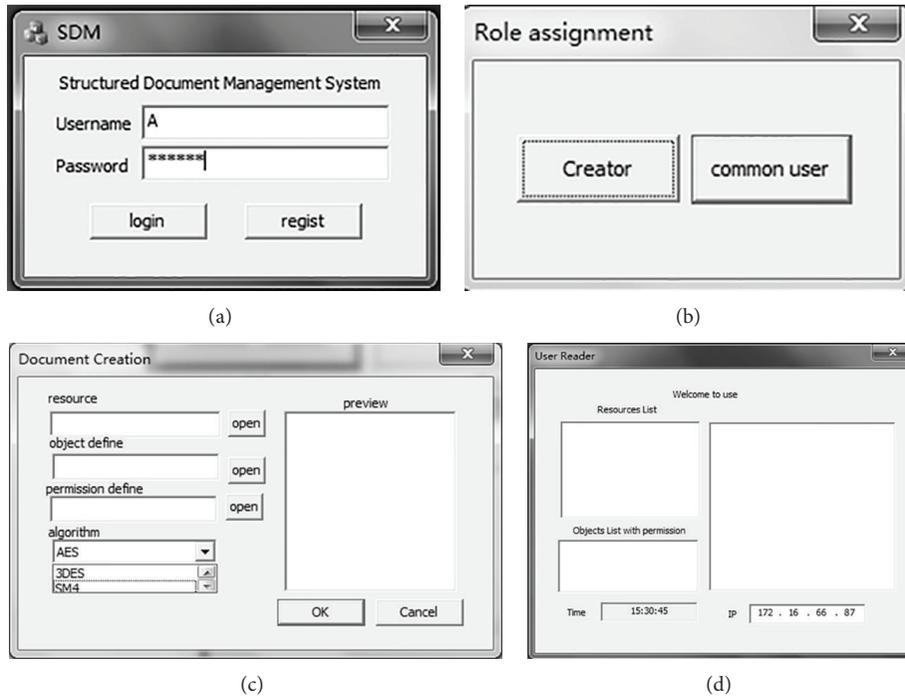


FIGURE 6: The user interface of the system.

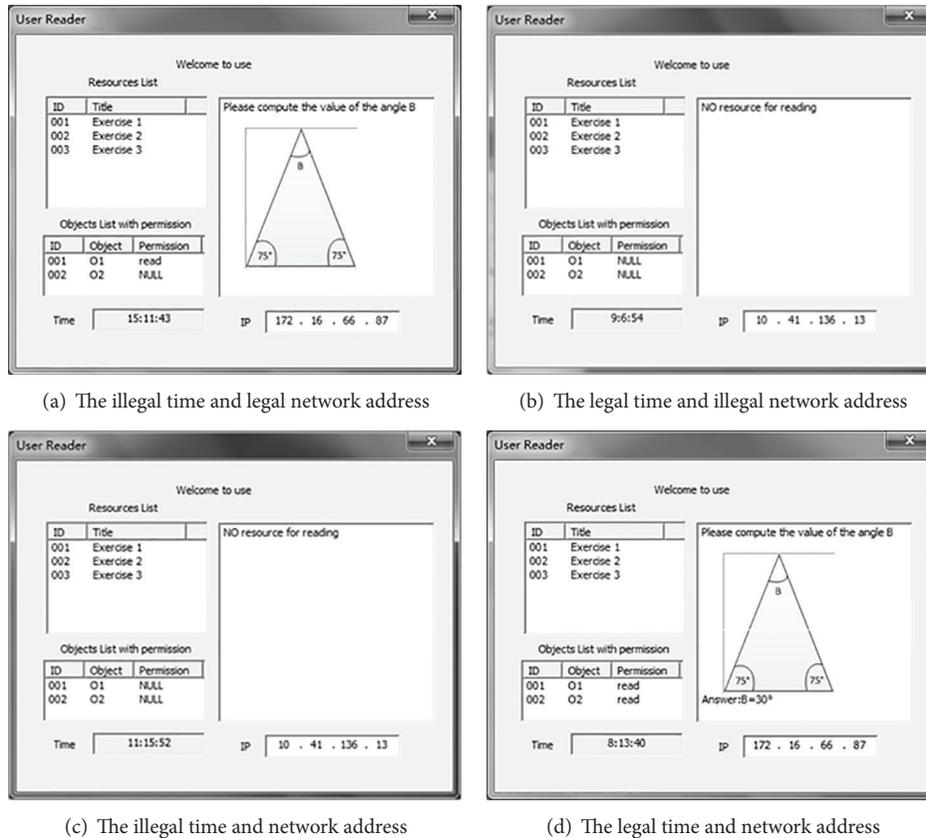


FIGURE 7: The results of experiments.

TABLE 1: Comparison between our model and current models.

Model	Role	Temporal state	Property		
			Environmental state	Fine-grained	Permission specification
GB-RBAC [11]	YES	NO	NO	NO	NO
H-RBAC [12]	YES	NO	NO	NO	NO
GTRBAC [1]	YES	YES	NO	NO	NO
LRBAC [2]	YES	NO	PART	NO	NO
TLRBAC [3]	YES	YES	PART	NO	NO
Reference [4]	YES	YES	PART	NO	NO
Attribute-based access control [7]	NO	YES	YES	NO	NO
Action-based access control [9]	YES	YES	YES	NO	NO
ITRBAC [13]	YES	NO	PART	NO	NO
Reference [14]	YES	NO	PART	NO	NO
Reference [10]	YES	NO	NO	PART	NO
Our model	YES	YES	YES	YES	YES

Acknowledgments

The authors would like to thank those anonymous reviewers for their suggestions and comments. This work is supported by foundations as follows: The National Natural Science Foundation of China (61170251); the Beijing Natural Science Foundation (4102056); the Major Science and Technology Project of Press and Publication-Research and Development (1681300000119); and the National High Technology Research and Development Program of China (863 Program) (2012AA013102).

References

- [1] J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "A generalized temporal role-based access control model," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 1, pp. 4–23, 2005.
- [2] R. Indrakshi, K. Mahendra, and Y. Lijun, "LRBAC: a location-aware role-based access control model," in *Proceedings of the 2nd International Conference on Information Systems Security (ICISS '06)*, pp. 147–161, Springer.
- [3] H.-C. Chen, S.-J. Wang, J.-H. Wen, and C. Chen, "Temporal and location-based RBAC model," in *Proceedings of the 5th International Joint Conference on INC, IMS and IDC*, pp. 2111–2116, IEEE Computer Society, Seoul, Korea, August 2009.
- [4] Y.-J. Zhang and D.-G. Feng, "A role-based access control model based on space, time and scale," *Journal of Computer Research and Development*, vol. 47, no. 7, pp. 1252–1260, 2010.
- [5] B. Shafiq, J. B. D. Joshi, E. Bertino, and A. Ghafoor, "Secure inter-operation in a multidomain environment employing RBAC policies," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1557–1577, 2005.
- [6] J. Park and R. Sandhu, "Towards usage control models: beyond traditional access control," in *Proceedings of 7th ACM Symposium on Access Control Models and Technologies (SACMAT '02)*, pp. 57–64, Monterey, Calif, USA, June 2002.
- [7] X. Jin, R. Krishnan, and R. Sandhu, "A unified attribute-based access control model covering DAC, MAC and RBAC," *Lecture Notes in Computer Science*, vol. 7371, pp. 41–55, 2012.
- [8] S. Chakraborty and I. Ray, "TrustBAC—integrating trust relationships into the RBAC model for access control in open systems," in *Proceedings of the 11th ACM Symposium on Access Control Models and Technologies (SACMAT '06)*, pp. 49–58, June 2006.
- [9] F. H. Li, W. Wang, J. F. Ma, and X. Y. Liang, "Action-based access control model and administration of actions," *Acta Electronica Sinica*, vol. 36, no. 10, pp. 1881–1890, 2008.
- [10] L. Li, Y.Z. He, and D.-G. Feng, "A fine-grained mandatory access control model for xml documents," *Journal of Software*, vol. 15, no. 10, pp. 1528–1537, 2004.
- [11] Q. Li, X. Zhang, M. Xu, and J. Wu, "Towards secure dynamic collaborations with group-based RBAC model," *Computers and Security*, vol. 28, no. 5, pp. 260–275, 2009.
- [12] D. C. Li, C. Liu, and B. S. Liu, "H-RBAC: a hierarchical access control model for SaaS systems," *International Journal of Modern Education and Computer Science*, vol. 5, pp. 47–53, 2011.
- [13] R. Yang, C. Lin, Y. Jiang, and X. Chu, "Trust based access control in infrastructure-centric environment," in *Proceedings of the IEEE International Conference on Communications (ICC '11)*, IEEE, Kyoto, Japan, June 2011.
- [14] N. Boustia and A. Mokhtari, "A dynamic access control model," *Applied Intelligence*, vol. 36, no. 1, pp. 190–207, 2012.

Research Article

Quality of Protection Evaluation of Security Mechanisms

Bogdan Ksiezopolski,^{1,2} Tomasz Zurek,¹ and Michail Mokkas²

¹ *Institute of Computer Science, Maria Curie-Skłodowska University, Plac Marii Curie-Skłodowskiej 5, 20-031 Lublin, Poland*

² *Polish-Japanese Institute of Information Technology, Koszykowa 86, 02-008 Warsaw, Poland*

Correspondence should be addressed to Bogdan Ksiezopolski; bogdan.ksiezopolski@acm.org

Received 10 March 2014; Revised 18 June 2014; Accepted 19 June 2014; Published 17 July 2014

Academic Editor: Fei Yu

Copyright © 2014 Bogdan Ksiezopolski et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recent research indicates that during the design of teleinformatic system the tradeoff between the systems performance and the system protection should be made. The traditional approach assumes that the best way is to apply the strongest possible security measures. Unfortunately, the overestimation of security measures can lead to the unreasonable increase of system load. This is especially important in multimedia systems where the performance has critical character. In many cases determination of the required level of protection and adjustment of some security measures to these requirements increase system efficiency. Such an approach is achieved by means of the quality of protection models where the security measures are evaluated according to their influence on the system security. In the paper, we propose a model for QoP evaluation of security mechanisms. Owing to this model, one can quantify the influence of particular security mechanisms on ensuring security attributes. The methodology of our model preparation is described and based on it the case study analysis is presented. We support our method by the tool where the models can be defined and QoP evaluation can be performed. Finally, we have modelled TLS cryptographic protocol and presented the QoP security mechanisms evaluation for the selected versions of this protocol.

1. Introduction

Balancing security against performance in IT systems is one of the important issues to be solved. The traditional approach assumes that the best way is to apply the strongest possible security measures, which makes the system as secure as possible. Unfortunately, such reasoning can lead to the overestimation of security measures which causes an unreasonable increase in the system load [1, 2]. This problem is especially important in multimedia systems. The example from [2] may relate to the audio/video streaming as the real-time service. This teleconference can be protected by the VPN data transmission which will be accomplished by the TLS tunneling. After the service requirements analyses one can assign different versions of the protocol depending on the level of protection. The selection is presented in Table 1. In the first version one chooses the RC2-CBC algorithm and MD5 hash function. In the second version one selects the strongest symmetric algorithm (DES-CBC) and hash function with the longest digest (SHA1). In the third version one selects

the symmetric algorithm with the key 3DES-CBC longer than the one selected in the second version.

In [2] authors checked how the security mechanism influences the efficiency of the peers during the video teleconference. The speed of transmitting data in the video conference which was secured by the VPN connection was checked. The results are presented in Table 2. The required quality of the video conference can be guaranteed only if we can transmit 240 KB/s and simultaneously receive the same amount of data. The transfer of the required level (480 KB/s) is guaranteed by the first version of the protocol (low). For the second version (medium) it is equal to the required bit rate. The third version of the protocol, which accomplished the VPN connection on the high level, cannot make the video conference of the required quality. The presented results show that overestimation of security mechanisms during data transmission leads to the decreasing efficiency of the devices from which the transmission is accomplished.

The system performance is also important in the systems with limited resources, such as wireless system networks

TABLE 1: Selection of the specific cryptographic algorithms.

Ciphers
<i>Version 1—low</i>
RC2-CBC + MD5
<i>Version 2—medium</i>
DES-CBC + SHA1
<i>Version 3—high</i>
3DES-CBC + SHA1

TABLE 2: The bit rate of VPN connections.

<i>Version 1—low</i>	
Bit rate	501 KB/s
<i>Version 2—medium</i>	
Bit rate	480 KB/s
<i>Version 3—high</i>	
Bit rate	453 KB/s

or mobile devices. Another example where such an analysis should be performed is cloud architecture. The latest research indicates that the three main barriers for using cloud computing are security, performance, and availability [3]. Unfortunately, when the strongest security mechanisms are used, system performance decreases and system availability is further influenced. The solution is using the QoP models. The latest results show [2, 4–6] that in many cases the better way is determination of the required level of protection and adjustment of some security measures to these requirements. Such an approach is achieved by means of the quality of protection models where the security measures are evaluated according to their influence on system security.

One of the most challenging issues in all QoP models is performing quality of protection evaluation of security mechanisms for the different versions of the cryptographic protocol (security policies). All of the approaches [4, 7–9] introduce different formulae which estimate the influence of security mechanisms for QoP, but they also have one significant limitation. These models can evaluate only these versions which were previously directly defined and described in an evaluation system. As directly defined scenarios we understand previously predefined models of configurations of security mechanisms which secure the IT processes. Unfortunately, defining all possible scenarios for all IT processes is very complex and in many cases is not feasible. The result of a nondefined scenario can be the situation in which security mechanisms of a specific IT process would not be evaluated; for example, adding a new security mechanism to an existing system may entail its lack of adjustment to any existing scenario and, consequently, the failure of its evaluation. The QoP evaluation of security mechanisms can be performed as part of risk analysis process or can be part of the decision support system, which in an adaptable way define appropriate configuration of security mechanisms. As a result of the lack of QoP evaluation of security mechanisms, the decision support system would not perform action adequate to

the situation. This limitation is especially important in real-time systems [2, 10].

In the presented paper, we introduce a model of QoP evaluation of security mechanisms. Our main contribution is that the QoP evaluation of security mechanisms can be performed for not directly defined configurations of security mechanisms. The additional contribution of the paper is implementing the security mechanisms evaluation tool (SMETool) which supports the presented method. This tool can be used either by researchers or by security engineers. The SMETool can be downloaded from the web page of the Quality of Protection Modelling Language Project [11].

The paper is organized as follows. In the second section the related work about QoP models is presented. In the third section the formal model definition is presented. In the fourth section the methodology of QoP evaluation of security mechanisms is described. The fifth section deals with the case study of QoP evaluation of security mechanisms, where the TLS handshake protocol is presented. Finally, section six includes the conclusions.

2. Related Work

In the literature the security adaptable models are introduced as the quality of protection (QoP) models [4, 7–9, 12–17]. These models were created for different purposes and have different features and limitations. The related research in this area is presented below.

Lindskog attempt to extend the security layers in a few quality of service (QoS) architectures [14]. Unfortunately, the descriptions of the methods are limited to the confidentiality of data and based on different configurations of the cryptographic modules. Ong et al. in [15] present the QoP mechanisms, which define security levels depending on security parameters. These parameters are as follows: key length, block length, and contents of an encrypted block of data. Schneck and Schwan [16] propose an adaptable protocol concentrating on the authentication. By means of this protocol, one can change the version of the authentication protocol which finally changes the parameters of the asymmetric and symmetric ciphers. Sun and Kumar [17] create the QoP models based on the vulnerability analysis which is represented by the attack trees. The leaves of the trees are described by means of the special metrics of security. These metrics are used for describing individual characteristics of the attack. In [4] Ksiezopolski and Kotulski introduce mechanisms for adaptable security which can be used for all security services. In this model the quality of protection depends on the risk level of the analysed processes. Luo et al. [8] provide the quality of protection analysis for the IP multimedia systems (IMS). This approach presents the IMS performance evaluation using Queuing Networks and Stochastic Petri Nets. LeMay et al. [13] create the adversary-driven, state-based system security evaluation, the method which quantitatively evaluates the strength of systems security. In [9] Petriu et al. present the performance analysis of security aspects in UML models. This approach takes as an input of a UML model of the system designed by

the UMLsec extension [18] of the UML modelling language. This UML model is annotated with the standard UML profile for schedulability, performance, and time and then analysed for performance. In [12] Ksiezopolski introduce the quality of protection modelling language (QoP-ML) which provides the modelling language for making abstraction of cryptographic protocols that put emphasis on the details concerning quality of protection. The intended use of QoP-ML is to represent the series of steps which are described as a cryptographic protocol. The QoP-ML introduced the multilevel [19, 20] protocol analysis that extends the possibility of describing the state of the cryptographic protocol. In [7] the authors present the impact of used security mechanisms in wireless local area networks for its quality of service. In this approach the QoP model is introduced which quantifies the benefits of security policies and demonstrates the relationship between QoS and QoP.

3. Model

Main aim of the model is to create a tool which helps to evaluate the quality of security protection of a given IT system. We are going to achieve it by evaluation of a set of security attributes of a given system, where as security attributes we understand various aspects of protection of a given system. At the beginning of the evaluation process we have a system which can be described by a set of facts. These facts represent all the elements of a given system. On the grounds of a set of facts we may, based on knowledge base, knowledge representation mechanism, and expert system-like forward chaining mechanism, infer a more general description of a given system and perform evaluation of the quality of protection of the analyzed system. The model is a semiformal tool which should allow to represent features of the analyzed system, as well as knowledge required to perform evaluation of the quality of protection of the system. The model also assumes the utilisation of the inference mechanisms which are a slightly modified version of expert systems like forward chaining mechanisms.

The goals of the presented model are as follows.

- (1) The QoP evaluation of security mechanisms can be performed for not directly defined scenarios.
- (2) Making quality of protection evaluation of all security mechanisms possible.
- (3) Analysis refers to all security attributes.
- (4) The model can be used for any QoP models.

3.1. Facts and Rules. We assume that the system behaviour modelled in one of the QoP models is represented by means of a set of propositions which we call facts:

$$F = \{f_1, f_2, f_3, \dots, f_n\}, \quad (1)$$

where f_1, \dots, f_n are the facts describing a system.

By means of the facts one can define any security mechanisms. We assume a set of operators $OP = \{\neg, \sim, \vee, \wedge, \Rightarrow, \rightarrow\}$, where

- (i) \neg is a classical (strong) negation;
- (ii) \sim is a negation as failure;
- (iii) \vee is a disjunction;
- (iv) \wedge is a conjunction;
- (v) \Rightarrow is a defeasible implication;
- (vi) \rightarrow is a strict implication.

Definition 1 (literals). Facts (negated in a strict way or non-negated) are literals. The set of all literals is $L = \{l_1, l_2, \dots, l_m\}$.

For example if $F = \{f_1, f_2\}$ is a set of facts then $L = \{f_1, f_2, \neg f_1, \neg f_2\}$ is a set of literals.

Definition 2 (case). A case is a model of an evaluated system represented by a set of literals $C = \{l_a, l_b, \dots, l_s, l_z, \dots\}$, which may be also expressed by a set of positive or negated facts: $C = \{f_a, f_b, \dots, \neg f_s, \neg f_z, \dots\}$.

Definition 3 (security attribute). Security attribute is an attribute which describes system behaviour in the case of information security requirements.

For example, one can enumerate the following security attributes [4, 21, 22]: integrity, confidentiality, authentication, availability, or anonymity. The security attributes (SA) set consists of an unlimited but finite number of security attributes. Each of them has its own evaluation value expressed by a positive integer number. Evaluation value represents the estimation of its security attribute. A security attribute may have a positive or negative character, in the sense that bigger value of security attribute evaluation may mean better (for positive) or worse (for negative) evaluation.

Definition 4 (rule). Rule is a formula in the following form:

$$\text{Conditions} \longrightarrow \text{Conclusion}, \quad (2)$$

where

- (i) Conditions is a list of rule conditions.
A list of conditions is in the form $wl_a \text{ func } wl_b \text{ func } \dots \text{ func } wl_d$, where func is one of the operators from the set $\{\vee, \wedge\}$, and $\{wl_a, wl_b, \dots, wl_d\}$ are the facts (nonnegated or negated by negation as failure). Only one kind of operators can be used in one rule.
- (ii) Conclusion is a rule conclusion in the form $\text{Conclusion} = (lx \wedge ly \wedge \dots)$, where $(lx \wedge ly \wedge \dots) \in L$.

Conditions may be negated by negation as failure and conclusions may be negated by classical negation. In the antecedent part of the rule, it is forbidden to use classical negation. In the consequent part of the rule, it is forbidden to use negation as failure. The set of rules is denoted as RF .

Rules allow us to represent relations between various facts, because in real life systems the existence of a chosen

feature causes the existence (or not) of some other features. They also allow us to express which facts are exclusive in the sense that the existence of one of them causes the nonexistence of others. It is important because it helps to preserve consistency of the model.

3.2. Evaluation Rules. The assessment of the security attributes of a given computer system is based on facts and evaluation rules.

Definition 5 (evaluation rule). Evaluation rules are formulae in the following form:

$$\text{Conditions} \implies \text{Inf}^V(\text{sa}), \quad (3)$$

where

- (i) Conditions is a list of rule conditions in the form $wl_a \text{ func } wl_b \text{ func } \dots wl_d$, where *func* are the operators from the set = $\{\vee, \wedge\}$ and $\{wl_a, wl_b, \dots, wl_d\}$ are facts (nonnegated or negated by negation as failure).
- (ii) Inf is a function changing value of evaluation of security attribute sa by adding value V (security influence) to the security attribute evaluation, when V is an integer number which represents the evaluation of a given security attribute.

Security attribute evaluation cannot be lower than 0. Special function $\text{Inf}^0(\text{sa})$ means that the security attribute sa evaluation is reduced to 0. This function will reduce sa to 0 regardless of sa, the current value. When this operator is used, it means that this sa is not guaranteed. The V value of sa will be equal to 0 disregarding other evaluation rules which could increase V value. This mechanism will be described more precisely later.

It is important to notice that in fact evaluation rules are not rules in a traditional sense of this term. They are conditionals in which the satisfaction of their conditions causes change of value of evaluation attribute.

We denote the set of evaluation rules as a ER. The example of the evaluation rules set ER is

$$\begin{aligned} f_1 \wedge f_2 &\implies \text{Inf}^{10}(\text{Confidentiality}), \\ f_3 \wedge \sim f_4 &\implies \text{Inf}^0(\text{Integrity}). \end{aligned} \quad (4)$$

The fulfillment of the evaluation rule conditions causes an appropriate change of the security attribute evaluation value. For example, when we have the above rules and we have the case $P = \{f_1, f_2, f_3\}$, we may conclude that the value of evaluation of security attribute Confidentiality should increase by 10 points and the value of evaluation of security attribute Integrity should decrease to 0 points.

The evaluation rule uses defeasible implication because such a rule may be defeated by another one.

Definition 6 (strict satisfaction of rule conditions). Rule conditions are satisfied in a strict way if positive (nonnegated) conditions are true and negative conditions (facts negated

with negation as failure) are false or it is impossible to conclude that they are true.

The example of the rule is

$$f_1 \wedge f_2 \wedge \sim f_3 \longrightarrow f_4. \quad (5)$$

From the above rule we may conclude that if f_1 and f_2 are true and f_3 is false (it is not declared, it cannot be concluded from other rules, and it is declared that $\neg f_3$ or there is a rule with the conclusion $\neg f_3$ and its conditions are satisfied), then f_4 is true.

3.2.1. Orders between Facts. It is easy to notice that the evaluation of complex systems requires building of a large set of rules, which should allow for evaluation of any real life system. It is also possible that such a set of rules may be, due to many reasons, incomplete in the sense that there may be facts which are not used in any rule and evaluation rule.

Such a situation may lead the evaluation process to misleading consequences which come from the lack of evaluation of potentially important facts. We can also notice that such new facts unpredicted in the rule base may be in a way connected to the other ones which are already regulated. They may, for example, represent better satisfaction of a condition of a chosen rule. On the other hand it is sometimes much easier to declare that, for example, fact f_1 means more than f_2 and may satisfy the condition of a chosen rule in a better way.

Based on the above, we assume the possibility of the declaration of orders between facts. The partial order $f_1 > f_2$ denotes that f_1 means more than f_2 and if there is a rule in which f_2 is one of the conditions and we know that f_1 is satisfied, then we may conclude that f_2 should also be satisfied (even if it is not literally true). Such an order represents better satisfaction of the rule condition. For example, if f_2 means cipher with the default key length, f_1 may denote cipher with the longer key length.

It is worth mentioning that these relations may not be the same for every security attribute. For example, a longer key length is better in the matter of confidentiality, but it is worse in the matter of efficiency. According to the above, we have to add to the order additional information about security attribute in which this reasoning concerns.

We also assume that the relation of order between facts is transitive:

$$\forall_{(X,Y,Z)} ((X > Y) \wedge (Y > Z) \longrightarrow (X > Z)). \quad (6)$$

We introduce the structure OF: $OF = \langle F, >_{SA} \rangle$, where $>_{SA}$ is a relation of strict partial order which represents preferences between various facts from the set F in the context of security attribute SA ($f_1 >_{sa} f_2$ denotes that f_1 means more than f_2 in the context of security attribute sa).

Definition 7 (unstrict satisfaction of rule conditions). Condition f_x of a given rule is satisfied in an unstrict way when f_x is not true, but

- (i) there is a fact f_y ;
- (ii) it is not known that $\neg f_x$ ($\sim \neg f_x$);
- (iii) we know that $f_y >_{SA} f_x$;
- (iv) reasoning concerns evaluation of security attribute SA.

Such kind of satisfaction of the condition of the rule we call unstrict satisfaction of rule conditions. The root of unstrict satisfaction of the rule lies in the so called a'fortiori reasoning (reasoning from more to less), which is commonly used in legal domain. The example of formalisation of such a way of reasoning is presented in [23].

Looking more generally at the above, we may notice that falsehood of the condition is not sufficient to assume that it is not satisfied. In other words, we have to distinguish truthfulness of the condition from its satisfaction. In the model we assume that false condition may be, in the above mentioned cases, treated as satisfied one. We also assume that it is not possible to treat true condition as unsatisfied.

Another important thing which is connected to the above defined unstrict satisfaction of the rule conditions is a necessity of preservation of the consistency of the model of the system (as consistency we understand here the exclusion of the possibility of existing complementary facts, for example, $f_1, \neg f_1$). The second clause of the above definition ($\sim \neg f_x$) controlling condition may be satisfied in the unstrict way only if it is not known that it is false and it is impossible to derive that it is false. This clause determines that unstrict satisfaction may be defeated by strict declaration of another fact or by another rule.

Definition 8 (satisfaction of the conditions of the rule). If there is a given case C described by a set of literals $C = \{l_x, l_y, \dots, l_z\}$ which satisfies in a strict or unstrict way conditions (Conditions) of a rule $rf \in RF$, then we denote it as $C \bullet$ Conditions.

3.3. Inference Rule. The above system has one important feature: there is a distinction between truthfulness of the condition and its satisfaction. In order to create an inference mechanism for our model, we have to modify the classic *Modus Ponens* rule.

Definition 9 (inference rule). As an inference rule we understand the following rule:

$$\frac{(\text{Conditions} \rightarrow \text{Conclusions}) \wedge f \bullet \text{Conditions}}{\text{Conclusions}}, \quad (7)$$

where \rightarrow is a strict implication and $f \bullet$ Conditions mean that fact f satisfies (in a strict or unstrict way) conditions of a given rule.

Strict or unstrict satisfaction of the rule antecedents allows us to treat rule conclusion as true and, consequently, also satisfied.

3.4. Inference Mechanism. On the basis of the above defined inference rule, we have to define our inference mechanism.

Definition 10 (fact based inference mechanism). As a fact based inference mechanism we understand forward chaining mechanism using the inference rule defined earlier. As C' we denote a set of conclusions whose inference mechanism concludes from a case C , a set of rules RF and a set of orders OF . We may also denote it as $C \vdash C'$. The union of sets $C \cup C'$ we call a complete description of the case and denote as P .

3.5. Security Attributes. As described above, the security attributes' set SA consists of an unlimited but finite number of security attributes. Each of them has its own evaluation value expressed by a positive integer number.

Definition 11 (set of security attributes pairs.). S is a set of pairs $O = \langle sa, o \rangle$, where $sa \in SA$ is a security attribute and o is its evaluation value.

For example, we have three security attributes with their evaluation values:

$$SA = \text{confidentiality, integrity, authorisation};$$

$$S = \{(\text{confidentiality}, 10), (\text{integrity}, 20), (\text{author.}, 30)\}. \quad (8)$$

It is important to notice that security attributes may have positive or negative character, in the sense that a higher value of security attribute evaluation may mean better (for positive) or worse (for negative) evaluation.

3.6. Conflicts between Rules. Some specific conditions may cause conflicts between evaluation rules.

Definition 12 (conflicting rules). There is a conflict between two or more evaluation rules if these rules cannot be executed together.

Such conflicts may appear when there are two rules whose antecedents are satisfied, who are in a way connected and the execution of both of them may cause improper influence on the security attribute evaluation.

The problem of conflicting and subsuming rules is the main reason for the utilisation of defeasible implication. In this work as defeasibility of the evaluation rules we understand the possibility of exclusion from the evaluation process of a chosen rule by another rule. If antecedents of two conflicting rules are satisfied, only one of them may be executed (but such a rule may be also defeated by another one).

To represent priorities between the evaluation rules we assume partial order between rules from a set ER . Such an order allows us to express that if $r_1 > r_2$ and $r_1, r_2 \in ER$, then rules r_1 and r_2 are in conflict and when the conditions of both of these rules are satisfied rule r_1 should defeat rule r_2 .

3.6.1. Reasoning about Orders between Conflicting Rules. The main problem of the above mentioned issue of conflicting

rules lies in the mechanism of recognition of conflicting rules. Generally, such recognition should be based on common-sense reasons, but there is one phenomenon, which allows us to recognize conflict and to find order between conflicting rules. This situation concerns a case where there are two rules, one of which has a more general set of conditions than the other one. In other words, every case satisfying condition of the rule r_2 also satisfies conditions of the rule r_1 . Such subsumption of the rules allows us to conclude that the rule r_2 is a specific case of the rule r_1 , and in the case of satisfaction of the antecedents of both rules, the rule r_2 should defeat the rule r_1 . More generally we may say that two rules are in conflict when the list of conditions of one of them subsumes that of the second one and the conclusions of both concern the modification of the same security attribute evaluation value.

Definition 13 (subsuming rules). When we have the following two rules:

$$\begin{aligned} r_x &: \text{Condition}_x \implies \text{Inf}^{V1}(\text{sa}), \\ r_y &: \text{Condition}_y \implies \text{Inf}^{V2}(\text{sa}), \end{aligned} \quad (9)$$

where Condition_x and Condition_y are the lists of antecedents of these rules, both rules influence the same security attribute evaluation value sa (but they may have a different level of influence), and if for any case P represented by a set of literals:

$$\forall_{P \in L} ((P \bullet \text{Condition}_x) \longrightarrow (P \bullet \text{Condition}_y)), \quad (10)$$

then we recognise the rules r_x and r_y as subsuming and conflicting ones and in the view of a more restrictive character of the rule r_x we may conclude that the rule r_x has priority over the rule r_y , which we denote: $r_x > r_y$ and while conditions of both rules are satisfied, the rule r_x should defeat the rule r_y .

Rules with the function $\text{Inf}^0(\text{sa})$ on the consequent part of the rule have the highest possible priority and they are in conflict with all the rules concerning the security attribute sa . Everytime when they satisfy conditions, they exclude all evaluation rules concerning the security attribute sa from reasoning.

Another aspect which is important and requires explanation is connected with the reason why a more specific rule defeats a more general one. Such a mechanism comes from the theory of law and is called *lex specialis derogat legi generali*. It is one of the tools which allow to find a solution in conflicting legal rules, saying that a specific act (provision) derogates from (prevails over) the general regulation. In modelling legal rules there are many problems connected with the difficulties with recognition which of the rules are more or less specific [24]. In the case of our model it is much simpler because the hierarchy of generality of antecedents of the rules is easy to establish based on a finite number of possible facts and their explicitness.

3.7. Evaluation Rules System

Definition 14 (evaluation rules system). Evaluation rules system RO is a structure described by a set of evaluation rules ER and relation $\text{OR} = \langle \text{ER}, > \rangle$.

The relation OR represents partial order between rules from a set ER. This order maps preferences between conflicting rules. These preferences can come from strict declaration or from a previously defined mechanism of finding and resolving a problem of subsuming rules. If there is a relation of partial order between two rules, we treat them as conflicting ones. If these rules are not comparable, we treat them as conflict free. We also assume that the relation of order between rules is transitive:

$$\forall_{r_x, r_y, r_z} ((r_x > r_y) \wedge (r_y > r_z) \longrightarrow (r_x > r_z)). \quad (11)$$

3.8. QoP Evaluation Process of Security Mechanisms. The process of QoP evaluation of security mechanisms is expressed by means of evaluation of security attributes. The values of the SA depend on the used security mechanisms which are represented in the model by facts F .

The evaluation process of security mechanisms may be described in terms of a sequence of steps as presented by Algorithm 1. The parameters and variables used in this algorithm are presented in Table 3.

3.9. Background of the Model. Looking more generally at our model, one can notice that it contains some elements taken from the formal models of legal reasoning. Legal reasoning has a very specific character, as it requires mechanisms of dealing with incomplete knowledge, mechanisms of resolving conflicts between legal rules and arguments, various ways of interpretation of rules, and so forth. Computer systems which aims to support human reasoning very often, have to face similar problems, which is the main reason for the utilisation of the *a'fortiori* rule or the *lex specialis*... rule in our system. Similar models of legal reasoning have also been applied in other computer science utilisations; for example, in [25] the authors are forced to implement one of the methods of resolving the conflicts between rules in multiagent systems.

Our model is based on proposition logic but there are some additions which allow for a better representation of specific features of the analysed problem. First of all, we introduce a distinction between two kinds of negation: *classical negation* (strong) and *negation as failure*. As negation as failure we understand the negation used in the conditional part of the rule. Such a negated condition is fulfilled if it is impossible to satisfy such a condition (it is false, it is not declared, or it is impossible to derive that it is satisfied). These two kinds of negation are used in a few logical systems, for example, in the Prakken and Sartor logic [26] or the Kowalski and Toni logic [27]. In our model the way of the utilisation of negation as failure is similar to the one presented by Prakken and Sartor in [26], but we do not allow to use construction like $\sim \neg P$, because in our model it is forbidden to use negation as failure in the consequence part of the rule and it is also

```

(1) SET C
(2) for i = 1 to n do
(3)  oi ← 0
(4)  SET OF[i]
(5)  SET C'[i] = RES(C, OF[i], R)
(6)  P[i] = C[i] ∪ C'[i]
(7)  SET ER[i]
(8)
(9)  if ER[i] = ∅ then
(10)   oi ← 0
(11)   CONTINUE
(12)   for k = 1 to NER[i] do
(13)     for m = 1 to NER[i] do
(14)       if (er[k][i], er[m][i] ∧ er[k][i] > er[m][i]) then
(15)         EXCLUDE er[m][i] from ER[i]
(16)       end if
(17)     end for
(18)   end for
(19) end if
(20)
(21) for l = 1 from NER[i] do
(22)   if exists er[l][i] in ER[i] such that conclusion is Inf0(i)
(23)     then
(24)       oi ← 0
(25)       CONTINUE
(26)     else
(27)       READ V[l][i]
(28)       oi = oi + V[l][i]
(29)     end if
(30) end for

```

ALGORITHM 1: Algorithm of security attributes evaluation.

TABLE 3: The parameters and variables for the security attributes evaluation algorithm.

SET	Make a choice indication
EXCLUDE	Excluding from the ER indication
READ	Reading indication
CONTINUE	Processing statement will be skipped
RES(C, OF[i], R)	The reasoning function based on a set of facts C and order of facts OF[i] for the security attribute i and rules R (inference mechanisms)
OF[i]	Orders between facts referred to a security attribute i
C	A case expressed by a set of facts
C'	A set of facts obtained from the inference mechanism
P[i]	A full description of a case for a security attribute i
R	A set of rules
k, m, l	Indicates the current evaluation rule
o _i	The evaluation of i th security attribute
ER[i]	A set of evaluation rules with satisfied conditions for the security attribute i
NER[i]	The number of rules with satisfied conditions for the security attribute i
er[x][i]	The evaluation rule x for the security attribute i
i	The index of the current security attribute
n	The quantity of security attributes
V[l][i]	The value of the security influence of the security mechanisms represented by the evaluation rule l for the security attribute i

forbidden to use classical negation in the antecedent part of the rule.

Another important point in our model, the conception of orders between facts, is based on a simplified version of the a fortiori reasoning (reasoning from more to less: if norm N1 obliging to do more is binding, then norm N2 obliging to do less is binding more). In our work, this way of reasoning has been slightly modified: when condition X of a rule r_1 is not satisfied literally, but there is a fact Y which satisfies this condition in a better way we may treat such a condition as satisfied. A more profound analysis and model of the a fortiori reasoning can be found in [23].

The utilisation of defeasible implication in evaluation rules is another important feature of our model. The idea of distinction of two kinds of implication comes from the formal models of legal argumentation in which the problem of defeasibility of the rules is broadly discussed. In the Prakken and Sartor logic [26], all arguments are defeasible. Vreeswijk in his abstract argumentation system [28] uses two kinds of implication (material \supset and defeasible \triangleright). He assumes that the utilisation of defeasible implication requires the definition of a separate defeasible inference rule. Hage in his reason based logic [29] looks at the problem of defeasibility of the rules from another point of view, stating that this is a problem of the applicability of the rule. The rule could have satisfied conditions, but cannot be not applicable due to, for example, a conflict with another rule. Another interesting model of argumentation which uses defeasible and strict implication is introduced by Kowalski and Toni in [27]. In their system, each defeasible rule r has a condition \sim defeated(r) (where \sim is negation as failure and defeated(r) is a predicate which denotes that this rule is defeated by another one) which allows defeating it if it remains in conflict with another rule r' and has lower priority than r' .

Defeasible rules in our model are slightly different from those in the above mentioned models. The most important point in our model lies in the fact that only evaluation rules (which are not rules in a strict meaning of this word) are defeasible. Such a rule can be defeated only if its conditions are satisfied; it is in conflict with another rule with satisfied conditions and has a lower priority over the other one. Such a defeated rule is excluded from the reasoning process.

Torre and Tan in [30] define a few types of defeasibility and the one used in our model is closest to the *overridden defeasibility* which formalizes the cancellation of a rule by another one.

The ways of dealing with conflicts between rules and orders between these rules are discussed in the aforementioned works by Prakken and Sartor (i.e., [26, 31]) where the authors introduce their formal model of legal argumentation. The notion of conflict between rules in our approach is different from the one presented in their works, but the way of resolving it by the declaration of orders between rules and the assumption of defeasibility of rules is similar to the Prakken and Sartor model. In the above mentioned Kowalski and Toni logic, two rules are in conflict when they have complementary conclusions and the conditions of both rules are satisfied.

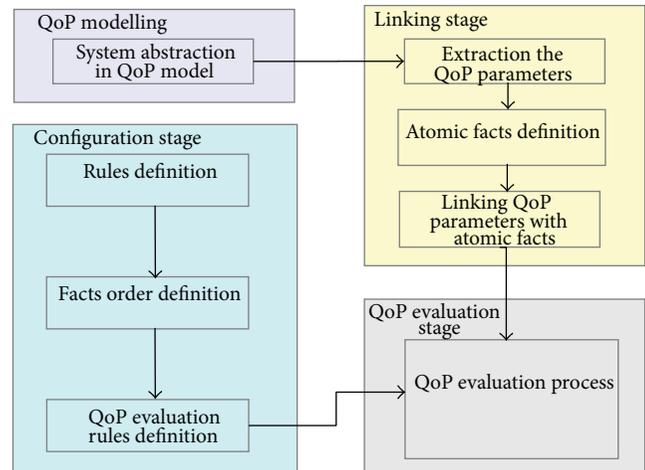


FIGURE 1: The methodology of QoP evaluation of security mechanisms.

Their model also uses two kinds of implication as well as two kinds of negation, but the way of dealing with conflict is slightly different from the one in our system. They define a few new predicates which are used to denote that the condition of the rule r is satisfied (holds(r)), rule r is defeated (defeated(r)) or two rules are in conflict (conflict(r, r')).

4. Methodology of QoP Evaluation of Security Mechanisms

The QoP evaluation of security mechanisms is a process which can be divided into four stages: QoP modelling, linking, configuration, and QoP evaluation. In Figure 1 the methodology of the process is presented.

QoP Modelling Stage. The first stage refers to modelling the system in one of the QoP models [4, 7–9, 12–17]. This phase depends on the chosen QoP approach, but the result should be the same, and the system must be modelled with QoP meaning (*system abstraction in the QoP model*). The proposed model can be used for different QoP modelling approaches.

Linking Stage. The next stage is responsible for the linking system modelled in the QoP model with the structure defined in the proposed model. Firstly, one has to extract all parameters used in the QoP model which refer to the factors which influence QoP parameters (*extraction of the QoP parameters*). After this, one creates atomic facts in the proposed model which are explicit to these extracted from the model (*atomic facts definition*). The parameters' names used in the QoP model can be different from those defined in the model, so one has to link them together (*linking QoP parameters with atomic facts*).

Configuration Stage. The next stage is the main phase where all structures proposed in the model for QoP evaluation of security mechanisms are defined. One can enumerate: *rules definition*, *facts order definition*, and *QoP evaluation rules*

definition. All of these structures are described in detail in Section 2.

QoP Evaluation Stage. The last stage is responsible for QoP evaluation of security mechanisms. The analysed specification can be defined directly in the QoP model and thanks to the previously defined links between the QoP parameter and facts, a model of a case is generated. Finally, after the system version indication, the QoP evaluation of security mechanisms is performed (*QoP evaluation process*).

5. Case Study: TLS Handshake Protocol

In this section we are going to present a case study of the QoP evaluation of security mechanisms for the TLS Handshake protocol. The TLS protocol is used each day in real business situations in the actual enterprise environment. Given the enterprise network infrastructure in Figure 2, one should analyse different roles which refer to different levels of the quality of protection of used security mechanisms. The users are allowed to access e-mail, FTP, web, and application servers with the communication channel protected by means of the TLS protocol at a different QoP level. The utilized versions of the TLS protocol together with equivalent cryptographic algorithms are summarized in Table 7.

The model for the TLS Handshake protocol can be found in the models library in the SMETool, which can be downloaded from the web page of the Quality of Protection Modelling Language Project [11]. In this case study, the client wants to verify the server and, next, to connect with the server by a secure connection. All of these security requirements can be realized by means of different security measures. We are analysing six versions of the protocol. The flow of the TLS Handshake protocol, which will be analysed further, is realized in five steps and the scheme is presented as shown in the following steps:

- (1) $C \rightarrow S$: *ClientHello*
- (2) $S \rightarrow C$: *ServerHello*, *Certificate*, *ServerKeyExchange*, *ServerHelloDone*
- (3) $C \rightarrow S$: *ClientKeyExchange*, *ChangeCipherSpec*, *Finished*
- (4) $S \rightarrow C$: *ChangeCipherSpec*, *Finished*
- (5) $C \rightarrow S$: *Encrypted data*.

Below we present a description of these steps.

- (1) A Client sends the *ClientHello* message. This message contains the following attributes: the TLS Protocol version, session id, a list of available cipher suite, compression method, and random values.
- (2) The server responds with a *ServerHello*, which establishes the version of the TLS Protocol, session id (when session is resumed), cipher suite, and compression method. It also sends random values within *ServerHello*. Next the server sends the *Certificate* message which includes its certificate. Sending this message is not necessary; it depends on the selected

cipher suite. The next message which may be sent is *ServerKeyExchange*. The server sends it when the server certificate is only for signing, or the server has no certificate. After this, the server sends *ServerHelloDone*, signalling that this phase is completed.

- (3) Next, the client sends the *ClientKeyExchange* message. Depending on the selected cipher, this message may have different contents. After this, the *ChangeCipherSpec* message is sent. The client sends it to signal the server that it has started to use the encryption. Finally, the encrypted *Finished* message is sent by the client.
- (4) Similarly, in response, the server sends *ChangeCipherSpec* and the encrypted *Finished* message.
- (5) The handshake is complete. Now, the server and client can exchange *Encrypted data*.

The methodology of the QoP evaluation of security mechanisms based on our model is presented in Section 3. In this section we have used the proposed methodology for analysing the TLS Handshake cryptographic protocol.

5.1. QoP Modelling. In the first step one has to model the system in one of the QoP models. In the presented case study, the TLS protocol is analysed as a full system. The QoP modelling process is a complex task so we are not going to present it in this paper. The example of the QoP model of TLS cryptographic protocol is presented in [6], where the protocol is modelled in the QoP-ML modelling language [12].

5.2. Linking Stage

5.2.1. Extraction of the QoP Parameters. The first step in the linking stage refers to the extraction of all parameters used in the QoP model which refer to the factors which influence QoP. The form of the QoP parameters depends on the chosen QoP model. For example, in the QoP-ML modelling language, the system behaviour is changed by the functions which modify the states of the variables and pass the objects by communication channels. This structure is responsible for indicating the parameters which influence the system QoP.

Let us assume that qp is the QoP parameter in the QoP model which influences system security. The QoP parameter is atomic, which means that it cannot be split into other atomic QoP parameters. The QP is a set of the QoP parameters:

$$QP = \{qp_1, qp_2, qp_3, \dots, qp_n\}, \quad (12)$$

where

$qp_1, qp_2, qp_3, \dots, qp_n$ are the QoP parameters which influence system security;

QP is a set of the QoP parameters.

5.2.2. Atomic Facts Definition. In the next step, we declare a set of all possible facts F which will represent the features of the analysed systems.

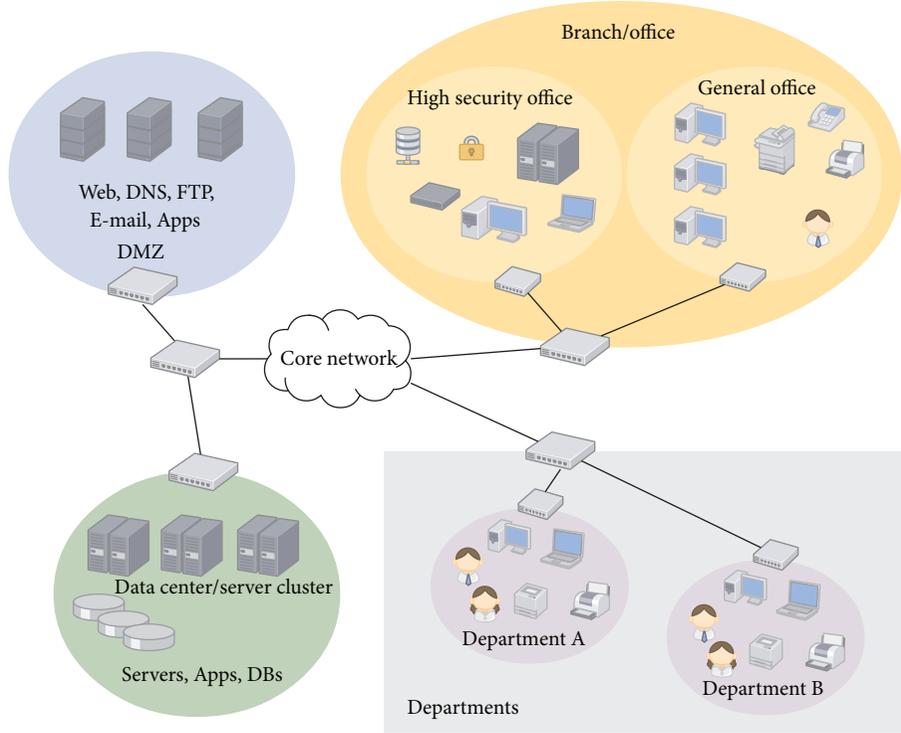


FIGURE 2: Example enterprise network architecture.

TABLE 4: The group of facts refers to symmetric encryption.

Facts
Group name: cipher (symmetric encryption algorithm)
$f_1(\text{cipher}) = \text{RC4}$
$f_2(\text{cipher}) = \text{3DES}$
$f_3(\text{cipher}) = \text{AES}$
Group name: bs (block size in bytes)
$f_1(\text{bs}) = 8$
$f_1(\text{bs}) = 16$
Group name: IV (initiate vector in bytes)
$f_1(\text{IV}) = 8$
$f_1(\text{IV}) = 16$
Group name: key (key length in bytes)
$f_1(\text{key}) = 16$
$f_2(\text{key}) = 24$
$f_3(\text{key}) = 32$

In the presented case study we include the TLS cryptographic protocol. In the following section we define the atomic facts in the proposed model for the TLS protocol. These facts are taken from the official specification of the TLS protocol [32] where all possible versions of the protocol are defined. The facts are divided into three groups which refer to different factors: symmetric encryption (Table 4), message digest (Table 5), asymmetric encryption, and common facts (Table 6).

TABLE 5: The group of facts refers to message digest.

Facts
Group name: mac (message authentication code algorithm)
$f_1(\text{mac}) = \text{HMAC-MD5}$
$f_2(\text{mac}) = \text{HMAC-SHA1}$
$f_3(\text{mac}) = \text{HMAC-SHA256}$
Group name: mac-len (message digest length in bytes)
$f_1(\text{mac-len}) = 16$
$f_2(\text{mac-len}) = 20$
$f_3(\text{mac-len}) = 32$
Group name: k-len (mac key length in bytes)
$f_1(\text{k-len}) = 16$
$f_2(\text{k-len}) = 20$
$f_3(\text{k-len}) = 32$

5.2.3. *Linking QoP Parameters with Atomic Facts.* The names of the QoP parameters used in the QoP model can be different from atomic facts defined in the proposed model. In this step, the QoP parameters are linked with the facts in the proposed model in an explicit way.

Definition 15 (linking operator). The linking operator \mapsto denotes that one set of objects is mapped to another set of objects in an explicit way.

As we assume that the QP is the set of the QoP parameters used in the QoP model for the TLS protocol abstraction

TABLE 6: The group of facts refers to asymmetric cryptography and common facts.

Facts
Group name: PK (key exchange algorithm scheme)
$f_1(\text{PK}) = \text{RSA}$
$f_2(\text{PK}) = \text{DH-DSS}$
$f_3(\text{PK}) = \text{DH-RSA}$
$f_4(\text{PK}) = \text{DHE-DSS}$
$f_5(\text{PK}) = \text{DHE-RSA}$
$f_6(\text{PK}) = \text{DH-anon}$
Group name: mode (mode of operation)
$f_1(\text{mode}) = \text{CBC}$
Group name: com (bit compression before encryption)
$f_1(\text{com}) = \text{Compression}$

TABLE 7: The analysed versions of TLS protocol.

Version	The cipher suite
1	TLS_RSA_WITH_RC4_128_MD5
2	TLS_RSA_WITH_AES_128_CBC_SHA
3	TLS_DH_anon_WITH_RC4_128_MD5
4	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
5	TLS_RSA_WITH_RC4_128_newSHA512
6	TLS_RSA_WITH_RC4_128_MD5 + COM

and F is the set of all atomic facts defined in the proposed model. Usually, the set of the QoP parameters QP is a subset of the facts in the model F , but in a special case this set can be equal to

$$\begin{aligned} QP &= \{qp_1, qp_2, qp_3, \dots, qp_n\}; \\ F &= \{f_1, f_2, f_3, \dots, f_n\}; \\ Z &\in F, \end{aligned} \quad (13)$$

where

- $qp_1, qp_2, qp_3, \dots, qp_n$ are QoP parameters which influence the system security;
- QP is the set of QoP parameters;
- $f_1, f_2, f_3, \dots, f_n$ are facts in the formal model;
- F is the set of all facts in the model;
- Z is the subset of all facts in the model.

QoP parameters from the QoP model are linked with the subset of the facts defined in the model in an explicit way by the linking operator \mapsto :

$$QP \mapsto Z. \quad (14)$$

5.3. Configuration Stage. The next stage is the main phase where all structures proposed in the model for the QoP evaluation of security mechanisms are defined. Among them, one can enumerate definitions of rules, facts order, and the QoP evaluation rules.

5.3.1. Rules Definition. Based on the TLS cryptographic protocol specification [32], we specify the rules which refer to a possible realization of the protocol. In Appendix A one can find the rules defined for the TLS protocol.

5.3.2. Facts Order Definition. In the next step, the facts order must be defined. The order between facts can be defined only for the same security attribute (SA). For the presented example, we evaluate the QoP of security mechanisms in the case of four security attributes: integrity (I), confidentiality (C), authentication (Au), and availability (A). We define the facts order according to the expert knowledge in the field of cryptographic protocols. All the defined fact orders are presented in Appendix B.

5.3.3. QoP Evaluation Rules Definition. The last step in the configuration stage is to define the QoP evaluation rules. For the TLS cryptographic protocol, the evaluation rules refer to the same four security attributes: integrity (I), confidentiality (C), authentication (Au), and availability (A). According to these rules, the QoP evaluation of security mechanism of the TLS protocol is performed. In the literature, defining the influence of specific security mechanisms is based on expert knowledge in the field of cryptology and system security [7, 8]. In our example, we perform the same expert knowledge analysis which refers to the TLS cryptographic protocol. The defined QoP evaluation rules are presented in Appendix C.

5.4. QoP Evaluation Stage. Having finished the configuration stage, one can start the QoP evaluation process. In the presented example we choose six versions of the TLS protocol which are presented in Table 7. These cipher suites are described in detail in the TLS specification [32]. The fifth and sixth versions are modified according to the versions presented in the TLS specification. In the fifth version we analyse a case where a new cryptographic module, compared to the previously defined ones, will be possible. This module is the implementation of HMAC-SHA512 [33]. The sixth version is identical with the first one except that compression is enabled.

The first version is the most popular cipher suite for online banking. Some banks operate their online services using the second TLS protocol version. The third version can be used when authorisation is not required. The fourth version accomplishes the TLS protocol with the strongest set of security parameters. The fifth version analyses the hypothetical scenario when a new implementation of one of the cryptographic modules is possible. The sixth version is identical with the first one, but the data are compressed before encryption.

These six versions are analysed as 6 cases which represent 6 different realizations of the TLS protocol. These versions can be represented as the following set of facts.

Case 1. $C_1 = \{f_1(\text{PK}), f_1(\text{cipher}), f_1(\text{mac}), f_1(\text{key})\}.$

Case 2. $C_2 = \{f_3(\text{cipher}), f_1(\text{key}), f_1(\text{mode}), f_2(\text{mac}), f_1(\text{PK})\}.$

Case 3. $C_3 = \{f_1(\text{cipher}), f_6(\text{PK}), f_1(\text{key}), f_1(\text{mac})\}$.

Case 4. $C_4 = \{f_5(\text{PK}), f_3(\text{cipher}), f_3(\text{key}), f_1(\text{mode}), f_3(\text{mac})\}$.

Case 5. $C_5 = \{f_1(\text{PK}), f_1(\text{cipher}), f_4(\text{mac}), f_1(\text{key})\}$.

Case 6. $C_6 = \{f_1(\text{PK}), f_1(\text{cipher}), f_1(\text{mac}), f_1(\text{key}), f_1(\text{com})\}$.

In Case 5, one can find the fact $f_4(\text{mac})$, which is not defined in Table 5. The example is a situation in which a new protocol implementation will be realized and the QoP evaluation system does not provide analysis for this version. In such a case, this fact will be added later during the detailed analysis of this case.

After defining the facts which describe the analysed cases (C), one has to derive (\vdash) other atomic or complex facts (C'): $C \vdash C'$. They can be derived by using the earlier described inference mechanism and the rules defined in Appendix A. On the basis of this knowledge, one can prepare final evaluation of security mechanisms represented as the security attributes. Below we present the evaluation of the six analysed cases.

QoP Evaluation of Case 1.

Consider the following:

$$C_1 \vdash C'_1;$$

$$\begin{aligned} C'_1 = \{ & f_1(\text{mac-len}), f_1(k\text{-len}), \neg f_1(IV), \neg f_2(IV), \\ & \neg f_1(\text{bs}), \neg f_2(\text{bs}), f(\text{mac-len}), f(k\text{-len}), \\ & f(\text{PK}), f(\text{cipher}), f(\text{mac}), f(\text{key}), \neg f_2(\text{PK}), \\ & \neg f_3(\text{PK}), \neg f_4(\text{PK}), \neg f_5(\text{PK}), \neg f_6(\text{PK}), \\ & \neg f_2(\text{cipher}), \neg f_3(\text{cipher}), \neg f_2(\text{mac}), \neg f_3(\text{mac}), \\ & \neg f_2(\text{key}), \neg f_3(\text{key}), \neg f_4(\text{key}), \neg f_2(\text{mac-len}), \\ & \neg f_3(\text{mac-len}), \neg f_2(k\text{-len}), \neg f_3(k\text{-len})\}. \end{aligned} \quad (15)$$

Case 1 is denoted as P_1 and is a union of sets C_1 and C'_1 :

$$P_1 = (C_1 \cup C'_1). \quad (16)$$

The QoP evaluation of the security attributes

$$\begin{aligned} O_1 = \{ & \langle \text{confidentiality}, 2 \rangle, \langle \text{integrity}, 3 \rangle, \\ & \langle \text{availability}, 5 \rangle, \langle \text{authorisation}, 1 \rangle\}. \end{aligned} \quad (17)$$

QoP Evaluation of Case 2.

Consider the following:

$$C_2 \vdash C'_2;$$

$$\begin{aligned} C'_2 = \{ & f_2(\text{mac-len}), f_2(k\text{-len}), f_2(IV), \\ & f_2(\text{bs}), f_2(\text{CS}), f(\text{cipher}), f(\text{key}), f(\text{mac}), \\ & f(\text{PK}), f(\text{mac-len}), f(k\text{-len}), f(IV), \\ & f(\text{bs}), \neg f_2(\text{PK}), \neg f_3(\text{PK}), \neg f_4(\text{PK}), \\ & \neg f_5(\text{PK}), \neg f_6(\text{PK}), \neg f_2(\text{cipher}), \neg f_1(\text{cipher}), \\ & \neg f_1(\text{mac}), \neg f_3(\text{mac}), \neg f_2(\text{key}), \neg f_3(\text{key}), \\ & \neg f_4(\text{key}), \neg f_1(\text{mac-len}), \neg f_3(\text{mac-len}), \\ & \neg f_1(k\text{-len}), \neg f_3(k\text{-len}), \neg f_1(\text{bs}), \neg f_1(IV)\}. \end{aligned} \quad (18)$$

Case 2 is denoted as P_2 and is a union of sets C_2 and C'_2 :

$$P_2 = (C_2 \cup C'_2). \quad (19)$$

The QoP evaluation of the security attributes

$$\begin{aligned} O_2 = \{ & \langle \text{confidentiality}, 8 \rangle, \langle \text{integrity}, 6 \rangle, \\ & \langle \text{availability}, 10 \rangle, \langle \text{authorisation}, 1 \rangle\}. \end{aligned} \quad (20)$$

QoP Evaluation of Case 3.

Consider the following:

$$C_3 \vdash C'_3;$$

$$\begin{aligned} C'_3 = \{ & f_1(\text{mac-len}), f_1(k\text{-len}), \neg f_1(IV), \\ & \neg f_2(IV), \neg f_1(\text{bs}), \neg f_2(\text{bs}), f(\text{mac-len}), \\ & f(k\text{-len}), f(\text{PK}), f(\text{cipher}), f(\text{mac}), f(\text{key}), \\ & \neg f_2(\text{PK}), \neg f_3(\text{PK}), \neg f_4(\text{PK}), \neg f_5(\text{PK}), \\ & \neg f_1(\text{PK}), \neg f_2(\text{cipher}), \neg f_3(\text{cipher}), \neg f_2(\text{mac}), \\ & \neg f_3(\text{mac}), \neg f_2(\text{key}), \neg f_3(\text{key}), \neg f_4(\text{key}), \\ & \neg f_2(\text{mac-len}), \neg f_3(\text{mac-len}), \\ & \neg f_2(k\text{-len}), \neg f_3(k\text{-len})\}. \end{aligned} \quad (21)$$

Case 3 is denoted as P_3 and is a union of sets C_3 and C'_3 :

$$P_3 = (C_3 \cup C'_3). \quad (22)$$

The QoP evaluation of the security attributes

$$\begin{aligned} O_3 = \{ & \langle \text{confidentiality}, 2 \rangle, \langle \text{integrity}, 3 \rangle, \\ & \langle \text{availability}, 5 \rangle, \langle \text{authorisation}, 0 \rangle\}. \end{aligned} \quad (23)$$

QoP Evaluation of Case 4.

Consider the following:

$$C_4 \vdash C'_4;$$

$$\begin{aligned} C'_4 = \{ & f_3(\text{mac-len}), f_3(k\text{-len}), f_2(IV), f_2(\text{bs}), \\ & f_2(\text{CS}), f(PK), f(\text{cipher}), f(\text{key}), f(\text{mode}), \\ & f(\text{mac}), f(\text{mac-len}), f(k\text{-len}), f(IV), \\ & f(\text{bs}), \neg f_2(PK), \neg f_3(PK), \neg f_4(PK), \neg f_6(PK), \\ & \neg f_1(PK), \neg f_1(\text{cipher}), \neg f_2(\text{cipher}), \neg f_1(\text{mac}), \\ & \neg f_2(\text{mac}), \neg f_1(\text{mac-len}), \neg f_2(\text{mac-len}), \\ & \neg f_1(k\text{-len}), \neg f_2(k\text{-len}), \neg f_1(IV), \neg f_1(\text{bs}) \}. \end{aligned} \quad (24)$$

Case 4 is denoted as P_4 and is a union of sets C_4 and C'_4 :

$$P_4 = (C_4 \cup C'_4). \quad (25)$$

The QoP evaluation of the security attributes

$$O_4 = \{ \langle \text{confidentiality}, 10 \rangle, \langle \text{integrity}, 9 \rangle, \langle \text{availability}, 15 \rangle, \langle \text{authorisation}, 3 \rangle \}. \quad (26)$$

QoP Evaluation of Case 5. Case 5 is different from the previous four cases because there is a fact declared ($f_4(\text{mac})$) which is not in the conditional part of any rule. As we have stated before, a declaration of order between facts is much easier than adding new rules. Based on that, we declare two new orders:

$$\begin{aligned} f_4(\text{mac}) >_I f_3(\text{mac}) >_I f_2(\text{mac}) >_I f_1(\text{mac}); \\ f_1(\text{mac}) >_A f_2(\text{mac}) >_A f_3(\text{mac}) >_A f_4(\text{mac}). \end{aligned} \quad (27)$$

These orders mean that *HMAC – SHA512* is more than *HMAC – SHA256* in the context of integrity, but it is also less than *HMAC – SHA256* in the context of availability.

Adding these orders to the knowledge base results in satisfying conditions of all rules which have $f_3(\text{mac})$ in their conditional parts:

$$C_5 \vdash C'_5;$$

$$\begin{aligned} C'_5 = \{ & f_3(\text{mac-len}), f_3(k\text{-len}), \neg f_1(IV), \\ & \neg f_2(IV), \neg f_1(\text{bs}), \neg f_2(\text{bs}), f(\text{mac-len}), \\ & f(k\text{-len}), f(PK), f(\text{cipher}), f(\text{mac}), f(\text{key}), \\ & \neg f_2(PK), \neg f_3(PK), \neg f_4(PK), \neg f_1(\text{mac}), \\ & \neg f_5(PK), \neg f_6(PK), \neg f_2(\text{cipher}), \neg f_3(\text{cipher}), \\ & \neg f_2(\text{mac}), f_3(\text{mac}), \neg f_2(\text{key}), \neg f_3(\text{key}), \\ & \neg f_4(\text{key}), \neg f_2(\text{mac-len}), \neg f_1(\text{mac-len}), \\ & \neg f_2(k\text{-len}), \neg f_1(k\text{-len}) \}. \end{aligned} \quad (28)$$

Case 5 is denoted as P_5 and is a union of sets C_5 and C'_5 :

$$P_5 = (C_5 \cup C'_5). \quad (29)$$

The QoP evaluation of the security attributes

$$O_5 = \{ \langle \text{confidentiality}, 2 \rangle, \langle \text{integrity}, 9 \rangle, \langle \text{availability}, 11 \rangle, \langle \text{authorisation}, 1 \rangle \}. \quad (30)$$

QoP Evaluation of Case 6. Case 6 is the same as Case 1 with the difference that the compression is enabled:

$$C_6 \vdash C'_6;$$

$$\begin{aligned} C'_6 = \{ & f_1(\text{mac-len}), f_1(k\text{-len}), \neg f_1(IV), \\ & \neg f_2(IV), \neg f_1(\text{bs}), \neg f_2(\text{bs}), f(\text{mac-len}), \\ & f(k\text{-len}), f(PK), f(\text{cipher}), f(\text{mac}), \\ & f(\text{key}), \neg f_2(PK), \neg f_3(PK), \neg f_4(PK), \\ & \neg f_5(PK), \neg f_6(PK), \neg f_2(\text{cipher}), \neg f_3(\text{cipher}), \\ & \neg f_2(\text{mac}), \neg f_3(\text{mac}), \neg f_2(\text{key}), \neg f_3(\text{key}), \\ & \neg f_4(\text{key}), \neg f_2(\text{mac-len}), \neg f_3(\text{mac-len}), \\ & \neg f_2(k\text{-len}), \neg f_3(k\text{-len}) \}. \end{aligned} \quad (31)$$

Case 6 is denoted as P_6 and is a union of sets C_6 and C'_6 :

$$P_6 = (C_6 \cup C'_6). \quad (32)$$

This case presents the situation in which we have two conflicting evaluation rules:

$$\begin{aligned} f_1(\text{cipher}) &\implies \text{Inf}^1(A) \quad \text{which we denote as } er_1, \\ f_1(\text{COM}) \wedge f_1(\text{cipher}) &\implies \text{Inf}^4(A), \quad (33) \\ &\text{which we denote as } er_{36}. \end{aligned}$$

The rule er_{36} is subsumed by the rule er_1 because every case which satisfies the rule er_{36} also satisfies the rule er_1 . Both of these rules evaluate the same security attribute and both of them, in our case, satisfy the conditions. Based on the previously defined mechanism of recognition of subsuming and conflicting rules, we will treat these rules as conflicting ones and the rule er_{36} will defeat the rule er_1 . The QoP evaluation of the security attributes

$$O_6 = \{ \langle \text{confidentiality}, 2 \rangle, \langle \text{integrity}, 3 \rangle, \langle \text{availability}, 8 \rangle, \langle \text{authorisation}, 1 \rangle \}. \quad (34)$$

The results obtained by the QoP evaluation of security mechanisms are presented in Table 8. These results are quantitative.

TABLE 8: The QoP evaluation of the analysed versions of TLS protocol.

Version	C	I	A	Au
1	2	3	5	1
2	8	6	10	1
3	2	3	5	0
4	10	9	15	3
5	2	9	11	1
6	2	3	8	1

5.4.1. *Qualitative Estimation.* The results are presented as quantitative estimation of security attributes. During the QoP evaluation of security mechanisms one can introduce qualitative interpretation of the results. That kind of estimation is made for the all security attributes.

In the presented example, we introduce 5 levels of evaluation: very low, low, medium, high, and very high. It is important that the existing correlations between the quantitative and qualitative results have not only a theoretical character but also a real one. A practical character of the qualitative estimation of the security attributes is obtained because the minimal and maximal possible values of the security attributes for a particular version of the analyzed protocol are calculated. The ranges of parameters for the qualitative evaluation are calculated by formula (35).

For the analysed versions of the TLS protocol, the qualitative assessment will be prepared according to the ranges presented in Table 9. These ranges are calculated according to formula (35). On the basis of the calculated ranges, the qualitative assessment of the TLS protocol is obtained. These marks are presented in Table 10:

$$\begin{aligned}
 \text{very low} &= (Q_{\min}, Q_{\min} + X) ; \\
 \text{low} &= (Q_{\min} + X, Q_{\min} + 2X) ; \\
 \text{medium} &= (Q_{\min} + 2X, Q_{\min} + 3X) ; \\
 \text{high} &= (Q_{\min} + 3X, Q_{\min} + 4X) ; \\
 \text{very high} &= (Q_{\min} + 4X, Q_{\min} + 5X) ,
 \end{aligned} \tag{35}$$

where

$$X = \frac{Q_{\max} - Q_{\min}}{5},$$

where

Q_{\max} is the maximum value for the security attribute among all analysed versions of the protocol;

Q_{\min} is the minimum value for the security attribute among all analysed versions of the protocol.

After the QoP evaluation of security mechanisms one can interpret the results. The first and third versions of the TLS protocol are the most efficient ones in the case of CPU performance. This fact is indicated by the availability attribute. This is due to the fact that the applied security

TABLE 9: The ranges for the qualitative interpretation of QoP evaluation of the analysed versions of TLS protocol.

Mark	C	I	A	Au
Very low	(2, 3.6)	(3, 4.2)	(5, 7)	(1, 1.4)
Low	(3.6, 5.2)	(4.2, 5.4)	(7, 9)	(1.4, 1.8)
Medium	(5.2, 6.8)	(5.4, 6.6)	(9, 11)	(1.8, 2.2)
High	(6.8, 8.4)	(6.6, 7.8)	(11, 13)	(2.2, 2.6)
Very high	(8.4, 10)	(7.8, 9)	(13, 15)	(2.6, 3)

TABLE 10: The qualitative interpretation of QoP evaluation of the analysed versions of TLS protocol.

Version	C	I	A	Au
1	Very low	Very low	Very low	Very low
2	High	Medium	Medium	Very low
3	Very low	Very low	Very low	No
4	Very high	Very high	Very high	Very high
5	Very low	Very high	Medium	Very low
6	Very low	Very low	Low	Very low

mechanisms are the most efficient ones. However, the analysis of confidentiality and integrity indicates that these attributes are accomplished on the lowest level of all analysed versions.

One of the main functions of the TLS protocol is server authorisation. Versions 1, 3, 5, and 6 guarantee this attribute on the lowest level, but the third version does not guarantee it at all. The fourth protocol version achieves the strongest possible security level. This fact influences the system performance which is indicated by the availability attribute.

In a certain scenario one can use the TLS protocol version which is between the strongest and poorest security levels (*medium*). Then one can use the second version. This version guarantees the confidentiality on the high level and the integrity on the medium level. The authorisation is still guaranteed, but on the very low level. The set of parameters used in the second version allows the decrease of the system performance according to the strongest fourth version to the medium level.

The fifth and sixth versions should be compared with reference to the first version because they are a modification of the first version. In the fifth version, the new cryptographic module which guarantees the integrity on the strongest possible level is introduced. Increasing the level of protection results in a decreased performance, which is indicated by the medium level of the availability attribute. In the sixth version, the compression is enabled which decreases the system performance in comparison to the first version.

5.5. *Formal Model Goals Evaluation.* At the beginning of the formal model definition we enumerated the goals of our model. In this section, we would like to present the goals achieved by our model.

(1) *Goal 1: Automatic QoP Evaluation of the Not Directly Defined Scenarios.* One of the most important features of the model is the QoP security evaluation based on the not directly

defined scenarios. It means that during the analysis, one can prepare the QoP evaluation of one of the cryptographic protocol versions which is not directly defined. In our model the inference mechanism and rules are defined and owing to that, one can automatically derive the set of facts which describes the analysed version of the protocol. Since our model is based on the analysis of basic mechanisms of security protection and relations between them, the evaluation process does not require an advance preparation of direct scenarios describing all possible configurations of analysed systems. In our model the mechanism which allows for the recognition and resolution of conflicts between evaluation rules is introduced.

(2) *Goal 2: Quality of Protection Evaluation of All Security Mechanisms.* In the proposed model one can evaluate any of the security mechanisms with regard to the quality of protection factor. The security mechanisms, modelled in one of the QoP models, are mapped to the atomic or complex facts. In our model, one can define the facts that any of the security mechanisms can be represented by. The linking stage described in the methodology illustrates the mapping process. Finally, on the basis of the QoP evaluation rules and their order one can prepare the QoP evaluation of security mechanisms.

(3) *Goal 3: Analysis Refers to All Security Attributes.* The security attribute describes system behaviour in terms of information security requirements. That kind of behaviour is changed by the security mechanisms which modify the modelled system. As we mentioned above, in our model one can represent any security mechanism and analyse its influence on any security attribute.

(4) *Goal 4: The model Can Be Used for any QoP Models.* The presented model can be applied to the systems which are modelled by any of the QoP models. This goal is achieved due to the *linking operator* which links the QoP parameters from the QoP model with the subset of facts defined in the model. These objects are mapped in an explicit way.

5.6. *Comparison of Our Model with Existing Approaches.* In the literature one can find different approaches [4, 7–9] dealing with the quality of protection evaluation of security mechanisms. In Table II we compare the model presented in this paper with the existing approaches. These approaches can be characterized by the following main attributes.

Quantitative assessment refers to the quantitative assessment of the estimated quality of protection. All of the presented approaches, except one, allow quantitative assessment of the QoP evaluation of security mechanisms. Petriu et al. [9] discuss performance analysis in terms of the used security level, but the analysis has a qualitative character.

Formal representation refers to the representation of the quality of protection evaluation of security mechanisms by mathematical formulae. Among the enumerated approaches only the one presented in this

paper has formal representation, while the others are represented by means of an analytical model, without any formal definition of the objects and rules required for formal evaluation.

Executability specifies the possibility of the implementation of an automated tool able to perform the evaluation of QoP mechanisms. The tool support is provided for three approaches: Ksiezopolski and Kotulski [4] model is supported by the SPOT tool, Petriu et al. [9] create the UMLsec tool, and the presented model is supported by the SME tool, which can be downloaded from [11].

Indirect reasoning reasoning for not directly defined scenarios. All of the presented approaches, except the new model presented in this paper, have one significant limitation. These models can evaluate only these versions which were previously directly defined and described in detail. The presented model provides the method for indirect reasoning.

Holistic is the possibility of the evaluation of all security attributes. All presented models, except one, can be used for the evaluation of all security attributes. Only the model presented by Petriu et al. [9] focuses on performance analysis and is related to availability.

Completeness is the possibility of the representation of all security mechanisms. This attribute is provided for all models.

6. Conclusions

In the paper we propose a formal model for the quality of protection evaluation of security mechanisms. The presented method has four main features. Firstly, the presented approach allows for the QoP evaluation for the system for which the scenarios of assessment of security mechanisms are not directly defined. Secondly, all security mechanisms can be analysed with regard to the QoP factor. Thirdly, analysis can be performed for all security attributes. Finally, the proposed model can be used for any QoP models.

Our method also has some interesting features which make the evaluation of a system easier. A system can be incrementally developed by adding new knowledge. To avoid conflicts between new and old rules, our model includes the mechanism of recognition and resolution of a specific kind of conflict between evaluation rules.

In the paper we have presented the methodology of our model's preparation. On the basis of this methodology, we have created the model of the TLS cryptographic protocol. We have also performed the QoP evaluation of the sixth selected version of the TLS protocol. The main analysis refers to the quantitative assessment of four security attributes: confidentiality, integrity, availability, and authorisation. Finally, we have introduced the formula for qualitative interpretation of the prepared QoP evaluation.

An additional contribution of the paper is the implementation of the security mechanisms evaluation tool (SMETool) which supports the presented method. The SMETool can be

TABLE II: The characterization of the security mechanisms evaluation models.

	Agarwal and Wang [7]	Ksiezopolski and Kotulski [4]	Luo et al. [8]	Petriu et al. [9]	Our model
Quantitative assessment	√	√	√	—	√
Formal representation	—	—	—	—	√
Executability	—	√	—	√	√
Indirect reasoning	—	—	—	—	√
Holistic	√	√	√	—	√
Completeness	√	√	√	√	√

downloaded from the web page of the Quality of Protection Modelling Language Project [11]. The analysed model for the TLS Handshake protocol can be found in the models library in the SMETool.

Appendices

A. The Rules Definition for TLS Cryptographic Protocol

The fact based rules are as follows:

$$\begin{aligned}
& f_1(\text{cipher}) \vee f_2(\text{cipher}) \vee f_3(\text{cipher}) \rightarrow f(\text{cipher}) \\
& f_1(\text{bs}) \vee f_2(\text{bs}) \rightarrow f(\text{bs}) \\
& f_1(IV) \vee f_2(IV) \rightarrow f(IV) \\
& f_1(\text{key}) \vee f_2(\text{key}) \vee f_3(\text{key}) \rightarrow f(\text{key}) \\
& f_1(\text{mac}) \vee f_2(\text{mac}) \vee f_3(\text{mac}) \rightarrow f(\text{mac}) \\
& f_1(\text{mac-len}) \vee f_2(\text{mac-len}) \vee f_3(\text{mac-len}) \rightarrow f(\text{mac-len}) \\
& f_1(k\text{-len}) \vee f_2(k\text{-len}) \vee f_3(k\text{-len}) \rightarrow f(k\text{-len}) \\
& f_1(\text{PK}) \vee f_2(\text{PK}) \vee f_3(\text{PK}) \vee f_4(\text{PK}) \vee f_5(\text{PK}) \vee f_6(\text{PK}) \rightarrow f(\text{PK}) \\
& \sim f(\text{cipher}) \rightarrow \neg f_1(\text{key}) \wedge \neg f_2(\text{key}) \wedge \neg f_3(\text{key}) \wedge \neg f_1(\text{bs}) \wedge \neg f_2(\text{bs}) \wedge \neg f_1(IV) \wedge \neg f_2(IV) \\
& \sim f(\text{mac}) \rightarrow \neg f(\text{mac-len}) \wedge \neg f(k\text{-len}) \\
& f_1(\text{mac}) \rightarrow f_1(\text{mac-len}) \wedge f_1(k\text{-len}) \\
& f_2(\text{mac}) \rightarrow f_2(\text{mac-len}) \wedge f_2(k\text{-len}) \\
& f_3(\text{mac}) \rightarrow f_3(\text{mac-len}) \wedge f_3(k\text{-len}) \\
& f_1(\text{cipher}) \rightarrow f_1(\text{key}) \wedge \neg f_1(IV) \wedge \neg f_2(IV) \wedge \neg f_1(\text{bs}) \wedge \neg f_2(\text{bs}) \\
& f_2(\text{cipher}) \rightarrow f_2(\text{key}) \wedge f_1(IV) \wedge f_1(\text{bs}) \\
& f_3(\text{cipher}) \rightarrow f_2(IV) \wedge f_2(\text{bs}).
\end{aligned}$$

The rules which define the exclusive facts are as follows:

$$\begin{aligned}
& f_1(\text{cipher}) \rightarrow \neg f_2(\text{cipher}) \wedge \neg f_3(\text{cipher}) \\
& f_2(\text{cipher}) \rightarrow \neg f_1(\text{cipher}) \wedge \neg f_3(\text{cipher}) \\
& f_3(\text{cipher}) \rightarrow \neg f_2(\text{cipher}) \wedge \neg f_1(\text{cipher}) \\
& f_1(\text{bs}) \rightarrow \neg f_2(\text{bs}) \\
& f_2(\text{bs}) \rightarrow \neg f_1(\text{bs}) \\
& f_1(IV) \rightarrow \neg f_2(IV)
\end{aligned}$$

$$\begin{aligned}
& f_2(IV) \rightarrow \neg f_1(IV) \\
& f_1(\text{key}) \rightarrow \neg f_2(\text{key}) \wedge \neg f_3(\text{key}) \wedge \neg f_4(\text{key}) \\
& f_2(\text{key}) \rightarrow \neg f_1(\text{key}) \wedge \neg f_3(\text{key}) \wedge \neg f_4(\text{key}) \\
& f_3(\text{key}) \rightarrow \neg f_2(\text{key}) \wedge \neg f_1(\text{key}) \wedge \neg f_4(\text{key}) \\
& f_4(\text{key}) \rightarrow \neg f_2(\text{key}) \wedge \neg f_3(\text{key}) \wedge \neg f_1(\text{key}) \\
& f_1(\text{mac}) \rightarrow \neg f_2(\text{mac}) \wedge \neg f_3(\text{mac}) \\
& f_2(\text{mac}) \rightarrow \neg f_1(\text{mac}) \wedge \neg f_3(\text{mac}) \\
& f_3(\text{mac}) \rightarrow \neg f_2(\text{mac}) \wedge \neg f_1(\text{mac}) \\
& f_1(\text{mac-len}) \rightarrow \neg f_2(\text{mac-len}) \wedge \neg f_3(\text{mac-len}) \\
& f_2(\text{mac-len}) \rightarrow \neg f_1(\text{mac-len}) \wedge \neg f_3(\text{mac-len}) \\
& f_3(\text{mac-len}) \rightarrow \neg f_2(\text{mac-len}) \wedge \neg f_1(\text{mac-len}) \\
& f_1(k\text{-len}) \rightarrow \neg f_2(k\text{-len}) \wedge \neg f_3(k\text{-len}) \\
& f_2(k\text{-len}) \rightarrow \neg f_1(k\text{-len}) \wedge \neg f_3(k\text{-len}) \\
& f_3(k\text{-len}) \rightarrow \neg f_2(k\text{-len}) \wedge \neg f_1(k\text{-len}) \\
& f_1(\text{PK}) \rightarrow \neg f_2(\text{PK}) \wedge \neg f_3(\text{PK}) \wedge \neg f_4(\text{PK}) \wedge \neg f_5(\text{PK}) \wedge \neg f_6(\text{PK}) \\
& f_2(\text{PK}) \rightarrow \neg f_1(\text{PK}) \wedge \neg f_3(\text{PK}) \wedge \neg f_4(\text{PK}) \wedge \neg f_5(\text{PK}) \wedge \neg f_6(\text{PK}) \\
& f_3(\text{PK}) \rightarrow \neg f_2(\text{PK}) \wedge \neg f_1(\text{PK}) \wedge \neg f_4(\text{PK}) \wedge \neg f_5(\text{PK}) \wedge \neg f_6(\text{PK}) \\
& f_4(\text{PK}) \rightarrow \neg f_2(\text{PK}) \wedge \neg f_3(\text{PK}) \wedge \neg f_1(\text{PK}) \wedge \neg f_5(\text{PK}) \wedge \neg f_6(\text{PK}) \\
& f_5(\text{PK}) \rightarrow \neg f_2(\text{PK}) \wedge \neg f_3(\text{PK}) \wedge \neg f_4(\text{PK}) \wedge \neg f_1(\text{PK}) \wedge \neg f_6(\text{PK}) \\
& f_6(\text{PK}) \rightarrow \neg f_2(\text{PK}) \wedge \neg f_3(\text{PK}) \wedge \neg f_4(\text{PK}) \wedge \neg f_5(\text{PK}) \wedge \neg f_1(\text{PK}).
\end{aligned}$$

B. The Facts Order Definition for the TLS Cryptographic Protocol

Orders between facts are as follows:

$$\begin{aligned}
& f_3(\text{cipher}) >_C f_2(\text{cipher}) >_C f_1(\text{cipher}) \\
& f_1(\text{cipher}) >_A f_2(\text{cipher}) >_A f_3(\text{cipher}) \\
& f_2(\text{bs}) >_C f_1(\text{bs}) \\
& f_2(IV) >_C f_1(IV) \\
& f_3(\text{key}) >_C f_2(\text{key}) >_C f_1(\text{key}) \\
& f_1(\text{key}) >_A f_2(\text{key}) >_A f_3(\text{key}) \\
& f_3(\text{mac}) >_I f_2(\text{mac}) >_I f_1(\text{mac})
\end{aligned}$$

$$\begin{aligned}
& f_1(\text{mac}) >_A f_2(\text{mac}) >_A f_3(\text{mac}) \\
& f_3(\text{mac-len}) >_I f_2(\text{mac-len}) >_I f_1(\text{mac-len}) \\
& f_1(\text{mac-len}) >_A f_2(\text{mac-len}) >_A f_3(\text{mac-len}) \\
& f_3(k\text{-len}) >_I f_2(k\text{-len}) >_I f_1(k\text{-len}) \\
& f_1(k\text{-len}) >_A f_2(k\text{-len}) >_A f_3(k\text{-len}) \\
& f_2(\text{PK}) >_{AU} f_1(\text{PK}) \\
& f_3(\text{PK}) >_{AU} f_1(\text{PK}) \\
& f_4(\text{PK}) >_{AU} f_2(\text{PK}) \\
& f_5(\text{PK}) >_{AU} f_2(\text{PK}) \\
& f_4(\text{PK}) >_{AU} f_3(\text{PK}) \\
& f_5(\text{PK}) >_{AU} f_3(\text{PK}) \\
& f_1(\text{PK}) >_A f_2(\text{PK}) \\
& f_1(\text{PK}) >_A f_3(\text{PK}) \\
& f_2(\text{PK}) >_A f_4(\text{PK}) \\
& f_2(\text{PK}) >_A f_5(\text{PK}) \\
& f_3(\text{PK}) >_A f_4(\text{PK}) \\
& f_3(\text{PK}) >_A f_5(\text{PK}).
\end{aligned}$$

C. The QoP Evaluation Rules Definition for the TLS Cryptographic Protocol

The evaluation rules are as follows:

$$\begin{aligned}
& f_1(\text{cipher}) \Rightarrow \text{Inf}^1(A) \\
& f_2(\text{cipher}) \Rightarrow \text{Inf}^2(A) \\
& f_3(\text{cipher}) \Rightarrow \text{Inf}^3(A) \\
& f_1(\text{cipher}) \Rightarrow \text{Inf}^1(C) \\
& f_2(\text{cipher}) \Rightarrow \text{Inf}^2(C) \\
& f_3(\text{cipher}) \Rightarrow \text{Inf}^3(C) \\
& f_1(\text{bs}) \Rightarrow \text{Inf}^1(C) \\
& f_2(\text{bs}) \Rightarrow \text{Inf}^2(C) \\
& f_1(IV) \Rightarrow \text{Inf}^1(C) \\
& f_2(IV) \Rightarrow \text{Inf}^2(C) \\
& f_1(\text{key}) \Rightarrow \text{Inf}^1(C) \\
& f_2(\text{key}) \Rightarrow \text{Inf}^2(C) \\
& f_3(\text{key}) \Rightarrow \text{Inf}^3(C) \\
& f_1(\text{key}) \Rightarrow \text{Inf}^1(A) \\
& f_2(\text{key}) \Rightarrow \text{Inf}^2(A) \\
& f_3(\text{key}) \Rightarrow \text{Inf}^3(A) \\
& f_1(\text{mac}) \Rightarrow \text{Inf}^1(I) \\
& f_2(\text{mac}) \Rightarrow \text{Inf}^2(I) \\
& f_3(\text{mac}) \Rightarrow \text{Inf}^3(I) \\
& f_1(\text{mac}) \Rightarrow \text{Inf}^1(A) \\
& f_2(\text{mac}) \Rightarrow \text{Inf}^2(A)
\end{aligned}$$

$$\begin{aligned}
& f_3(\text{mac}) \Rightarrow \text{Inf}^3(A) \\
& f_1(\text{mac-len}) \Rightarrow \text{Inf}^1(I) \\
& f_2(\text{mac-len}) \Rightarrow \text{Inf}^2(I) \\
& f_3(\text{mac-len}) \Rightarrow \text{Inf}^3(I) \\
& f_1(\text{mac-len}) \Rightarrow \text{Inf}^1(A) \\
& f_2(\text{mac-len}) \Rightarrow \text{Inf}^2(A) \\
& f_3(\text{mac-len}) \Rightarrow \text{Inf}^3(A) \\
& f_1(k\text{-len}) \Rightarrow \text{Inf}^1(I) \\
& f_2(k\text{-len}) \Rightarrow \text{Inf}^2(I) \\
& f_3(k\text{-len}) \Rightarrow \text{Inf}^3(I) \\
& f_1(k\text{-len}) \Rightarrow \text{Inf}^1(A) \\
& f_2(k\text{-len}) \Rightarrow \text{Inf}^2(A) \\
& f_3(k\text{-len}) \Rightarrow \text{Inf}^3(A) \\
& f_1(\text{PK}) \Rightarrow \text{Inf}^1(Au) \\
& f_2(\text{PK}) \Rightarrow \text{Inf}^2(Au) \\
& f_3(\text{PK}) \Rightarrow \text{Inf}^2(Au) \\
& f_4(\text{PK}) \Rightarrow \text{Inf}^3(Au) \\
& f_5(\text{PK}) \Rightarrow \text{Inf}^3(Au) \\
& f_1(\text{COM}) \wedge f_1(\text{cipher}) \Rightarrow \text{Inf}^4(A).
\end{aligned}$$

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The research is partially supported by the Grant “Reconcile: Robust Online Credibility Evaluation of Web Content” from Switzerland through the Swiss Contribution to the enlarged European Union.

References

- [1] B. Ksieżopolski, Z. Kotulski, and P. Szalachowski, “Adaptive approach to network security,” *Communications in Computer and Information Science*, vol. 39, pp. 233–241, 2009.
- [2] B. Ksieżopolski, Z. Kotulski, and P. Szalachowski, “On QoP method for ensuring availability of the goal of cryptographic protocols in the real-time systems,” in *Proceedings of the 1st European Teletraffic Seminar*, pp. 195–202, Poznań, Poland, 2011.
- [3] J. Jürjens, “Security and compliance in clouds. IT-compliance 2011,” in *Proceedings of the 4th Pan-European Conference*, 2011.
- [4] B. Ksieżopolski and Z. Kotulski, “Adaptable security mechanism for dynamic environments,” *Computers & Security*, vol. 26, no. 3, pp. 246–255, 2007.
- [5] B. Ksieżopolski, D. Rusinek, and A. Wierzbicki, “On the modelling of Kerberos protocol in the quality of protection modelling language (QoP-ML),” *Annales UMCS, Informatica*, vol. 12, pp. 69–81, 2012.

- [6] B. Ksiezopolski, D. Rusinek, and A. Wierzbicki, "On the efficiency modelling of cryptographic protocols by means of the Quality of Protection Modelling Language (QoP-ML)," in *Information and Communication Technology*, vol. 7804 of *Lecture Notes in Computer Science*, pp. 261–270, Springer, Berlin, Germany, 2013.
- [7] A. K. Agarwal and P. W. Wang, "On the impact of quality of protection in wireless local area networks with IP mobility," *Mobile Networks and Applications*, vol. 12, no. 1, pp. 93–110, 2007.
- [8] A. Luo, C. Lin, K. Wang, L. Lei, and C. Liu, "Quality of protection analysis and performance modeling in IP multimedia subsystem," *Computer Communications*, vol. 32, no. 11, pp. 1336–1345, 2009.
- [9] D. C. Petriu, C. M. Woodside, D. B. Petriu et al., "Performance analysis of security aspects in UML models," in *Proceedings of the 6th International Workshop on Software and Performance (WOPS '07)*, pp. 91–102, Buenos Aires, Argentina, February 2007.
- [10] M. Wazny and G. M. Wojcik, "Shifting spatial attention—numerical model of Posner experiment," *Neurocomputing*, vol. 135, pp. 139–144, 2014.
- [11] The official web page of the QoP-ML project, <http://qopml.org/>.
- [12] B. Ksiezopolski, "QoP-ML: quality of protection modelling language for cryptographic protocols," *Computers and Security*, vol. 31, no. 4, pp. 569–596, 2012.
- [13] E. LeMay, W. Unkenholz, D. Parks, C. Muehrcke, K. Keefe, and W. H. Sanders, "Adversary-driven state-based system security evaluation," in *Proceedings of the 6th International Workshop on Security Measurements and Metrics (MetriSec '10)*, September 2010.
- [14] S. Lindskog, *Modeling and tuning security from a quality of service perspective [Ph.D. thesis]*, Department of Computer Science and Engineering, Chalmers University of Technology, Goteborg, Sweden, 2005.
- [15] C. S. Ong, K. Nahrstedt, and W. Yuan, "Quality of protection for mobile applications," in *Proceedings of the IEEE International Conference on Multimedia & Expo*, pp. 137–140, 2003.
- [16] P. Schneck and K. Schwan, "Authenticast: an adaptive protocol for high-performance, secure network applications," Tech. Rep. GIT-CC-97-22, 1997.
- [17] Y. Sun and A. Kumar, "Quality-of-protection (QoP): a quantitative methodology to grade security services," in *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS '08)*, pp. 394–399, June 2008.
- [18] J. Jürjens, *Secure System Development with UML*, Springer, 2007.
- [19] T. D. Breaux, A. I. Antón, and E. H. Spafford, "A distributed requirements management framework for legal compliance and accountability," *Computers & Security*, vol. 28, no. 1-2, pp. 8–17, 2009.
- [20] M. Theoharidou, P. Kotzanikolaou, and D. Gritzalis, "A multi-layer criticality assessment methodology based on interdependencies," *Computers and Security*, vol. 29, no. 6, pp. 643–658, 2010.
- [21] ISO/IEC 27001:2005, "Information technology—security techniques—information security management systems—requirements," 2005.
- [22] C. Lambrinouidakis, S. Gritzalis, F. Dridi, and G. Pernul, "Security requirements for e-government services: a methodological approach for developing a common PKI-based security policy," *Computers & Security*, vol. 26, no. 16, pp. 1873–1883, 2003.
- [23] T. Zurek, "Modelling of a fortiori reasoning," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10772–10779, 2012.
- [24] L. Leszczynski, *Zagadnienia Teorii Stosowania Prawa: Issues of Theory of Application of Law*, Zakamycze, Krakow, Poland, 2001.
- [25] W. W. Vasconcelos, A. García-Camino, D. Gaertner, J. A. Rodríguez-Aguilar, and P. Noriega, "Distributed norm management for multi-agent systems," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5990–5999, 2012.
- [26] H. Prakken and G. Sartor, "A dialectical model of assessing conflicting arguments in legal reasoning," *Artificial Intelligence and Law*, vol. 4, no. 3-4, pp. 331–368, 1996.
- [27] R. A. Kowalski and F. Toni, "Abstract argumentation," *Artificial Intelligence and Law*, vol. 4, no. 3-4, pp. 275–296, 1996.
- [28] G. A. W. Vreeswijk, "Abstract argumentation systems," *Artificial Intelligence*, vol. 90, no. 1-2, pp. 225–279, 1997.
- [29] J. Hage, *Studies in Legal Logic*, Springer, 2005.
- [30] L. W. Torre and Y. H. Tan, "The many faces of defeasibility in defeasible deontic logic," in *Defeasible Deontic Logic*, D. Nute, Ed., pp. 79–121, Springer, 1997.
- [31] H. Prakken and G. Vreeswijk, "Logics for defeasible argumentation," in *Handbook of Philosophical Logic*, D. Gabbay, Ed., Kluwer Academic Publisher, 1983.
- [32] RFC 5246: The Transport Layer Security (TLS) Protocol v.1.2, 2008.
- [33] FIPS 180-3: Secure Hash Standard (SHS).

Research Article

Malware Analysis Using Visualized Image Matrices

KyoungSoo Han,¹ BooJoong Kang,² and Eul Gyu Im³

¹ Department of Computer and Software, Hanyang University, Seoul 133-791, Republic of Korea

² Department of Electronics and Computer Engineering, Hanyang University, Seoul 133-791, Republic of Korea

³ Division of Computer Science and Engineering, Hanyang University, Seoul 133-791, Republic of Korea

Correspondence should be addressed to Eul Gyu Im; imeg@hanyang.ac.kr

Received 14 March 2014; Accepted 19 May 2014; Published 16 July 2014

Academic Editor: Fei Yu

Copyright © 2014 KyoungSoo Han et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a novel malware visual analysis method that contains not only a visualization method to convert binary files into images, but also a similarity calculation method between these images. The proposed method generates RGB-colored pixels on image matrices using the opcode sequences extracted from malware samples and calculates the similarities for the image matrices. Particularly, our proposed methods are available for packed malware samples by applying them to the execution traces extracted through dynamic analysis. When the images are generated, we can reduce the overheads by extracting the opcode sequences only from the blocks that include the instructions related to staple behaviors such as functions and application programming interface (API) calls. In addition, we propose a technique that generates a representative image for each malware family in order to reduce the number of comparisons for the classification of unknown samples and the colored pixel information in the image matrices is used to calculate the similarities between the images. Our experimental results show that the image matrices of malware can effectively be used to classify malware families both statically and dynamically with accuracy of 0.9896 and 0.9732, respectively.

1. Introduction

Malware authors have been generating new malware and malware variants through various means, such as reusing modules or using automated malware generation tools. As some modules for malicious behavior are reused in malware variants, malware variants of the same family may have similar binary patterns, and these patterns can be used to detect malware and to classify malware families. Moreover, most antivirus programs focus on malware signatures, that is, string patterns, to detect malware [1]. However, various detection avoidance techniques such as obfuscation or packing techniques are applied to malware variants to avoid detection by signature-based antivirus programs and to make analysis difficult for security analysts [2, 3]. With the help of malware generation techniques, the amount of malware is increasing every year.

Although security analysts and researchers have been studying various analysis techniques to deal with malware variants, they cannot be analyzed completely because the

malware in which avoidance techniques are applied is exponentially increasing. Therefore, new malware analysis techniques are required to reduce the burden on security analysts. Recently, several malware visualization techniques have been proposed to help security analysts to analyze malware.

In this paper, we propose a novel method to analyze malware visually to classify malware families. The proposed method converts the opcode sequences extracted from the malware into images called image matrices and calculates the similarities between each image. In addition, we apply the proposed method to the execution traces extracted through dynamic analysis, so that malware employing detection avoidance techniques such as obfuscation and packing can be analyzed. To reduce the computational overheads, we extract the opcode sequences only from the blocks that are related to staple behaviors, such as functions and application programming interface (API) calls, by using a major block selection technique [2]. Representative images of individual malware families are generated and are used to classify the unknown sample rapidly. Using these image matrices, we

obtain the similarities between the images after the RGB-colored pixel information of the images is vectorized and the pixel similarities are calculated.

This paper is composed as follows. In Section 2, malware analysis-related studies are described. In Section 3, malware analysis methods using visualized opcode sequences and the methods to calculate similarity are proposed, and the experimental results are presented in Section 4. Finally, in Section 5, conclusions and future directions are provided.

2. Related Work

In general, malware analysis methods to detect and classify malware can be categorized as either static or dynamic analyses [4]. In the static analysis of malware, various methods such as control flow graph (CFG) analysis [5–7], call graph analysis [8, 9], byte level analysis [10], instruction-based analysis [11–14], and similarity-based analysis [15, 16] have been proposed.

CFGs are generated by dividing the instructions extracted through disassembling into blocks and by connecting the directed edges between the blocks. Some malware analysis methods using these CFGs as signatures have been proposed. Cesare and Xiang [5] proposed a method that defines CFGs as signatures in string form that consist of a list of graph edges for the ordered nodes and that measures the similarities among signatures by using the Dice coefficient algorithm [17]. Bonfante et al. [6] proposed a method that converts the CFGs into tree-based finite state machines through syntactic analysis and semantic analysis and then uses them as signatures. Briones and Gomez [7] proposed an automated classification system based on CFGs. The CFGs are summarized as three tuples including the number of basic blocks, the number of edges, and the number of subcalls, and then two functions can be compared. However, if the complex information is summarized into a small size, high false alarms may occur.

There is much research aimed at detecting malware based on information such as system-calls, functions, and API calls, which is used for malware execution in operating systems. Shang et al. [8] proposed a method that generates function-call graphs, which represent the caller and callee relationships between functions as signatures of malware samples, and they then compute the similarities by using those function-call graph signatures. Kinable and Kostakis [9] classified malware using the call graph clustering technique. Their proposed method generated the call graphs against the functions included in the malware samples, and they performed the clustering based on the structural similarity scores of the call graphs calculated through the graph edit distance algorithm.

Statistical information regarding the instructions extracted through disassembling can be used in the static analysis of malware. Rad and Masrom [11] proposed a method based on the instruction frequencies in order to classify metamorphic malware. Since instruction frequencies are mostly not changed, even though the obfuscation techniques are applied to the malware, the instruction frequencies can become the features of malware. Therefore, their proposed method calculated a distance by using the instruction frequencies extracted from each malware sample,

and they then classified metamorphic malware by using the distance value. Bilar [12] showed that there were different instruction frequencies in different malware. Particularly, they showed that rare instructions in malware could become better predictors to classify malware than other instruction could. Han et al. [13] proposed a method using instruction frequencies. The proposed method generated instruction sequences that were sorted according to the instruction frequencies, and they showed that the distances between instruction sequences from the same malware family had low distance values. Santos et al. [14] proposed a malware classification method using n -gram instruction frequencies in which n -gram instructions included n -instructions. In the proposed method, they generated the vectors for each n -instruction sequence and used some of the vectors as signatures.

In addition, dynamic analysis methods including tainting, behavior-based methods, and API call monitoring have been proposed. Egele et al. [18] proposed a method using tainting techniques, which tracks the behaviors related to the flow of information that are processed by any browser helper object (BHO). If the BHO leaks sensitive information to the outside, the BHO is classified as malware. Fredrikson et al. [19] proposed a method that automatically extracts the characteristics of behaviors by using graph mining techniques. Their proposed method made clusters by identifying core CFGs for each similar malicious behavior in a malware family, and these were then generalized as a significant behavior. Furthermore, methods based on dynamic monitoring techniques using an emulator have been proposed. Vinod et al. [20] traced malware API calls via dynamic monitoring within an emulator and measured their frequencies to extract critical APIs. Miao et al. [21] developed a tool called the “API Capture” that extracts the major characteristics automatically, such as system-call arguments, return values, and error conditions by monitoring malware behavior in an emulator.

Even though there are many static and dynamic analyses methods available, new techniques that can complement existing techniques are still needed to improve malware analysis performance and conveniences of analysis by security analysts. Recently, several visualization methods have been proposed to help security analysts to observe the features and behaviors of malware [22]. To visualize malware behavior, Trinius et al. [23] proposed a method that visualized the percentages of API calls as well as malware behavior into each of two images called a “treemap” and “thread graph,” respectively. Saxe et al. [24] developed a system that generated two types of images. One image showed the system-call sequences extracted from malware system-call behavior logs, and the other image showed similarities and differences between selected samples. Conti et al. [25] proposed a visualizing system that shows the images for the byte information of malware samples such as byte values, byte presence, and duplicated sequences of bytes contained within a sample. Anderson et al. [26] proposed a method to show the similarities between malware samples in an image named a “heatmap.” Nataraj et al. [27] converted the byte information into gray-scale images and classified the malware using image processing. After generating images

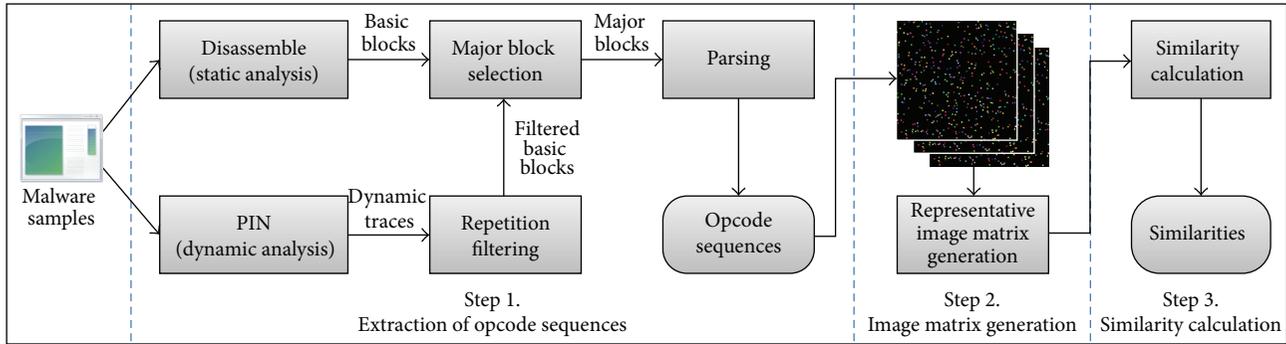


FIGURE 1: Overview of the proposed method.

using byte values, they applied an abstract representation technique for the scene image, that is, GIST [28, 29], to compute texture features. Moreover, they proved that the binary texture analysis techniques using image processing could classify malware more quickly than existing malware classification methods could [30]. However, since the texture analysis method has large computational overheads, the proposed method has problems in processing a large amount of malware [31].

In this paper, we propose a novel analysis method using image matrices to represent malware visually so that the features of the malware can be easily detected and the similarities between different malware samples can be calculated faster than with other visualization methods.

3. Our Proposed Method

3.1. Overview. Our proposed visualized malware analysis method consists of three steps, as shown in Figure 1. In Step 1, opcode sequences are extracted from malware binary samples or dynamic execution traces. Then, image matrices in which the opcode sequences are recorded as RGB-colored pixels are generated in Step 2. In Step 3, the similarities between the image matrices are calculated. In the following sections, each step is explained in detail.

3.2. Extraction of Opcode Sequences. Figure 2 shows the process to extract opcode sequences from malware binary samples for Step 1 through static analysis or dynamic analysis.

3.2.1. Basic Block Extraction. To extract opcode sequences from malware binary samples, the binary sample files are first disassembled and divided into basic blocks, using disassembling tools, such as IDA Pro [32] or OllyDbg [33]. However, if obfuscation or packing techniques are applied in malware samples, static analysis using a disassembler is not feasible [34, 35]. Therefore, some malware samples (in which obfuscation or packing techniques are applied) need to be executed in a dynamic analysis environment [36].

In dynamic analysis, as shown in Figure 3, some repeated instruction sequences are included in the dynamic execution traces because a program may have some loops or repeated calls, and these repeated sequences can increase the size

of not only the execution traces, but also the processing overheads. Kang et al. [37] proposed a repetition filtering method for dynamic execution traces. Our filtered basic blocks are extracted from the dynamic execution traces after the repetition filtering method is applied. Finally, if basic blocks are extracted from malware samples or dynamic execution traces, then major blocks are selected from the basic blocks by our proposed technique, which is explained in the next section.

3.2.2. Major Block Selection. The malware analysis method proposed in this paper does not target all of the basic blocks from the binary disassembling results or dynamic execution traces. If all the basic blocks are used for analysis, then some blocks for binary file execution in an operating system are included in the basic blocks. Moreover, many meaningless blocks may be included in the basic blocks extracted from malware samples. As a result, the number of basic blocks that have to be analyzed by the security analysts is increased and distinguishing malware features becomes difficult. In addition, the number of comparisons between the basic blocks from two malware samples is also increased dramatically. On the contrary, if the number of unnecessary blocks can be reduced as much as possible in the malware analysis, the analysis time cost for not only the individual malware sample, but also a large number of malware samples can be reduced. Therefore, we selected some blocks relating to suspicious behaviors and functions from among the entire set of basic blocks.

As shown in Figure 4, the blocks selected as major blocks are those that include the *CALL* instruction, which is used to invoke APIs, library functions, and other user-defined functions. This is because not only user-defined functions, but also various system calls are used to implement the behaviors and functions of most programs. If blocks that include these function invocation instructions are used in malware analysis, malware features can be extracted [2]. Through a major block selection technique, the image matrix generating time is reduced by recording only those selected blocks in the image matrix.

3.2.3. Opcode Sequence Extraction. To extract malware features, as shown in Figure 5, the opcode sequences in the

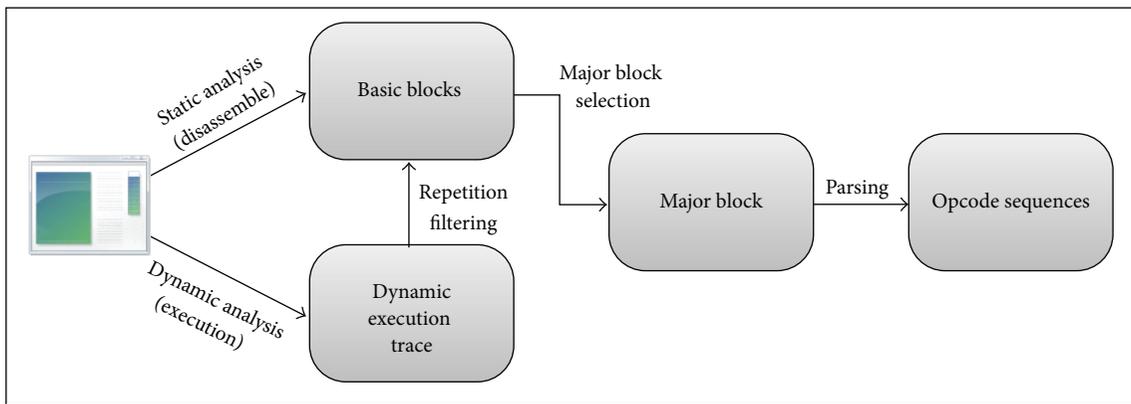


FIGURE 2: Opcode sequence extraction procedure.

0042A68B	Main	JBE	SHORT Exploit_0042A69C
0042A68D	Main	MOV	AL, BYTE PTR DS: [EDX]
0042A68F	Main	INC	EDX
0042A690	Main	MOV	BYTE PTR DS: [EDI], AL
0042A692	Main	INC	EDI
0042A693	Main	DEC	ECX
0042A694	Main	JNZ	SHORT Exploit_0042A68D
0042A68D	Main	MOV	AL, BYTE PTR DS: [EDX]
0042A68F	Main	INC	EDX
0042A690	Main	MOV	BYTE PTR DS:[EDI],AL
0042A692	Main	INC	EDI
0042A693	Main	DEC	ECX
0042A694	Main	JNZ	SHORT Exploit_0042A68D
0042A68D	Main	MOV	AL, BYTE PTR DS: [EDX]
0042A68F	Main	INC	EDX
0042A690	Main	MOV	BYTE PTR DS: [EDI], AL
0042A692	Main	INC	EDI
0042A693	Main	DEC	ECX
0042A694	Main	JNZ	SHORT Exploit_0042A68D
0042A68D	Main	MOV	AL, BYTE PTR DS: [EDX]
0042A68F	Main	INC	EDX
0042A690	Main	MOV	BYTE PTR DS: [EDI], AL
0042A692	Main	INC	EDI
0042A693	Main	DEC	ECX
0042A694	Main	JNZ	SHORT Exploit_0042A68D
0042A696	Main	JMP	Exploit_0042A5FE
0042A5FE	Main	ADD	EBX, EBX

FIGURE 3: An example of repeated instruction sequences.

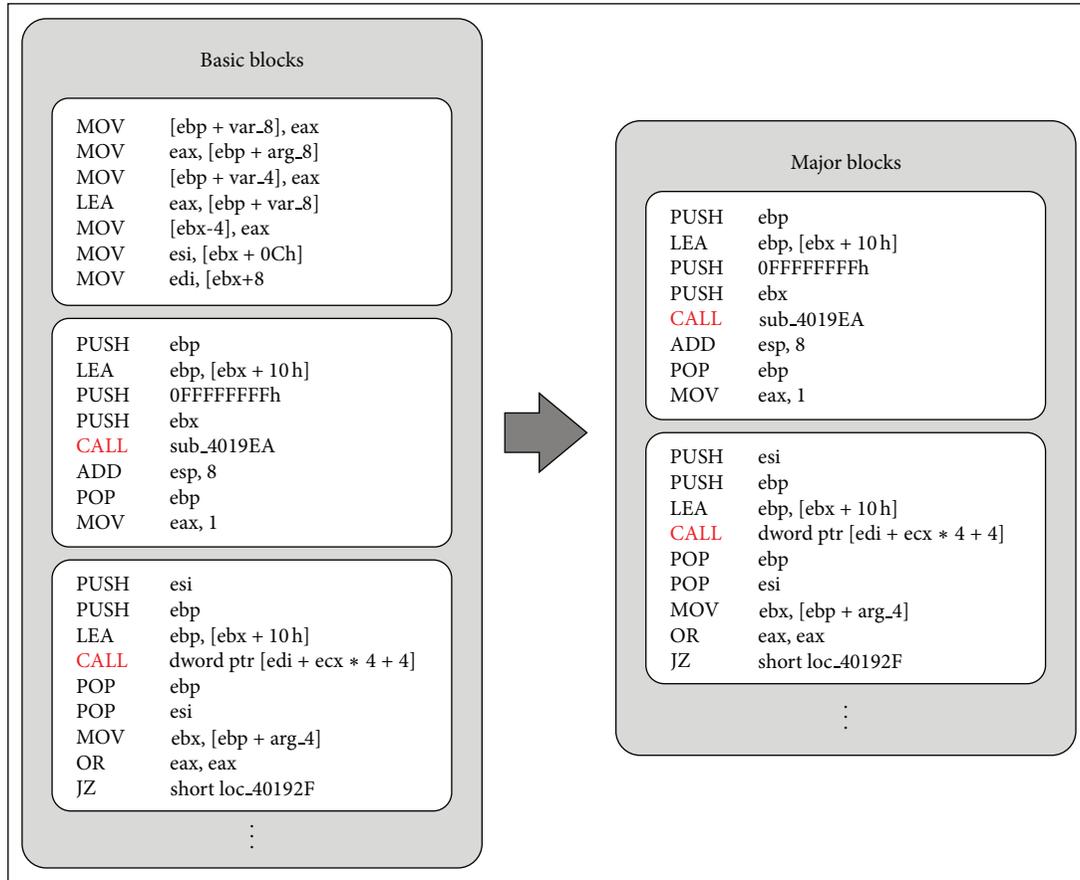


FIGURE 4: Major block selection.

individual major blocks are used as malware information. From each opcode, only the first three characters are used to generate information for the block. The reasons for using a three-character opcode are as follows. From the entire set of opcodes used in the Intel x86 assembly language, 41.4% of them have three characters, and the appearance frequencies of these opcodes within the binary files are higher than for other opcodes. On the other hand, 28.8% of opcodes have four characters, 17.8% have five characters, and 5.2% have over six characters, respectively; thus, their appearance frequencies are relatively low. In addition, since the meanings of the individual opcodes are maintained even though they are reduced to three characters, the different opcodes can be distinguished. For example, four-character opcodes such as *PUSH* are reduced to three characters, *PUS*, and two-character opcodes such as *OR* are expanded by adding a blank character to a three-character opcode. Then, these three-character opcodes are concatenated together, and the character string is used to represent the block as an opcode sequence, which is used to generate image pixels in an image matrix in the next step.

3.3. *Generation of the Image Matrix.* Figure 6 shows the procedure for Step 2 that converts the opcode sequences into

pixels in an image matrix. A hash function is used to decide the *X-Y* coordinates and RGB colors of the pixels.

To visualize a binary file as an image matrix, both the length and the width of an image matrix are initialized to 2^n , where n is selected by the users. To reduce the probability of collisions of the hash function; n should be large enough. In our experiments, we selected n as 8 to minimize collisions.

The coordinate-defining module and the RGB color-defining module are used to generate image matrices. First, the coordinate-defining module defines the (x, y) coordinates of pixels on the image matrix of each code block. Second, the RGB color-defining module defines the color values of pixels on the image matrix. RGB colors are defined by calculating values of 8 bits each for the red, green, and blue colors.

SimHash [38] is applied to opcode sequences extracted in Step 1 in order to define both the coordinates and the color values of the pixels. *SimHash* is a local-sensitive hash function used in the similar sentence detection system, which assumes that if the input values are similar, then the output values will also be similar. That is, since *SimHash* tokenizes the input strings and generates hash values for each token, if a few tokens are different in two input strings, then the generated hash values are not completely different, but are similar.

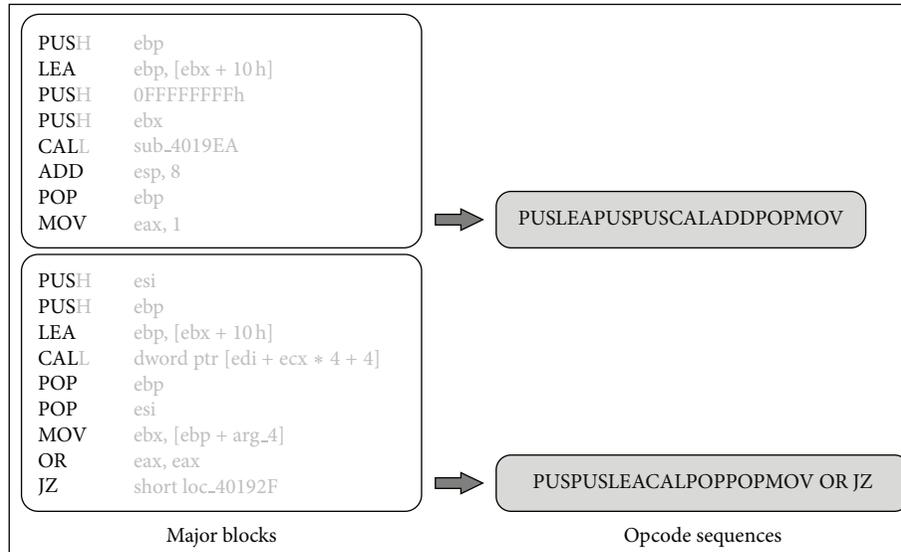


FIGURE 5: Opcode sequences used as malware information.

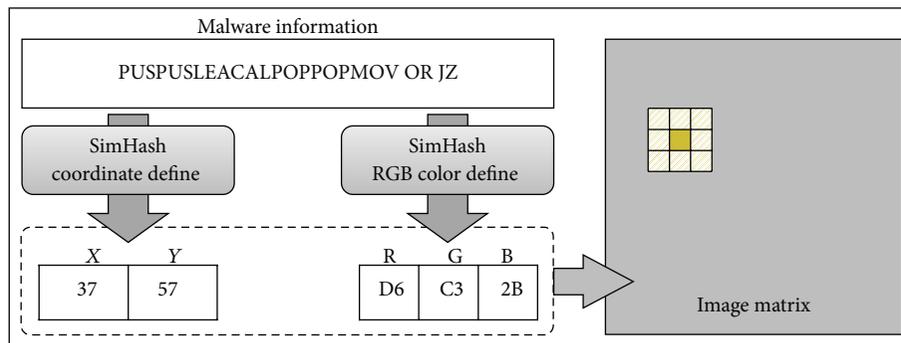


FIGURE 6: Generating images using opcode sequences.

Therefore, if the character strings of the opcode sequences are similar, then the outputs will be similar, and they will map onto similar coordinates in an image matrix.

Once the coordinates and RGB colors of the individual pixels have been defined, RGB-colored images are recorded on the individual coordinates of image matrices. To provide human analysts with a more convenient visual analysis, pixels around the defined coordinates are recorded simultaneously. As shown in Figure 7, nine pixels from $(x - 1, y - 1)$ to $(x + 1, y + 1)$ around an (x, y) coordinate for a block are recorded.

If the images overlap each other because the coordinates defined for multiple opcode sequences are adjacent, as shown in Figure 8, the sums of RGB colors become new pixel colors. If the result of a color summing exceeds 255 (0xFF), the result will be set to 255. For example, if RGB_1 is (255, 0, 0) and RGB_2 is (0, 176, 50), the new color will become (255, 176, 50).

The number of pixels recorded on an image matrix varies according to the major blocks, and the number of overlapping pixels will increase as the number of images increases. If there are too many overlapping images, then the size of the image matrix should be increased.

$x - 1,$ $y - 1$	$x,$ $y - 1$	$x + 1,$ $y - 1$
$x - 1,$ y	x, y	$x + 1,$ y
$x - 1,$ $y + 1$	$x,$ $y + 1$	$x + 1,$ $y + 1$

FIGURE 7: Nine pixels for one opcode sequence.

3.4. Representative Image Matrix Extraction. Since many malware variants exist in each malware family, as the number of malware samples increases, the total amount and time of the similarity calculation increase, too. Therefore, we

1	1	1		
1	1	1		
1	1	3	2	2
		2	2	2
		2	2	2

$$R_3 = \min(R_1 + R_2, FF) = \min(FF + 00, FF) = FF$$

$$G_3 = \min(G_1 + G_2, FF) = \min(FF + B0, FF) = B0$$

$$B_3 = \min(B_1 + B_2, FF) = \min(FF + 50, FF) = 50$$

$$\therefore RGB_3 = (FF, B0, 50)$$

FIGURE 8: Method of recording overlapping pixels.

extracted a representative image matrix of each malware family to reduce the costs of malware similarity calculations. That is, when a new malware sample is found, the amount of time to calculate the similarity is reduced by comparing the image matrix of the new malware with the image matrices that represent individual malware families instead of comparing it with all of the image matrices of the existing malware samples.

As shown in Figure 9, to extract a representative image matrix for each malware family, image matrices are generated for samples in malware families. Then, the representative image matrix is extracted by recording only the common pixels that have same coordinates and RGB colors from the image matrices of individual malware samples in the same family. Figure 9 shows an example of the generation of representative images of malware families.

3.5. Similarity Calculation Using Image Matrices. The advantage of the similarity calculation using the image matrices is a faster performance than with exact matching using the string type of opcode sequences, even though there are some extra false positives due to hash collision. When using the string, the time complexity is defined as $O(n^2)$ due to the process of finding pairs of exactly matched strings. However, if the image matrices are used to calculate similarities, since the coordinates and colors of the opcode sequences are defined through *SimHash*, the process of finding the pairs is skipped. Therefore, the time complexity of the similarity calculations using the image matrices is defined as $O(n)$, because only the color information of the pixels recorded on the same coordinates in both image matrices is used to calculate the similarities between the image matrices.

Pixel similarity calculations are carried out first for pixels in each image matrix. The most important consideration in a similarity calculation in this case is that only those RGB

color pixels recorded in the individual image matrices should be used. Image matrices have RGB-colored pixels on square images with black backgrounds. If black pixels are also used in similarity calculations, the similarities between samples from different malware families can be calculated as very high. Therefore, when the similarities of the image matrices are calculated, the following cases are considered for pixels on the same coordinates in the two image matrices, as shown in Figure 10. In this case, the vector angular-based distance measurement algorithm is used to calculate the similarities between color pixels. This algorithm calculates similarity values by expressing the color pixels constituting each image as 3D vectors, as shown in (1), and then using the angle information and size information

- (a) Case 1: if all of the pixels in the areas of both image matrices are black, the pixel similarity calculation will not be carried out and the next pixel will be selected.
- (b) Case 2: if one pixel in a selected area is black and the corresponding pixel in the other image is colored, the pixel similarity will be defined as 0.
- (c) Case 3: if both pixels are not black but colored, the color pixel similarity will be calculated using the vector angular-based distance measurement algorithm [39], as follows:

$$\delta(x_i, x_j) = \left[1 - \frac{2}{\pi} \cos^{-1} \left(\frac{x_i \cdot x_j}{|x_i| |x_j|} \right) \right] \left[1 - \frac{|x_i - x_j|}{\sqrt{3} \cdot 255^2} \right]. \quad (1)$$

The similarity values of the image matrix when considering individual cases are calculated, using the results from the pixel similarity calculations, as shown in (2). That is, the sum of pixel similarity values calculated in case 3 is divided by the number of pixels calculated in cases 2 and 3 to calculate the average:

$$\text{Sim}(A, B) = \frac{\text{sum of pixel similarity values in case 3}}{\# \text{ of pixels in case 2 and case 3}}. \quad (2)$$

4. Experimental Results

4.1. Experimental Data and Environment. Using the visual analysis tools implemented in this paper, and the malware samples shown in Table 1, image matrices were generated, and similarity calculations were performed. First, set A consists of 290 malware samples from 16 families in which the detection avoidance techniques, such as obfuscation and packing, are not applied. These malware samples are used to extract the basic blocks through static analysis using a disassembler. Second, set B consists of 560 malware samples from 14 families in which the packed and nonpacked malware samples coexist. We used these malware samples to generate dynamic execution traces through the PIN tool in a dynamic analysis environment, and the filtered basic blocks are extracted from the dynamic execution traces through the repetition filtering technique, as explained in Section 3.2.1.

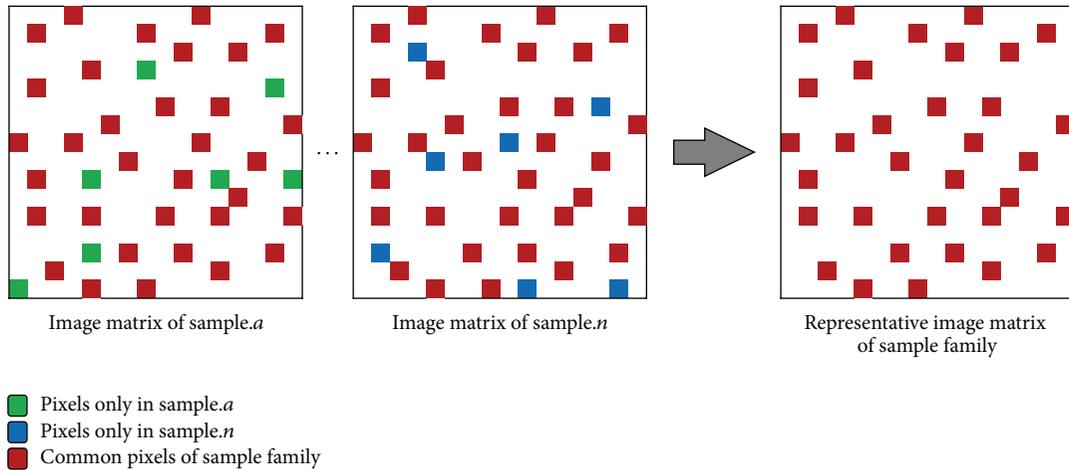


FIGURE 9: Representative image matrix extraction.

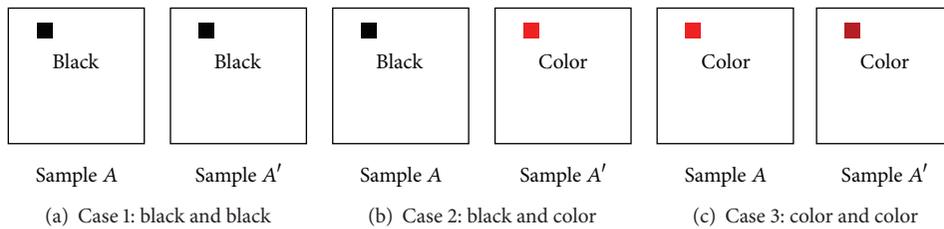


FIGURE 10: Three cases considered in pixel similarity calculations.

For the experiments, we constructed an experimental environment consisting of the analysis server, malware server, and monitoring machine, as shown Figure 11. We set up VMware vSphere ESXi 5.1 in the analysis server, which has an Intel Xeon E5-1607 processor and 24 GB of main memory, and we installed two Windows operating systems (OSs) as guest OSs. In the first Windows OS, the dynamic execution traces were extracted through the PIN. In the other Windows OS, the image matrices were generated and similarities were calculated through our visual analysis tool. Malware samples that are provided for the analysis server and the dynamic execution traces extracted from the analysis server are stored in the malware server. The monitoring machine controls the analysis server through the PowerCLI tool that is the remote command line interface.

4.2. Experiments with Static Analysis. For the experiments in this section, we disassembled the malware samples within set A and extracted major blocks from the basic blocks. We then generated the image matrices using opcode sequences of those major blocks and analyzed the similarities among them.

4.2.1. Image Matrix Generation. In this paper, we set the sizes of the generated image matrices to 256×256 pixels for the experiments. As shown in Table 2, the reasons for using this image matrix size can be briefly summarized as the middle ground between file size, similarity calculation time, and

classification accuracy. The accuracy was calculated by using (3):

$$\text{Accuracy} = \frac{\# \text{ of correctly classified malware samples}}{\# \text{ of total malware samples}} \tag{3}$$

Figure 12 shows examples of the image matrices generated from the malware samples of individual families within set A. Only three image matrices for each malware family and one representative image matrix extracted by recording only those pixels commonly existing in all image matrices were included. Since the number of opcode sequences used as malware information varied, the number of pixels recorded on the image matrices differed. In the case of malware, many of the same or similar RGB-colored pixels are found among the image matrices of malware samples classified as the same family. However, even if pixels are recorded on the same coordinates of different image matrices, the pixel similarities have different values if the RGB color information of the relevant pixels is different. Our results show that image matrices of variants included in the same malware family can be shown to be similar and that clear differences exist among malware samples from different families.

Figure 13 shows the image matrix differences before and after the application of the major block selection technique. The image progression indicates that the number of pixels recorded in the image matrices decreases because of the selection of major blocks from among the basic blocks. The

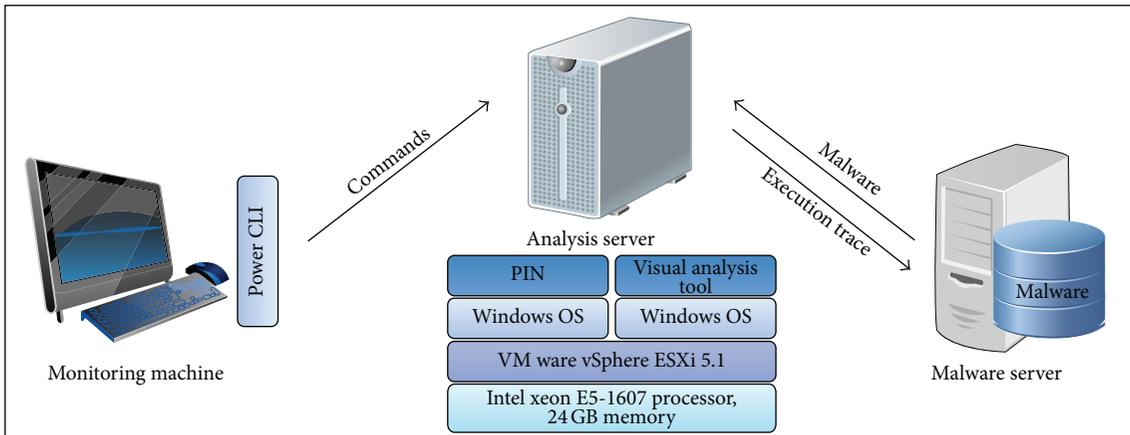


FIGURE 11: Experimental environment.

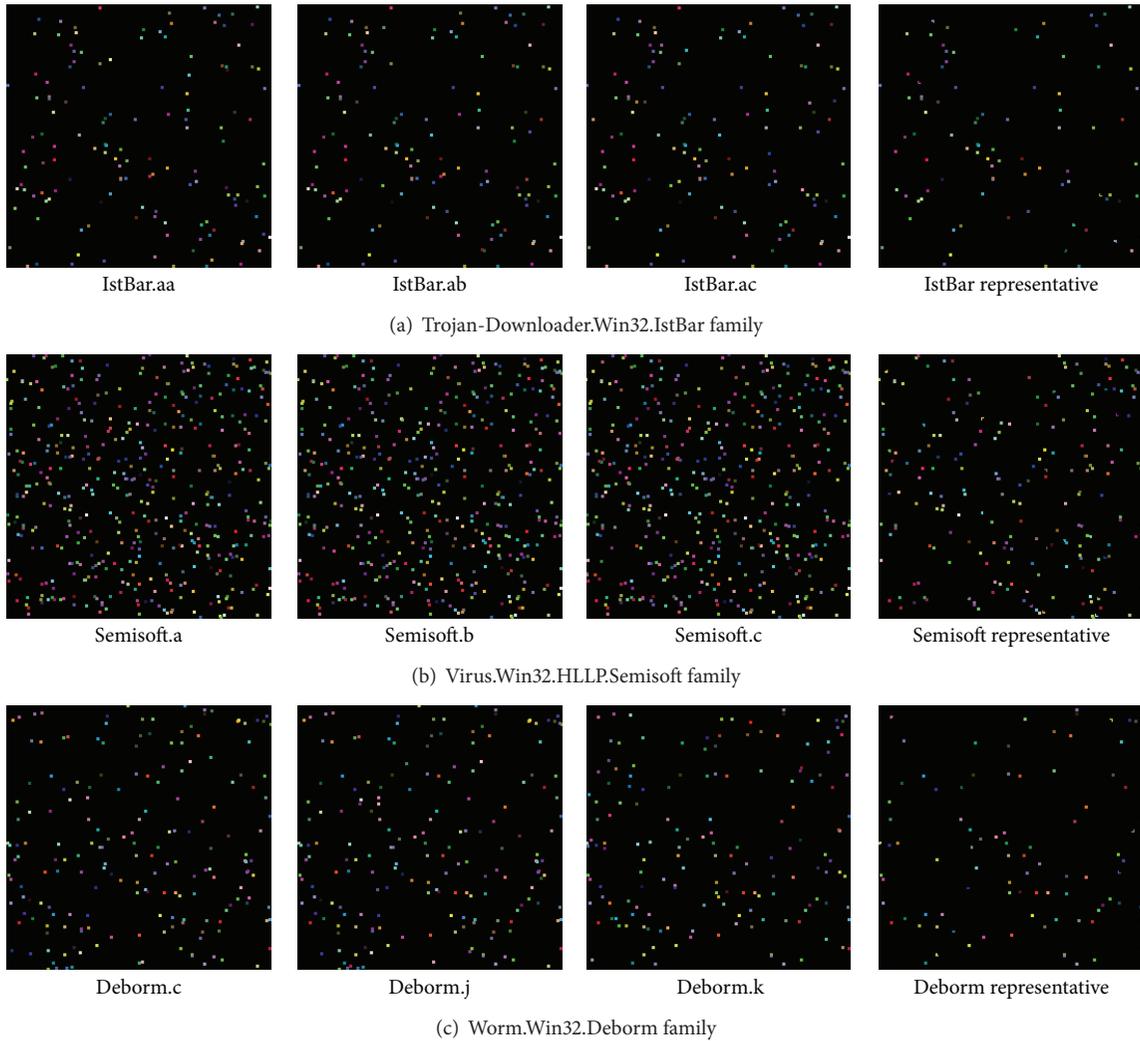


FIGURE 12: Image matrices of malware family samples.

TABLE 1: Malware samples.

Set	Type	Family	Number of variants
A	Email-Worm	Klez	9
	Trojan-DDos	Boxed	27
		IstBar	41
		Ladder	5
		Lemmy	26
	Trojan-Downloader	Mediket	43
		OneClickNetSearch	11
	Trojan-Dropper	Tab	8
		Eva	6
		Evol	3
	Virus	Fosforo	4
		Gpcode	35
		Halen	7
		Semisoft	14
		Zepp	11
	Worm	Deborm	40
	Backdoor	Agobot	40
		Bifrose	40
IRCBot		40	
SdBot		40	
Trojan	Dialer	40	
	StartPage	40	
Trojan-Downloader	Banload	40	
	Dyfuca	40	
	Swizzor	40	
Trojan-Spy	Bancos	40	
	Banker	40	
Email-Worm	Bagle	40	
IM-Worm	Kelvir	40	
P2P-Worm	SpyBot	40	

TABLE 2: The selection of image matrix size.

Size (resolution)	File size (KB)	Similarity calculation time (ms, avg.)	Classification accuracy (avg.)
128 × 128	48	5.3	0.9595
256 × 256	192	18.2	0.9814
512 × 512	768	66.4	0.9929

similarity changes after the application of the major block selection is described in the next subsection.

4.2.2. Major Block Selection. Similarity calculations of the image matrices after the application of major block selection are shown in Figures 14 and 15. When the major block selection technique was applied, the similarity changes ranged from a minimum of 0.002 (the Tab family) to a maximum of 0.147 (the Lemmy family) among the malware samples in the same families. The results of the similarity calculations for different families showed that the changes ranged from a minimum of 0.001 (the Eva family) to a maximum of

TABLE 3: Arbitrarily selected malware samples as unknown.

Number	Malware sample	Most similar family (similarity)
1	Klez.j	Klez (0.181)
2	Boxed.g	Boxed (0.190)
3	IstBar.gvf	IstBar (0.302)
4	Ladder.f	Ladder (0.339)
5	Lemmy.z	Lemmy (0.341)
6	Mediket.ec	Mediket (0.325)
7	OneClickNetSearch.k	OneClickNetSearch (0.268)
8	Tab.gd	Tab (0.348)
9	Eva.g	Eva (0.317)
10	Evol.c	Evol (0.329)
11	Fosforo.d	Fosforo (0.341)
12	Gpcode.x	Gpcode (0.306)
13	Halen.2619	Halen (0.352)
14	Semisoft.n	Semisoft (0.281)
15	Zepp.d	Zepp (0.265)
16	Deborm.ai	Deborm (0.339)
17	Agobot.02.a	Mediket (0.042)
18	SdBot.04.a	Boxed (0.054)

0.053 (the Klez family). As a result, while the range of the similarity values among the malware samples in the same family is more than 0.6, the range of the similarity values among malware samples from different families is below 0.1. Therefore, although the similarity values change due to applying the major block selection technique, we can reduce the image matrix generation time and can find obvious differences in the similarity values.

4.2.3. Representative Image Matrix Extraction. For this experiment, we selected an arbitrary malware sample not included in data set A as the unknown sample. We then analyzed the similarity calculation time and the similarity values of an image matrix for the unknown sample with 290 image matrices of all malware samples and with 16 representative image matrices extracted from individual families.

Figure 16 shows the results of the similarity calculations of an unknown sample both with all of the image matrices of malware samples and with the representative image matrices of individual families. When all of the image matrices were used, the Tab family was found to have an average similarity value of 0.781, while all the other families had values smaller than 0.05. When representative image matrices of individual families were used, the average similarity of the Tab family had a value of 0.348, while the other families had values of less than 0.03. Therefore, the unknown sample is expected to be a variant of the Tab family. In fact, the diagnostic name of the unknown sample used for this experiment was Trojan-Dropper. Win32.Tab.gd.

Table 3 shows the list of the malware samples selected as unknown samples for this experiment and it includes the results of the similarity calculations using representative image matrices. These malware samples except for Agobot.02.a and Sdbot.04.a were detected as variants of each

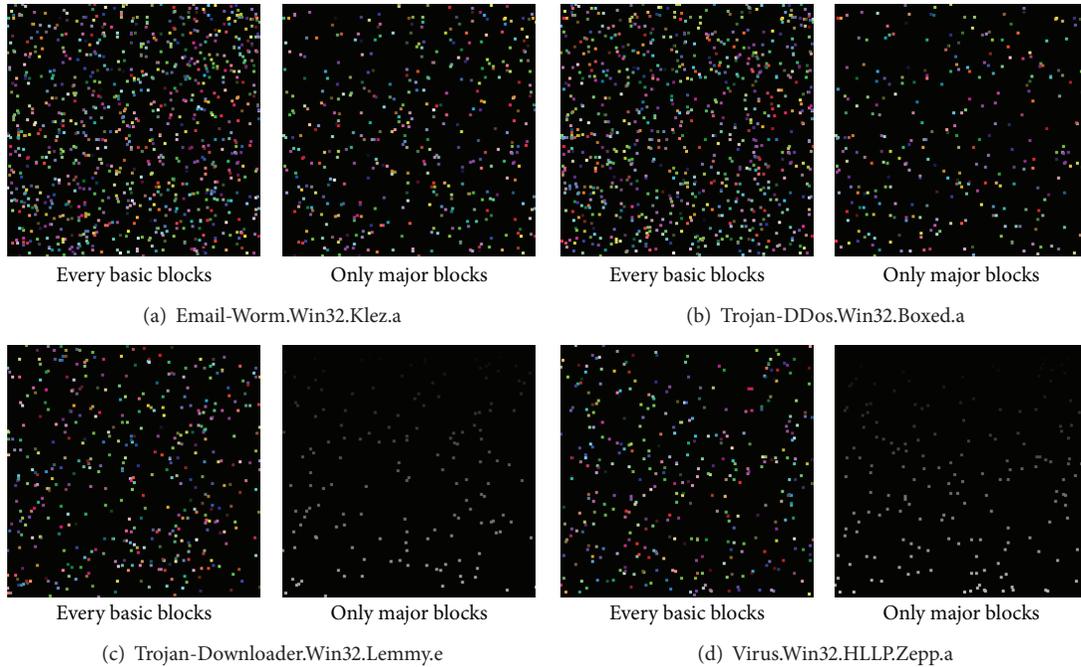


FIGURE 13: Comparison of image matrices with and without major block selection.

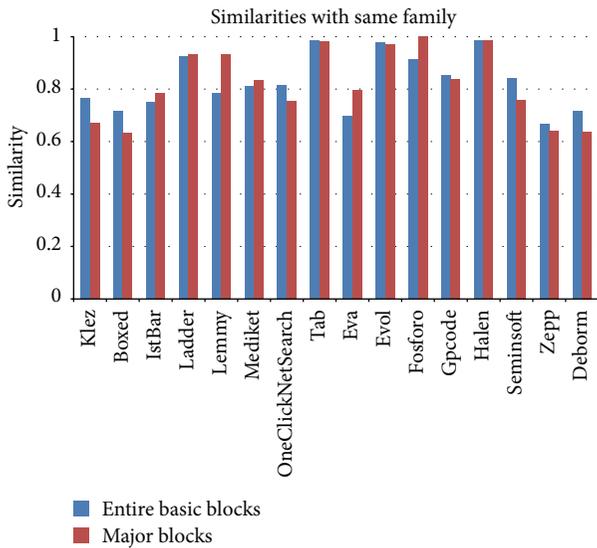


FIGURE 14: Image matrix similarity calculations of malware samples in each family following major block selection.

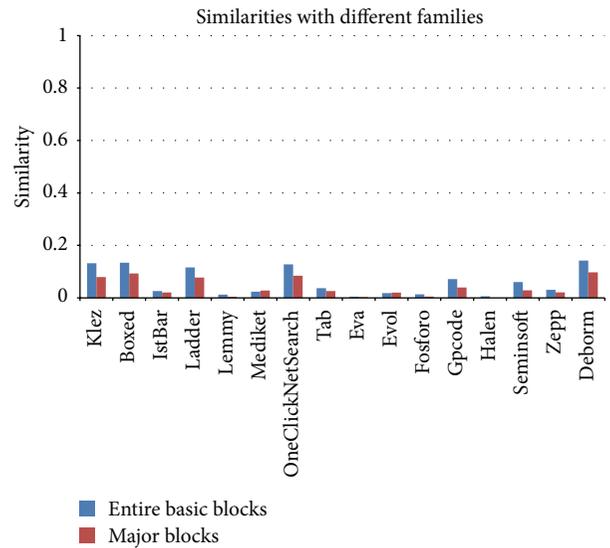


FIGURE 15: Image matrix similarity calculations of malware samples from different families following major block selection.

corresponding family in set A, with a range of similarity values among the representative image matrices from 0.181 to 0.352. Since the Agobot.02.a and the Sdbot.04.a samples are not variants of the malware families in set A, their similarity values compared to the existing individual family representative image matrices were very low.

4.2.4. Feasibility in Malware Classification. Figure 17 shows the changes in similarity values obtained by applying all the

proposed methods together, that is, the major block selection and representative image extraction techniques. Whereas the similarities between malware samples from the same families had values between 0.19 and 0.36, the similarities between malware samples from different families were less than 0.05. The classification accuracy, which was obtained by using the image matrices that were generated through the static analysis, was 0.9896. That is, only three malware samples in set A were misclassified into the other malware families.

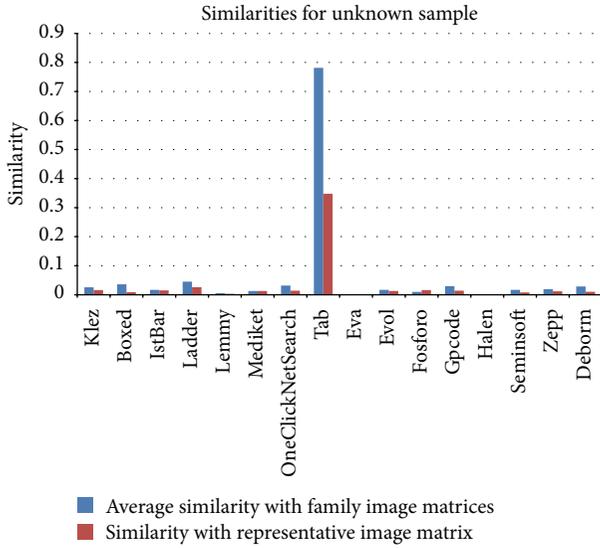


FIGURE 16: Similarity values of the unknown sample compared to the representative image matrices of individual families.

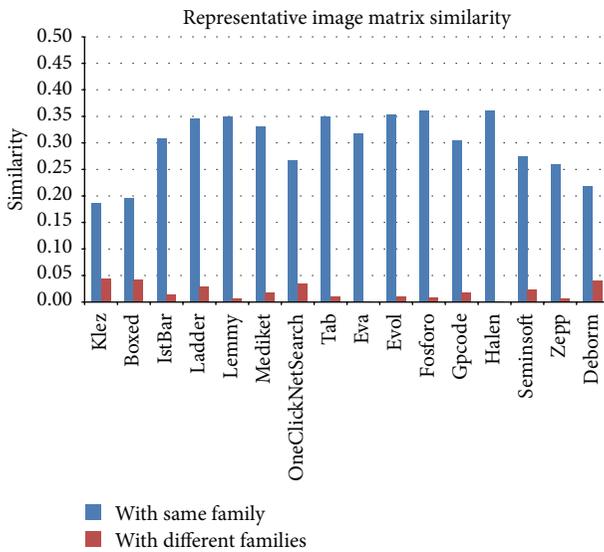


FIGURE 17: Results of similarity calculations using the three proposed techniques.

Our result was a little better than the average classification accuracy of 0.9757 using the binary texture analysis in [30]. Therefore, we conclude that our methods are feasible for malware classification because similarities within the same families will be relatively high compared to the similarities between malware samples from different families.

4.3. Execution Trace-Based Experiments. For the execution trace-based experiments, the malware samples within set B in Table 1 were executed in dynamic analysis environments using the PIN tool. Dynamic execution traces were then generated, and the repetition-filtered basic blocks were extracted from those execution traces. After filtering, the major blocks

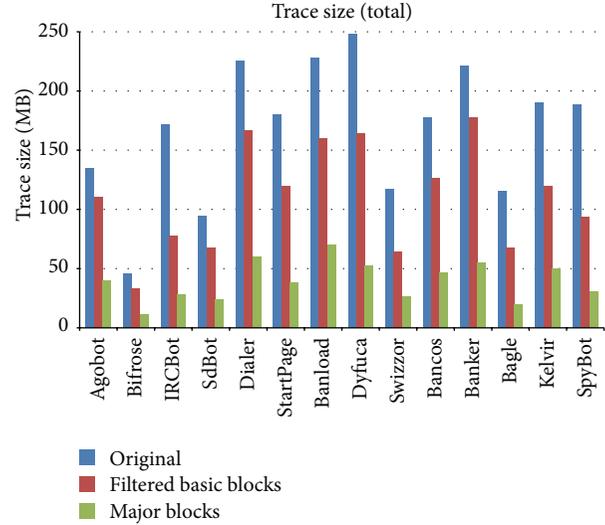


FIGURE 18: Changes in execution trace size following the application of repetition filtering and major block selection.

relating to suspicious behaviors and functions were selected. Our proposed techniques were applied to these execution traces to generate image matrices and to analyze similarities.

Figure 18 shows the decrease in size of the execution traces resulting from the application of the repetition filtering and major block selection techniques. If the sizes of the execution traces were first reduced through the repetition filtering technique and then the major block selection method was applied, the sizes of the execution traces were reduced by 76.5% on average (69.3% minimum, 83.6% maximum) compared to the original execution traces.

Figure 19 shows changes in the generated image matrices resulting from the application of the repetition filtering method and the major block selection. Decreases in the number of recorded pixels in the image matrices can be recognized when the three image matrices are compared.

Figures 20 and 21 show changes in the similarity values with the application of the repetition filtering technique and the major block selection. Although changes in the values are not large, some malware families are distinguishable if the threshold of similarity values is set properly.

In these experiments, the average similarity values of malware samples from the same families was approximately 0.65 and those from different families were approximately 0.36. Compared to the results of the static analysis described previously, the results from the execution trace-based experiments show relatively small differences. The reason for these results is that similar system dynamic link libraries (DLLs) were invoked when the malware samples of each family were executed in the dynamic analysis environment to extract the dynamic execution traces. As a result, similar opcode sequences due to the DLL calls and the executing of DLLs from the dynamic execution traces were recorded in the image matrices of individual families, so the similarity values increased. Nevertheless, the classification accuracy obtained through the similarity calculations using the image matrices

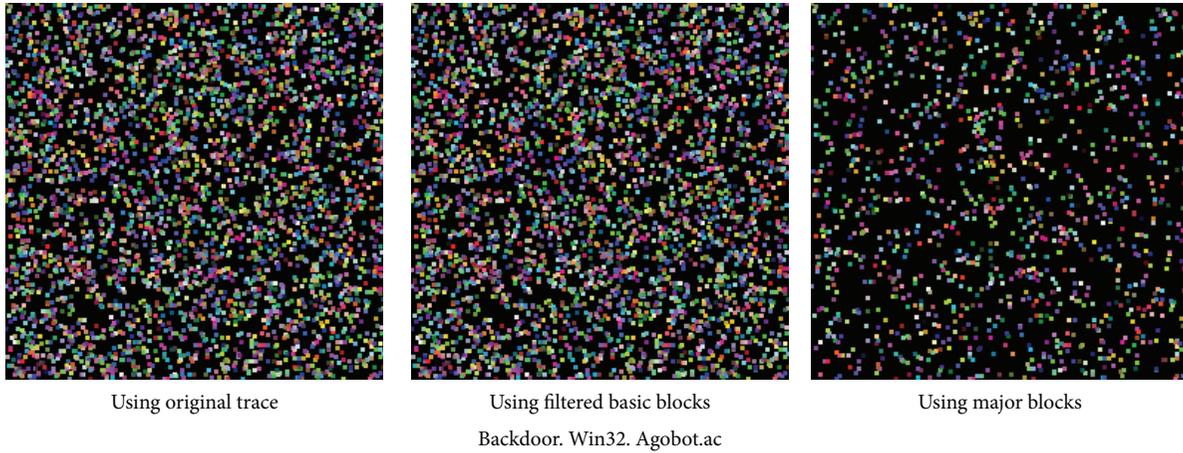


FIGURE 19: Changes in the generated image matrices from the application of repetition filtering and major block selection.

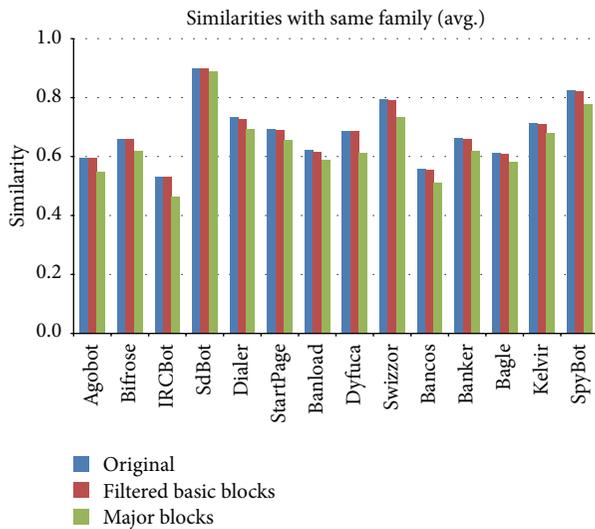


FIGURE 20: Changes in similarity between samples in the same families after repetition filtering and major block selection.

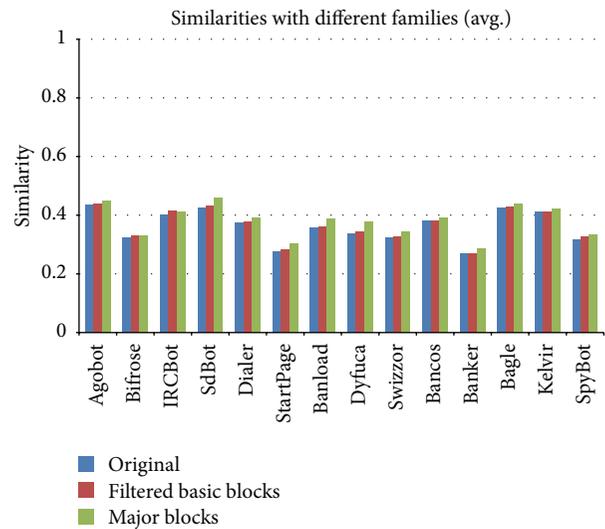


FIGURE 21: Changes in similarity between samples from different families after repetition filtering and major block selection.

that were generated based on the execution traces was 0.9732 because only 15 malware samples in set B were misclassified, and this result was similar to the accuracy in [30].

5. Conclusions and Future Work

In this paper, we proposed a novel method to analyze malware samples visually by generating image matrices. To generate the image matrices, opcode sequences were extracted through static analysis and dynamic analysis. In addition, we calculated the similarities between the malware variants using vectorized values of the RGB-colored pixels in the image matrices. The similarity calculation method using the image matrices has a faster performance than exact matching using the string type of opcode sequences or basic blocks. Our proposed method was implemented as a visual analysis tool. The experimental results showed that malware variants

included in the same family were similar when converted into image matrices, and the similarities between malware variants were shown to be higher. With our proposed method, security analysts can analyze malware samples visually and can distinguish similar malware samples for further analysis. Our future studies include faster malware detection and classification using the parallelization techniques and real-time processing based on GPGPU.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research project was supported by Ministry of Culture, Sports and Tourism (MCST) and from Korea Copyright

Commission in 2013. This paper was revised from an earlier version presented at the Research in Adaptive and Convergent Systems 2013 (RACS'13) [40].

References

- [1] M. Christodorescu and S. Jha, "Testing malware detectors," in *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA '04)*, pp. 34–44, July 2004.
- [2] B. Kang, T. Kim, H. Kwon, Y. Choi, and E. G. Im, "Malware classification method via binary content comparison," in *Proceedings of the ACM Research in Applied Computation Symposium (RACS '12)*, pp. 316–321, San Antonio, Tex, USA, October 2012.
- [3] A. Moser, C. Kruegel, and E. Kirda, "Limits of static analysis for malware detection," in *Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC '07)*, pp. 421–430, Miami Beach, Fla, USA, 2007.
- [4] M. D. Ernst, "Static and dynamic analysis: synergy and duality," in *Proceedings of the ICSE Workshop on Dynamic Analysis (WODA '03)*, pp. 24–27, Citeseer, 2003.
- [5] S. Cesare and Y. Xiang, "A fast flowgraph based classification system for packed and polymorphic malware on the endhost," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA '10)*, pp. 721–728, Perth, Australia, April 2010.
- [6] G. Bonfante, M. Kaczmarek, and J.-Y. Marion, "Architecture of a morphological malware detector," *Journal in Computer Virology*, vol. 5, pp. 263–270, 2009.
- [7] I. Briones and A. Gomez, "Graphs, entropy and grid computing: automatic comparison of malware," in *Proceedings of the Virus Bulletin Conference (VB '08)*, pp. 1–12, Ottawa, Canada, October 2008, <http://pandalabs.pandasecurity.com/blogs/images/PandaLabs/2008/10/07/IsmaelBriones-VB2008.pdf>.
- [8] S. Shang, N. Zheng, J. Xu, M. Xu, and H. Zhang, "Detecting malware variants via function-call graph similarity," in *Proceedings of the 5th International Conference on Malicious and Unwanted Software (MALWARE '10)*, pp. 113–120, Nancy, France, 2010.
- [9] J. Kinable and O. Kostakis, "Malware classification based on call graph clustering," *Journal in Computer Virology*, vol. 7, no. 4, pp. 233–245, 2011.
- [10] S. M. Tabish, M. Z. Shafiq, and M. Farooq, "Malware detection using statistical analysis of byte-level file content," in *Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics (CSI-KDD '09)*, pp. 23–31, ACM, June 2009.
- [11] B. B. Rad and M. Masrom, "Metamorphic virus variants classification using opcode frequency histogram," in *Proceedings of the 14th WSEAS International Conference on Computers*, pp. 147–155, Corfu Island, Greece, July 2010.
- [12] D. Bilar, "OpCodes as predictor for malware," *International Journal of Electronic Security and Digital Forensics*, vol. 1, pp. 156–168, 2007.
- [13] K. S. Han, S.-R. Kim, and E. G. Im, "Instruction frequency-based malware classification method," *Information*, vol. 15, no. 7, pp. 2973–2984, 2012.
- [14] I. Santos, F. Brezo, J. Nieves et al., "Idea: Opcode-sequence-based malware detection," in *Engineering Secure Software and Systems*, pp. 35–43, Springer, 2010.
- [15] A. H. Sung, J. Xu, P. Chavez, and S. Mukkamala, "Static analyzer of vicious executables (SAVE)," in *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC '04)*, pp. 326–334, December 2004.
- [16] A. Walenstein, M. Venable, M. Hayes, C. Thompson, and A. Lakhotia, "Exploiting similarity between variants to defeat malware," in *Proceedings of the BlackHat DC Conference*, 2007.
- [17] G. Chowdhury, *Introduction to Modern Information Retrieval*, Facet publishing, 2010.
- [18] M. Egele, C. Kruegel, E. Kirda, H. Yin, and D. X. Song, "Dynamic spyware analysis," in *Proceedings of the Usenix Annual Technical Conference*, pp. 233–246, 2007.
- [19] M. Fredrikson, S. Jha, M. Christodorescu, R. Sailer, and X. Yan, "Synthesizing near-optimal malware specifications from suspicious behaviors," in *Proceeding of the 31st IEEE Symposium on Security and Privacy (SP '10)*, pp. 45–60, Oakland, Calif, USA, May 2010.
- [20] P. Vinod, H. Jain, Y. K. Golecha, M. S. Gaur, and V. Laxmi, "Medusa: metamorphic malware dynamic analysis using signature from API," in *Proceedings of the 3rd International Conference on Security of Information and Networks (SIN '10)*, pp. 263–269, ACM, September 2010.
- [21] Q. G. Miao, Y. Wang, Y. Cao, X. G. Zhang, and Z. L. Liu, "APICapture—a tool for monitoring the behavior of malware," in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE '10)*, pp. V4-390–V4-394, August 2010.
- [22] K. S. Han, J. H. Lim, B. Kang, and E. G. Im, "Malware analysis using visualized images and entropy graphs," *International Journal of Information Security*, 2014.
- [23] P. Trinius, T. Holz, J. Göbel, and F. C. Freiling, "Visual analysis of malware behavior using treemaps and thread graphs," in *Proceedings of the 6th International Workshop on Visualization for Cyber Security (VizSec '09)*, pp. 33–38, Atlantic City, NJ, USA, October 2009.
- [24] J. Saxe, D. Mentis, and C. Greamo, "Visualization of shared system call sequence relationships in large malware corpora," in *Proceedings of the 9th International Symposium on Visualization for Cyber Security (VizSec '12)*, pp. 33–40, ACM, October 2012.
- [25] G. Conti, E. Dean, M. Sinda, and B. Sangster, "Visual reverse engineering of binary and data files," in *Visualization for Computer Security*, pp. 1–17, Springer, Berlin, Germany, 2008.
- [26] B. Anderson, C. Storlie, and T. Lane, "Improving malware classification: bridging the static/dynamic gap," in *Proceedings of the 5th ACM Workshop on Security and Artificial Intelligence (AISec '12)*, pp. 3–14, Raleigh, NC, USA, October 2012.
- [27] L. Nataraj, S. Karthikeyan, G. Jacob, and B. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, p. 4, ACM, 2011.
- [28] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.
- [29] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 273–280, IEEE, October 2003.
- [30] L. Nataraj, V. Yegneswaran, P. Porras, and J. Zhang, "A comparative assessment of malware classification using binary texture analysis and dynamic analysis," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence (AISec '11)*, pp. 21–30, 2011.

- [31] Y. Kang and A. Sugimoto, "Image categorization and semantic segmentation using scale-optimized textons," *Journal of IT Convergence Practice*, vol. 2, pp. 2–14, 2014.
- [32] C. Eagle, *The IDA Pro Book: The Unofficial Guide to the World's Most Popular Disassembler*, No Starch Press, 2008.
- [33] O. Yuschuk, "Ollydbg," 2007, <http://www.ollydbg.de>.
- [34] Y. Wei, Z. Zheng, and N. Ansari, "Revealing packed malware," *IEEE Security and Privacy*, vol. 6, no. 5, pp. 65–69, 2008.
- [35] P. Royal, M. Halpin, D. Dagon, R. Edmonds, and W. Lee, "PolyUnpack: automating the hidden-code extraction of unpack-executing malware," in *Proceeding of the 22nd Annual Computer Security Applications Conference (ACSAC '06)*, pp. 289–300, Miami Beach, Fla, USA, December 2006.
- [36] S. Berkowits, "Pin—A Dynamic Binary Instrumentation Tool," 2012, <https://software.intel.com/en-us/articles/pin-a-dynamic-binary-instrumentation-tool>.
- [37] B. Kang, K. S. Han, B. Kang, and E. G. Im, "Malware categorization using dynamic mnemonic frequency analysis with redundancy filtering," 2013.
- [38] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pp. 380–388, ACM, New York, NY, USA, 2002.
- [39] D. Androutsos, K. N. Plataniotis, and A. N. Venetsanopoulos, "Novel vector-based approach to color image retrieval using a vector angular-based distance measure," *Computer Vision and Image Understanding*, vol. 75, no. 1, pp. 46–58, 1999.
- [40] K. S. Han, J. H. Lim, and E. G. Im, "Malware analysis method using visualization of binary files," in *Proceedings of the 2013 Research in Adaptive and Convergent Systems*, pp. 317–321, ACM, 2013.

Research Article

Modeling the Propagation of Mobile Phone Virus under Complex Network

Wei Yang,¹ Xi-liang Wei,² Hao Guo,³ Gang An,⁴ Lei Guo,² and Yu Yao^{2,5}

¹ Software College, Northeastern University, Shenyang 110819, China

² College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

³ Quality and Technology Department, Liaoning Provincial Institute of Measurement, Shenyang 110004, China

⁴ Information Center, Panjin Finance Bureau, Panjin 124000, China

⁵ Key Laboratory of Medical Image Computing, Northeastern University, Ministry of Education, Shenyang 110819, China

Correspondence should be addressed to Wei Yang; yangwei@mail.neu.edu.cn

Received 14 March 2014; Accepted 26 June 2014; Published 15 July 2014

Academic Editor: Fei Yu

Copyright © 2014 Wei Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile phone virus is a rogue program written to propagate from one phone to another, which can take control of a mobile device by exploiting its vulnerabilities. In this paper the propagation model of mobile phone virus is tackled to understand how particular factors can affect its propagation and design effective containment strategies to suppress mobile phone virus. Two different propagation models of mobile phone viruses under the complex network are proposed in this paper. One is intended to describe the propagation of user-tricking virus, and the other is to describe the propagation of the vulnerability-exploiting virus. Based on the traditional epidemic models, the characteristics of mobile phone viruses and the network topology structure are incorporated into our models. A detailed analysis is conducted to analyze the propagation models. Through analysis, the stable infection-free equilibrium point and the stability condition are derived. Finally, considering the network topology, the numerical and simulation experiments are carried out. Results indicate that both models are correct and suitable for describing the spread of two different mobile phone viruses, respectively.

1. Introduction

More and more rogue programs called mobile phone virus, which can take control of a mobile device by exploiting its vulnerabilities, can be written to propagate from one phone to another. Security issues of mobile phones have become increasingly prominent. Though attacks from the mobile phone virus have not caused greater damage up to now, it is just a matter of time before it breaks out [1]. The large population of mobile users and the wide coverage of mobile communication network [2] create a breeding ground for the propagation of mobile phone virus. The propagation of mobile phone virus may be more potentially destructive than the computer virus. In this regard, mobile phone virus encounters a similar situation of Internet worms, so it is necessary to research the propagation behavior of mobile phone virus in the real world and design effective containment strategies to suppress them.

The usual ways for mobile phone virus to propagate include multimedia messaging service (MMS) [3] interface and e-mail services on that mobile phone. MMS messages are intended to contain media content such as photos, audios, and videos, but they can also contain infected malicious codes [4]. One noteworthy example is Commwarrior [5], which is the first mobile phone virus that can propagate via MMS. It searches for phone number through a user's local address book and sends MMS messages containing infected files to other users in the address book. It is an easy way for mobile phone virus to carry out because people are more likely to open and download the contents that they received from their friends. So the mobile phone virus could be sent out in just one click and travel to any mobiles all over the world with a larger chance of success in propagation [4].

The mobile phone virus is in the time of high-speed development. In the present, it only reproduces and propagates by tricking mobile phone users, called user-tricking

virus, but does not spread automatically. According to the development rules of the computer virus, the future trend of the mobile phone virus is that it can propagate by exploiting vulnerabilities existing in mobile phone operating systems or application software. That is, the propagation of such mobile phone virus called vulnerability-exploiting virus can be realized by itself without human participants.

The possible path through which mobile phone virus spreads depends on the social relationship of a user by exploiting the local address book or recent call records. Communications based on social network provide the environment for the spread of the mobile phone virus. And the social network will greatly influence the spread of the mobile phone virus. Recently some researchers have studied the structures of social network topologies [6–8]. They found that all of these networks are complex network and they have power-law degree distributions. Existing work on studying mobile phone virus [9, 10] does not take into account the capability of mobile phone virus to spread under complex network. Consequently this paper focuses on researching the behavior of the user-tricking virus in the present and the vulnerability-exploiting virus in the future combining the topology of the complex network.

Many particular factors can affect the propagation of mobile phone virus and its behavior is very complicated depending on the social relationship of mobile phone users. So an extremely fundamental and effective way to study the mobile phone virus is using the epidemiological models. Epidemiological models are the usual method used to understand and predict the propagation of Internet worms by many researchers [11–20].

The mobile phone virus has some commons with the Internet worms. As the behavior of mobile phone virus is more complex than that of Internet worms, it is necessary to construct a new model for virus propagation. Due to the characteristic of exponential propagation exhibited by mobile phone virus through complex network, it is challenging to model the user-tricking and vulnerability-exploiting mobile phone virus.

Through above observations, this paper models the propagation of mobile phone virus considering the characteristics of mobile phone viruses and the network topology structure. The contributions of this paper are as follows.

- (i) Two different propagation models of mobile phone viruses under the complex network are proposed in this paper. One is intended to describe the propagation of user-tricking virus, and the other is to describe the propagation of the vulnerability-exploiting virus.
- (ii) A detailed analysis is conducted to analyze the propagation models. Through analysis, the stable infection-free equilibrium point and the stability condition are derived.
- (iii) Considering the network topology, the numerical and simulation experiments are carried out. Results indicate that both models are correct and suitable for describing the spread of two different mobile phone viruses, respectively.

The rest of this paper is organized as follows. Section 2 presents related work about modeling the mobile phone virus. Section 3 conducts and analyzes the mobile phone virus SIS propagation model (M-SIS) and obtains the stability condition and the infection-free equilibrium point. Section 4 proposes the propagation model of vulnerability-exploiting mobile phone virus, which is called the mobile phone virus SIR propagation model (M-SIR). Section 5 describes the constructing process of complex network which is used to simulate the spread of the mobile phone virus. Section 6 concludes the paper and provides future research directions.

2. Related Work

Mobile phone users communicate and share files with their friends and they also take part in some activities or join groups online [21]. These characteristics give hackers the opportunities to attack mobile users. As a result, the mobile phone virus can spread quickly. More and more researchers pay attention to the area of mobile phone virus. But the research on the mobile phone virus is just in the beginning stage. Fundamental research works on it have been gradually carried out in order to raise the security awareness among users.

Leavitt lists some mobile phone viruses, such as Cabir, Skulls, and Mosquito and points out increasing virus attacks to mobile phones [1]. But he deems that a method always can be found to cope with the security issue caused by the mobile phone virus. Dwan takes the mobile phone virus “Cabir” as an example to emphasize the lack of security mechanism and expects to take certain security measures from both mobile phones’ software and hardware [22]. Jamaluddin et al. state the damage of the mobile phone virus and predict that the mobile phone virus will develop along the path of the computer virus and cause serious security problems [23]. Dagon et al. describe the security threat with which mobile users are faced and propose several security advices to mobile users [24].

With the popularity of Android platform based mobile phones, more and more attention is paid to the protection of mobile phones. Zhang et al. propose a browser-free multilevel smart phone privacy protection system by means of short message system [25]. Based on the specific network of short message, Jin et al. proposed an epidemic model of mobile phone virus based on the efficiency of immunization to reveal the spreading rule of mobile phone virus [26].

Based on the similarity between a malicious worm and a biological virus, some epidemic models representing worm propagation were presented to depict the propagation of worms, for example, SIS model (susceptible-infectious-susceptible) and SIR model (susceptible-infectious-recovered) [27]. Yao et al. research the worm propagation model by considering the time delay [28]. They found that time delay may lead to Hopf bifurcation phenomenon which will make the worm propagation system unstable and uncontrollable.

Propagation models and the stability of mobile phone virus become an attractive research field in recent years

since it facilitates worm prediction, detection, analysis and prevention, and so forth. There have been some models to simulate the mobile phone virus propagation.

Wang et al. modeled the mobility of mobile phone users in order to study the fundamental spreading patterns that characterize a mobile virus outbreak [29]. Their results explain the lack of a major mobile virus breakout so far and predict that once a mobile operating system's market share reaches the phase transition point, viruses will pose a serious threat to mobile communications. Zheng et al. analyze the communication of Bluetooth between mobile users and put forward a propagation model of the mobile phone virus which spreads through Bluetooth technique [30]. Xuetao et al. propose and evaluate a SI_1I_2S , a competition model that describes the spread of two mutually exclusive mobile viruses across heterogeneous composite networks [31].

Existing propagation models of mobile phone virus focus on the specific kind of virus. This kind of virus spreads using Bluetooth or short message, which is completely different from the virus spreading using MMS.

Mobile phone virus that spreads using MMS typically exploits the social network of users to propagate from one mobile device to another. So the topology of network is a key factor for this kind of mobile virus using MMS to propagation. As far as I know no one has studied the propagation model of this kind virus. So considering the characteristics of mobile phone virus and the social network relationship, two different propagation models of mobile phone viruses under the complex network are proposed in this paper to understand how particular factors can affect their propagation and design effective containment strategies to suppress mobile phone virus.

3. Modeling the Propagation of the User-Tricking Mobile Phone Virus

3.1. M-SIS Model. The user-tricking virus only reproduces and propagates by tricking when mobile phone users are in just one click. In this regard, the following assumption is made that the propagation path of a mobile virus can be approximated by the social network of mobile devices. Given that a user A has a higher probability to open and download a message from B with whom he periodically exchanges messages, the pair of users, A-B, would be considered more vulnerable. In contrast, if user A does not exchange messages with user C, the user A is unlikely to be infected by a mobile phone virus sent by C and hence the pair of A-C is considered less likely to be included in the propagation path of the mobile virus. This kind of virus is now prevailing on current mobile phone system and is difficult to kill completely. It will mislead users to install and then execute a norm application. Even if it removed, it can do the same thing with another guise again.

An undirected graph $G = (V, E)$ consisting of a set of vertices V and a set of edges E is used to denote mobile phone communication system. Each vertex $u \in V$ denotes a mobile in the cellular network and each edge $e(u, v)$ denotes that at least one traffic flow was exchanged between mobiles u and v . Let d_i denote the degree of any vertex $i \in V$. According to

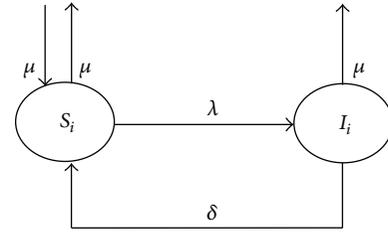


FIGURE 1: State transition graph of the i th kind of mobiles in M-SIS model.

the spread property of user-tricking mobile phone virus, the epidemic states of a mobile are divided as follows: susceptible state (S) and infectious state (I).

Susceptible state (S): nodes have not been infected by any user-tricking virus but are prone to infection.

Infectious state (I): nodes have been infected by user-tricking virus and they may infect some mobiles in state S .

Mobile users who have larger groups of friends in social network tend to appear in the contact list of many others [32]. Different nodes with different vertex's degree have different behavior to the spread of mobile phone virus. So it is necessary to study the propagation process of mobile nodes with different node's degree.

According to the nodes' degrees, these vertices in the undirected graph are classified into n kinds of nodes. The nodes with the same degree belong to a class. Let N_i denote the number of the i th kinds of nodes while the value of i ranges from 1 to n . It is assumed that there are totally N mobiles deployed in the communication network. So the sum of n kinds of mobiles is expressed as follows:

$$N = N_1 + N_2 + \dots + N_i + N_n. \quad (1)$$

Let $S_i(t)$ represent the number of the i th kinds of mobiles in the susceptible state at t time. $I_i(t)$ is defined as the number of the i th kinds of mobiles in the infected state at t time. So the number of the i th kinds of mobiles can be obtained as follows:

$$S_i(t) + I_i(t) = N_i. \quad (2)$$

In the social network, a mobile can leave or join the network randomly. So the "death" of a mobile is defined to refer to the fact that a phone drops out of the network for certain reason, such as breakdown. The "birth" means that the network adds a new mobile. But it is assumed that the system is a closed system and the number of "dead" rates of mobile is the same as that of the "birth" one.

The propagation modeling of user-tricking mobile phone virus under complex network called M-SIS model is proposed, which means mobile phone virus SIS (susceptible-infectious-susceptible) propagation model. In the M-SIS model, M represents mobile phone, and S stands for the susceptible state while I stands for the infectious state. The state transforming process of any kind of mobiles in M-SIS model is illustrated in Figure 1.

A node may change its states as follows.

Node i of any kind can transit to the infectious state if it is at the susceptible state. The infection probability, also called contact infection rate, is presented by λ .

TABLE 1: Parameters description in the M-SIS model.

Parameter	Meaning
N	The total number of mobiles
$S_i(t)$	The number of the i th kinds of mobiles in the susceptible state at time t
$I_i(t)$	The number of the i th kinds of mobiles in the infectious state at time t
N_i	The number of the i th kinds of mobiles at time t
λ	The contact infection rate
δ	The infection recovery rate
μ	The "birth/death" rate

A mobile is not permanently immune against the virus and has a risk of reinfection. So a mobile at the infectious state can kill the virus and recover to the susceptible state. The infection recovery rate is presented by δ .

To maintain the balance of the network system, the "death" rate and the "birth" rate are all μ . The "new born" mobiles are all in the susceptible state.

The description of related parameters in M-SIS model is showed in Table 1.

Based on the above analysis and compartment model of i th kind presented in Figure 1, given a topology of a mobile communication network, the number of susceptible and infected nodes of the i th kind at time t in the M-SIS model can be formulated by the equations as follows:

$$\begin{aligned} \frac{dS_i(t)}{dt} &= \mu N_i - \mu S_i(t) - \lambda k_i S_i(t) \Theta(t) + \delta I_i(t), \\ \frac{dI_i(t)}{dt} &= \lambda k_i S_i(t) \Theta(t) - \mu I_i(t) - \delta I_i(t). \end{aligned} \quad (3)$$

In (3), k_i is the degree of the i th kind of mobile phone nodes, where $i = 1, 2, \dots, n$. $\Theta(t)$ is the infected probability that any of neighbor nodes of one mobile phone node and the expression of $\Theta(t)$ are as

$$\Theta(t) = \frac{\sum_{i=1}^n k_i P(k_i) I_i(t)}{\langle k \rangle}. \quad (4)$$

In (4), $\langle k \rangle$ means the average degree of nodes in the network, which can be expressed as

$$\langle k \rangle = \sum k_i P(k_i), \quad (5)$$

where $P(k_i)$ is the probability density of nodes with the degree k_i . So the differential equations of the M-SIS model can be

concluded as the following equation:

$$\begin{aligned} \frac{dS(t)}{dt} &= \sum_{i=1}^n \frac{dS_i(t)}{dt}, \\ \frac{dI(t)}{dt} &= \sum_{i=1}^n \frac{dI_i(t)}{dt}. \end{aligned} \quad (6)$$

3.2. Infection-Free Equilibrium Point. The infection-free equilibrium refers to the fact that the mobile virus gets removed and the number of infected mobiles remains 0. To derive the infection-free equilibrium point, let both $dS_i(t)/dt$ and $dI_i(t)/dt$ be equal to 0, and the following expression is obtained as

$$\begin{aligned} \mu N_i - \mu S_i(t) - \lambda k_i S_i(t) \Theta(t) + \delta I_i(t) &= 0, \\ \lambda k_i S_i(t) \Theta(t) - \mu I_i(t) - \delta I_i(t) &= 0. \end{aligned} \quad (7)$$

When $I_i = 0$, $S_i(t)$ and $I_i(t)$ can be calculated as follows, where $i = 1, 2, \dots, n$:

$$\begin{aligned} S_i(t) &= N_i, \\ I_i(t) &= 0. \end{aligned} \quad (8)$$

The number of the i th kinds of mobile phones in the susceptible state is N_i , while that of the i th kinds of mobile phones in the infectious state is 0. The infection-free equilibrium point of the mobile phone virus propagation system under the M-SIS model is thus $E_0^*(N_1, 0, N_2, 0, \dots, N_n, 0)$.

3.3. Stability of the Infection-Free Equilibrium. Though the user-tricking virus is difficult to completely kill and mobiles are not permanently immune, it is ensured that the number of infected mobiles can dynamically remain 0. It means that the infection-free equilibrium can be achieved. Its stability for the propagation system of the mobile phone virus will be discussed.

Theorem 1. *If the basic reproduction number $R_0 < 1$, the propagation system under the M-SIS model of mobile phone virus will stabilize at the infection-free equilibrium point:*

$$R_0 = \frac{\lambda}{(\mu + \delta) N \langle k \rangle} \sum_{i=1}^n k_i^2 N_i^2. \quad (9)$$

Proof. Let $i = 1$ and put it into (3); the following equation can be obtained:

$$\begin{aligned} \frac{dS_1(t)}{dt} &= \mu N_1 - \mu S_1(t) - \lambda k_1 S_1(t) \Theta(t) + \delta I_1(t), \\ \frac{dI_1(t)}{dt} &= \lambda k_1 S_1(t) \Theta(t) - \mu I_1(t) - \delta I_1(t). \end{aligned} \quad (10)$$

Two equations from (10) are given the partial derivative with the aspects of $S_1, I_1, S_2, I_2, \dots, S_n, I_n$ and then set $I_i = 0$. $A_{2 \times 2n}$ dimensional matrix is obtained, where $g(j) = k_j P(k_j) / \langle k \rangle$, $j = 1, 2, \dots, n$:

$$\begin{pmatrix} -\mu & -\lambda k_1 S_1 g(1) & \cdots & 0 & -\lambda k_1 S_1 g(j) & \cdots & 0 & -\lambda k_1 S_1 g(n) \\ 0 & -(\mu + \delta) + \lambda k_1 S_1 g(1) & \cdots & 0 & \lambda k_1 S_1 g(j) & \cdots & 0 & \lambda k_1 S_1 g(n) \end{pmatrix}. \tag{11}$$

Similarly, when $i = 2, 3, \dots, n$, we take the derivative of formula (3) with the aspects of $S_1, I_1, \dots, S_n, I_n$ and then set

$I_i = 0$. With matrix (11), a $2n \times 2n$ dimensional matrix is obtained:

$$\begin{pmatrix} -\mu & -\lambda k_1 S_1 g(1) & \cdots & 0 & -\lambda k_1 S_1 g(j) & \cdots & 0 & -\lambda k_1 S_1 g(n) \\ 0 & -(\mu + \delta) + \lambda k_1 S_1 g(1) & \cdots & 0 & \lambda k_1 S_1 g(j) & \cdots & 0 & \lambda k_1 S_1 g(n) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 0 & -\lambda k_j S_j g(1) & \cdots & -\mu & -\lambda k_j S_j g(j) & \cdots & 0 & -\lambda k_j S_j g(n) \\ 0 & \lambda k_j S_j g(1) & \cdots & 0 & -(\mu + \delta) + \lambda k_j S_j g(j) & \cdots & 0 & \lambda k_j S_j g(n) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 0 & -\lambda k_n S_n g(1) & \cdots & 0 & -\lambda k_n S_n g(j) & -\mu & -\lambda k_n S_n g(n) \\ 0 & \lambda k_n S_n g(1) & \cdots & 0 & \lambda k_n S_n g(j) & 0 & -(\mu + \delta) + \lambda k_n S_n g(n) \end{pmatrix}. \tag{12}$$

According to [33], n eigen values of matrix (12) are all equal to $-\mu$. Lines or columns including any of these n

eigen values are removed, and a $n \times n$ dimensional matrix is obtained:

$$\begin{pmatrix} -(\mu + \delta) + \lambda k_1 S_1 g(1) & \lambda k_1 S_1 g(2) & \cdots & \lambda k_1 S_1 g(j) & \cdots & \lambda k_1 S_1 g(n) \\ \lambda k_2 S_2 g(1) & -(\mu + \delta) + \lambda k_2 S_2 g(2) & & \lambda k_2 S_2 g(j) & & \lambda k_2 S_2 g(n) \\ \vdots & \vdots & & \vdots & & \vdots \\ \lambda k_j S_j g(1) & \lambda k_j S_j g(2) & & -(\mu + \delta) + \lambda k_j S_j g(j) & & \lambda k_j S_j g(n) \\ \vdots & \vdots & & \vdots & & \vdots \\ \lambda k_n S_n g(1) & \lambda k_n S_n g(2) & & \lambda k_n S_n g(j) & & -(\mu + \delta) + \lambda k_n S_n g(n) \end{pmatrix}. \tag{13}$$

A series of transformations for matrix (13) are performed, and then the following matrix is given:

$$\begin{pmatrix} -(\mu + \delta) & 0 & \cdots & 0 & \cdots & \lambda k_1 S_1 g(n) \\ 0 & -(\mu + \delta) & \cdots & 0 & \cdots & \lambda (k_1 S_1 g(1) + k_2 S_2 g(2)) \frac{g(n)}{g(2)} \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & -(\mu + \delta) & \cdots & \lambda \sum_{i=1}^j [k_i S_i g(i)] \frac{g(n)}{g(j)} \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & -(\mu + \delta) + \lambda \sum_{i=1}^n [k_i S_i g(i)] \end{pmatrix}. \tag{14}$$

Obviously, the matrix (14) has an upper triangular one, and its characteristic equation is as follows:

$$[\lambda + (\mu + \delta)]^{n-1} \cdot \left[\lambda + (\mu + \delta) - \lambda \sum_{i=1}^n [k_i S_i g(i)] \right] = 0. \quad (15)$$

From (15), the characteristic values are obtained:

$$\lambda_1 = -(\mu + \delta), \quad \lambda_2 = -(\mu + \delta) + \lambda \sum_{i=1}^n [k_i S_i g(i)]. \quad (16)$$

According to Routh-Hurwitz criterion, if and only if all of characteristic values are less than zero, the propagation system will eventually be stable at the equilibrium point E_0 . Obviously, λ_1 are negative and the stability relies on λ_2 . If λ_2 is less than 0, the equilibrium will be achieved. By transformation, the stability condition is derived as

$$R_0 = \frac{\lambda}{(\mu + \delta) N \langle k \rangle} \sum_{i=1}^n k_i^2 N_i^2 < 1. \quad (17)$$

The proof is complete. \square

Corollary 2. *When the degree of a mobile node grows, the basic reproduction number R_0 gets increased, which means that it increases difficulty in realizing the stability for the propagation system of the mobile phone virus.*

Proof. Equation (17) can be converted into the following inequality:

$$R_0 = \frac{\lambda}{(\mu + \delta) N \langle k \rangle} \cdot \frac{\sum_{i=1}^n k_i^2 N_i^2}{\sum_{i=1}^n k_i N_i} < 1. \quad (18)$$

Obviously, $\sum_{i=1}^n k_i^2 N_i^2 / \sum_{i=1}^n k_i N_i$ is a monotonic function of k_i . When the degree k_i of the mobile phone node is increased, R_0 will also grow. It makes (18) more difficult to be satisfied. Corollary 2 is thus drawn. \square

4. Modeling the Propagation of the Vulnerability-Exploiting Mobile Phone Virus

4.1. M-SIR Model. According to the development of virus, the mobile virus will eventually become a mobile worm which is called vulnerability-exploiting virus. The vulnerability-exploiting virus will automatically propagate by exploiting vulnerabilities existing in mobile phone operating systems or application software. Patching can be applied to repair vulnerabilities and then protect mobile phones from attacks. According to the spread property of vulnerability-exploiting virus, the epidemic state of a node is divided as follows: susceptible state (S), infectious state (I), and recovered state (R).

Susceptible state (S): nodes have not been infected by any user-tricking virus but are prone to infection. Infectious

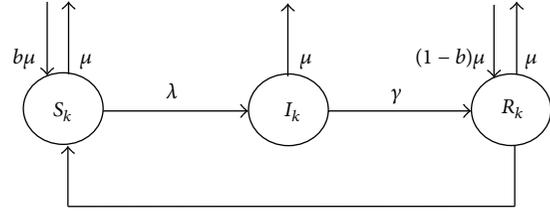


FIGURE 2: State transition graph of the k th kind of nodes in M-SIR model.

TABLE 2: Notations in the M-SIR model.

Parameter	Meaning
N	The total number of mobiles
$S_i(t)$	The number of the i th kinds of mobiles in the susceptible state at time t
$I_i(t)$	The number of the i th kinds of mobiles in the infected state at time t
$R_i(t)$	The number of the i th kinds of mobiles in the "immune" state at time t
N_i	The number of the i th kinds of mobiles at time t
λ	The contact infection rate
γ	The "immune" rate for infected mobiles
ω	The "immune" rate for susceptible mobiles
μ	The "birth/death" rate
b	The probability of new "born" susceptible mobiles
$1 - b$	The probability of new "born" immune mobiles

state (I): nodes have been infected by vulnerability-exploiting virus and they may infect some nodes in state S. Recovered state (R): nodes are cleaned of vulnerability-exploiting virus and immune to the same type of cleaned virus.

The propagation modeling of vulnerability-exploiting mobile phone virus under complex network called M-SIR model is proposed, which means mobile phone virus SIR (susceptible-infectious-recovered) propagation model. The state transforming process of any kind of nodes in M-SIR model is illustrated in Figure 2.

In the M-SIR model, a node in the k th kind can transit to the infectious state if it is at the susceptible state. The infection probability is presented by λ . The infectious node can clean the virus through patching with the immune rate γ . Once patched, the mobile is immune to the virus permanently. The susceptible node can also be patched in advance of infection with patching rate ω and transits to the recovered state. To maintain the balance of the network system, the "death" rate and the "birth" rate are all μ . The "new born" mobiles are all in the susceptible state. But the "new born" mobiles become not only susceptible ones but also "immune" ones, because new mobiles may install new versions of software with patches. The description of related parameters in M-SIR model is shown in Table 2.

Define $R_i(t)$ as the number of the i th kinds of immune mobiles at time t . A mobile can be in one of three states for a time, and the sum for three classes of mobiles is as

$$S_i(t) + I_i(t) + R_i(t) = N_i. \tag{19}$$

According to the above analysis and state transition graph in Figure 2, given a topology of a social network, the number of susceptible, infected, and recovered nodes of the i th kind at time t in the M-SIR model can be presented by

$$\begin{aligned} \frac{dS_i(t)}{dt} &= b\mu N_i - \mu S_i(t) - \lambda k_i S_i(t) \Theta(t) - \omega S_i(t), \\ \frac{dI_i(t)}{dt} &= \lambda k_i S_i(t) \Theta(t) - \mu I_i(t) - \gamma I_i(t), \\ \frac{dR_i(t)}{dt} &= (1 - b) \mu N_i + \gamma I_i(t) - \mu R_i(t). \end{aligned} \tag{20}$$

There are n kinds of nodes in the network, so the differential equations of the M-SIR model can be concluded as the following equation:

$$\begin{aligned} \frac{dS(t)}{dt} &= \sum_{i=1}^n \frac{dS_i(t)}{dt}, \\ \frac{dI(t)}{dt} &= \sum_{i=1}^n \frac{dI_i(t)}{dt}, \\ \frac{dR(t)}{dt} &= \sum_{i=1}^n \frac{dR_i(t)}{dt}. \end{aligned} \tag{21}$$

4.2. Infection-Free Equilibrium Point. In order to obtain the infection-free equilibrium point, (20) is converted into the following equation:

$$\begin{aligned} b\mu N_i - \mu S_i(t) - \lambda k_i S_i(t) \Theta(t) - \omega S_i(t) &= 0, \\ \lambda k_i S_i(t) \Theta(t) - \mu I_i(t) - \gamma I_i(t) &= 0, \\ (1 - b) \mu N_i + \gamma I_i(t) - \mu R_i(t) &= 0. \end{aligned} \tag{22}$$

Solving (22), $S_i(t)$, $I_i(t)$, and $R_i(t)$ are derived as follows:

$$\begin{aligned} S_i &= \frac{b\mu}{\mu + \omega} N_i, \\ I_i &= 0, \\ R_i &= N_i - \frac{b\mu}{\mu + \omega}. \end{aligned} \tag{23}$$

There are total n kinds of nodes. Each kind of nodes has an infection-free equilibrium point. So the infection-free equilibrium point of the mobile phone virus propagation system under the M-SIR model is $E_1^*(S_1, 0, R_1, S_2, 0, R_2, \dots, S_n, 0, R_n)$, where

$$S_i = \frac{b\mu}{\mu + \omega} N_i, \quad I_i = 0, \quad R_i = N_i - \frac{b\mu}{\mu + \omega}. \tag{24}$$

4.3. Stability of the Infection-Free Equilibrium

Theorem 3. If the basic reproduction number $R_1 < 1$, the mobile phone virus propagation system under the M-SIR model will stabilize at the infection-free equilibrium point:

$$R_1 = \frac{\lambda b \mu}{(\mu + \gamma)(\mu + \omega) N \langle k \rangle} \cdot \sum_{i=1}^n i^2 N_i^2. \tag{25}$$

Proof. Take the partial derivative of three equations to the right in (20) with the aspects of $S_1, I_1, S_2, I_2, \dots, S_n, I_n$. With $I_i = 0$, a $2n \times 2n$ dimensional matrix is given:

$$\begin{pmatrix} -\mu - \omega & -\lambda k_1 S_1 g(1) & \cdots & 0 & -\lambda k_1 S_1 g(j) & \cdots & 0 & -\lambda k_1 S_1 g(n) \\ 0 & -(\mu + \gamma) + \lambda k_1 S_1 g(1) & \cdots & 0 & \lambda k_1 S_1 g(j) & \cdots & 0 & \lambda k_1 S_1 g(n) \\ \vdots & \vdots \\ 0 & -\lambda k_j S_j g(1) & \cdots & -\mu - \omega & -\lambda k_j S_j g(j) & \cdots & 0 & -\lambda k_j S_j g(n) \\ 0 & \lambda k_j S_j g(1) & \cdots & 0 & -(\mu + \gamma) + \lambda k_j S_j g(j) & \cdots & 0 & \lambda k_j S_j g(n) \\ \vdots & \vdots \\ 0 & -\lambda k_n S_n g(1) & \cdots & 0 & -\lambda k_n S_n g(j) & \cdots & -\mu - \omega & -\lambda k_n S_n g(n) \\ 0 & \lambda k_n S_n g(1) & \cdots & 0 & \lambda k_n S_n g(j) & \cdots & 0 & -(\mu + \gamma) + \lambda k_n S_n g(n) \end{pmatrix}, \tag{26}$$

where $g(j) = k_j P(k_j) / \langle k \rangle$, $j = 1, 2, \dots, n$.

Removing the lines and columns including $-\mu + \omega$, a matrix of $n \times n$ dimensional is given as follows:

$$\begin{pmatrix} -(\mu + \gamma) + \lambda k_1 S_1 g(1) & \lambda k_1 S_1 g(2) & \dots & \lambda k_1 S_1 g(j) & \dots & \lambda k_1 S_1 g(n) \\ \lambda k_2 S_2 g(1) & -(\mu + \gamma) + \lambda k_2 S_2 g(2) & & \lambda k_2 S_2 g(j) & & \lambda k_2 S_2 g(n) \\ \vdots & \vdots & & \vdots & & \vdots \\ \lambda k_j S_j g(1) & \lambda k_j S_j g(2) & & -(\mu + \gamma) + \lambda k_j S_j g(j) & & \lambda k_j S_j g(n) \\ \vdots & \vdots & & \vdots & & \vdots \\ \lambda k_n S_n g(1) & \lambda k_n S_n g(2) & & \lambda k_n S_n g(j) & & -(\mu + \gamma) + \lambda k_n S_n g(n) \end{pmatrix}. \tag{27}$$

The second column of the matrix (26) multiplying by $-g(1)/g(2)$ is added to the first column, and then the third column multiplying by $-g(2)/g(3)$ is added to the second column and so on. After that, the first row multiplying by

$g(1)/g(2)$ is added to the second row, and then the second row multiplying by $g(2)/g(3)$ is added to the third row and so on. The following matrix is thus obtained:

$$\begin{pmatrix} -(\mu + \gamma) & 0 & \dots & 0 & \dots & \lambda k_1 S_1 g(n) \\ 0 & -(\mu + \gamma) & & 0 & & \lambda [k_1 S_1 g(1) + 2k_2 S_2 g(2)] \frac{g(n)}{g(2)} \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & & -(\mu + \gamma) & & \lambda \sum_{i=1}^j [k_i S_i g(i)] \frac{g(n)}{g(j)} \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & & 0 & & -(\mu + \gamma) + \lambda \sum_{i=1}^n [k_i S_i g(i)] \end{pmatrix}. \tag{28}$$

The characteristic equation of (28) is showed as follows:

$$[\lambda + (\mu + \delta)]^{n-1} \cdot \left[\lambda + (\mu + \delta) - \lambda \sum_{i=1}^n [k_i S_i g(i)] \right] = 0. \tag{29}$$

The characteristic values are as follows:

$$\lambda_1 = -(\mu + \delta), \quad \lambda_2 = -(\mu + \delta) + \lambda \sum_{i=1}^n [k_i S_i g(i)]. \tag{30}$$

According to Routh-Hurwitz criterion, if and only if all of the characteristic values are less than zero, the propagation system will eventually be stable at the equilibrium point E_1^* . By transformation of λ_2 , the stability condition is obtained as

$$R_1 = \frac{\lambda \sum_{i=1}^n [i S_i g(i)]}{\mu + \gamma} = \frac{\lambda b \mu}{(\mu + \gamma)(\mu + \omega) N \langle k \rangle} \cdot \sum_{i=1}^n i^2 N_i^2 < 1. \tag{31}$$

The proof is complete. □

5. Constructing the Network Topology

The attacks target of the mobile virus is to infect the smart phone. The propagation path of mobile virus obeys the mobile user's social network, which has its own characters and greatly affects the propagation of the mobile phone virus. Thus it is indispensable to construct such a network to simulate the propagation of the mobile phone virus and validate our models.

The social network which is the propagation environment of mobile virus is a typical complex network. In the real world lots of networks have been proved to be complex network such as World Wide Web and email. The complex network has the following two characteristics: the degree of a node follows the power-law distribution and the network appears as small-world phenomenon. It is hard to put the real mobile virus into the real mobile network. So network topology generator called Inet3.0 is used to create a complex network to simulate the environment of mobile virus.

Inet is a topology generator developed by the University of Michigan and its current version has been upgraded to 3.0. When giving the total number of N nodes, Inet3.0 could output the information of N nodes including the position, degree, and the neighbors. Inet3.0 simulates the topology

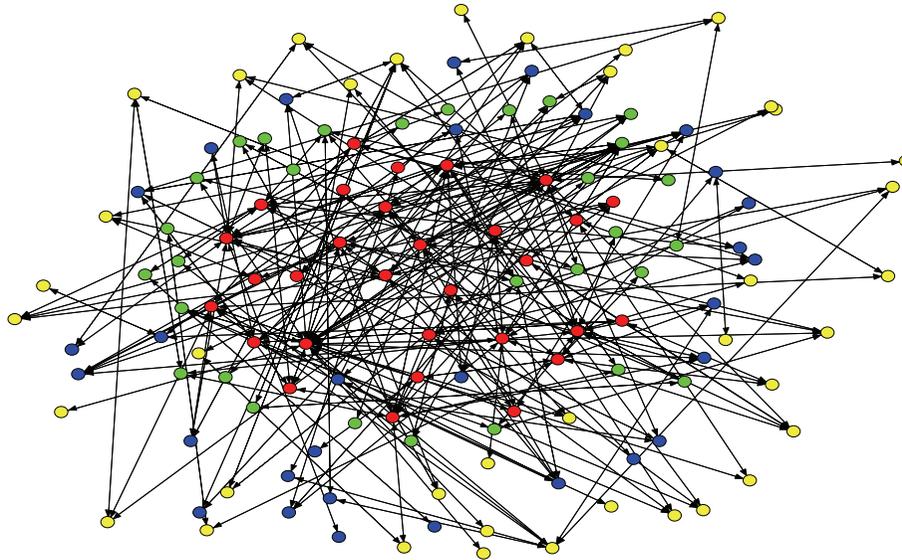


FIGURE 3: The distribution and connectivity of 130 nodes in the network generated by Inet3.0.

structure of the Internet and it accords with the characteristics of the complex network. Firstly, nodes' degrees generated by Inet3.0 follow the power-law distribution. Secondly, the characteristic path length created by Inet3.0 is short, which reflects the effect of the small-world phenomenon of social network. However, the clustering coefficient of the network built by Inet3.0 is relatively large. The network generated by Inet3.0 is much closed to the complex network and can be applied for simulating the propagation of the mobile phone virus.

In this paper, Inet3.0 is used to build a complex network which contains 10000 nodes. There are 118 different kinds of degrees among which the biggest value is 1799 and the least one is 1. Due to the high density of the topology and the page limit, it is difficult to differentiate the connectivity between nodes. Figure 3 shows the distribution and the connectivity of only 130 nodes in the topology structure, and the degrees of them are the biggest of all 10000 nodes.

Among the 130 nodes, the 30 red nodes are those with the biggest degrees; the 30 green ones are those with bigger degrees; the 30 blue ones are those with smaller degrees; the 40 yellow ones are those with the smallest degrees.

6. Numerical and Simulation Experiments

To verify the accuracy of theoretical analysis and the correctness of both M-SIS and M-SIR models, the numerical and simulation experiments are separately carried out. Numerical experiments are based on iterations of formulae and can directly reflect the property of the models. It is hard to simulate the real propagation environment of mobile phones virus. So the simulation experiments are carried out like other researchers [22–30]. Our simulation is a discrete-time simulation and well embodies the propagation of viruses in which node data are obtained on a time interval every second.

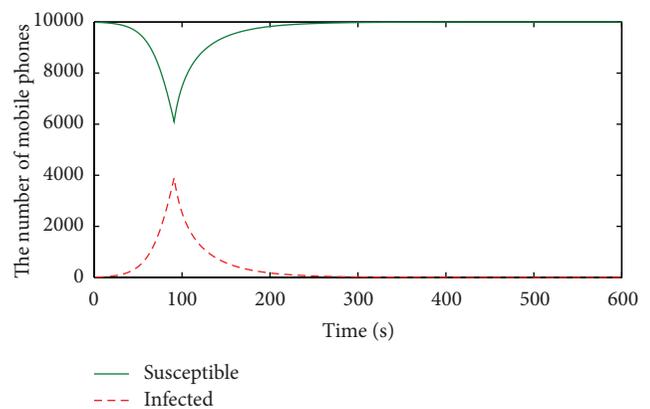


FIGURE 4: Numerical results in the M-SIS model.

Different from numerical experiments, the simulation imitates the real environment and is more closed to reality.

To raise the accuracy, the experiments under the same condition are carried out for 100 times, and the experiment result is derived from the average of 100 results. Algorithm 1 is the algorithm of the simulation which embodies the topology of the network. It is noted that one susceptible mobile can only be infected by its neighboring infected mobiles. The two-dimensional array `Link_Matrix [][]` is used to store the joined relationship between nodes.

6.1. Experiment for the M-SIS Model. The parameters in our experiments are chosen based on the research results of Zou et al. [34] and Wang et al. [35]. Due to the limit of computer memory and Inet3.0, 10,000 mobile phone nodes are set in our network system.

The contact infection rate λ of the mobile phone virus is set at 0.00003 with the same magnitude of the initial infection rate in Zou et al.'s research [34]. Similarly, the death/birth rate

```

for i = 0 .. virus_NUM-1
  if (node[i].status is susceptible)
    for j = 0 .. virus_NUM-1
      if (Link_Matrix[i][j] == 1 and node[j].status is infected)
        if (random number (rn) match the infection rate  $\lambda$ )
          node[i].status become infected
          break
        end if
      end if
    end for
    if (j > virus_NUM-1 and rn match the "immune" rate  $\omega$ )
      node[i].status become immune
    else if (j > virus_NUM-1 and rn match the "death" rate  $\mu$ )
      node[i].status become dead
    end if
    else if (node[i].status is infected)
      if (rn match the "immune" rate  $\gamma$ )
        node[i].status become immune
      else if (rn match the "death" rate  $\mu$ )
        node[i].status become dead
      end if
    else if (node[i].status is immunized)
      if (rn match the "death" rate  $\gamma$ )
        node[i].status become dead
      end if
    end if
  end for
  for i = 0 .. virus_NUM-1
    if (node[i].status is dead and rn match the probability  $b$ )
      node[i].status become susceptible
    else
      node[i].status become immune
    end if
  end for
end for

```

ALGORITHM 1

μ is assigned to be 0.00002 based on Wang et al.'s study [35]. The recovery rate δ is assumed to set 0.1. At the beginning, the mobile phone virus spreads along the edges of mobile phone nodes which own few contacts with others and then attacks core nodes. Therefore, there are 10 infected mobile phones with the degree of 1 initially, which means that the initial infected nodes only have one contact with other nodes.

The numerical results of the number of susceptible, infected mobile phones over time in M-SIS model are showed, respectively, in Figure 4.

To observe the propagation of the mobile user-tricking virus, virus-killing measure is taken after the 90 s, and sharp points appear in the curves at 90 s.

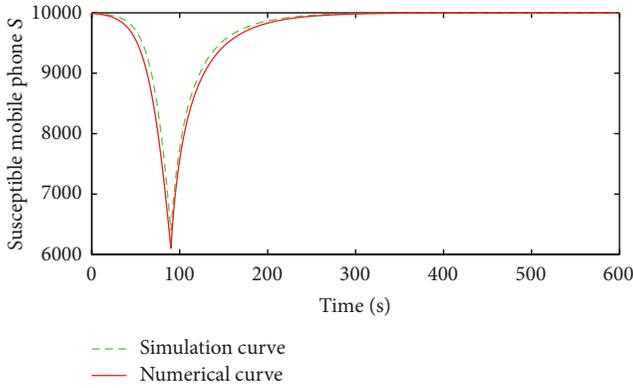
According to Theorem 1, the basic reproduction number R_0 is about 0.8 with the above parameters. It means that the propagation system of the mobile user-tricking virus under the M-SIS model will be eventually stable at its infection-free equilibrium point. Obviously, the number of infected mobile phones shrinks to 0 and that of susceptible ones is up to 10000 in Figure 4, which indicates that the infection-free equilibrium is achieved. The accuracy of theoretical analysis gets verified.

To check the correctness of M-SIS propagation model, the simulation experiments have been executed and the simulation results are compared with numerical results under the same parameters as shown in Figure 5.

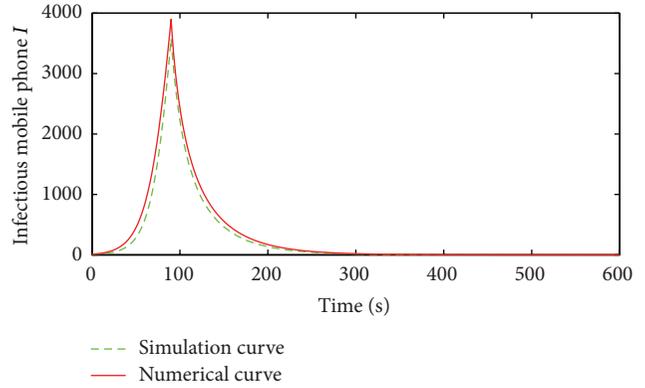
Figure 5(a) compares the number of susceptible mobiles in numerical and simulation result and Figure 5(b) compares the number of susceptible mobiles. It is seen that the numerical curves match the simulation ones very well, which verifies the correctness of the M-SIS propagation model.

The affections of different parameters on the mobile user-tricking virus propagation model are tested. The contact infection rate λ is firstly discussed. λ is specified as 0.00002, 0.00003, 0.00004, 0.00005, 0.00006, and 0.00007, respectively, and other parameters remain unchanged. With different contact infection rates, the propagation trends of the mobile user-tricking virus are showed in Figure 6.

The increase of the contact infection rate can fasten the propagation of mobile user-tricking virus. But when it increases to certain extent, the impact of the mobile user-tricking virus goes down. The larger contact infection rate is, the more nodes are infected. So the contact infection rates can



(a) Compare the number of susceptible mobiles in numerical and simulation result



(b) Compare the number of infectious mobiles in numerical and simulation result

FIGURE 5: Comparison of numerical and simulation result.

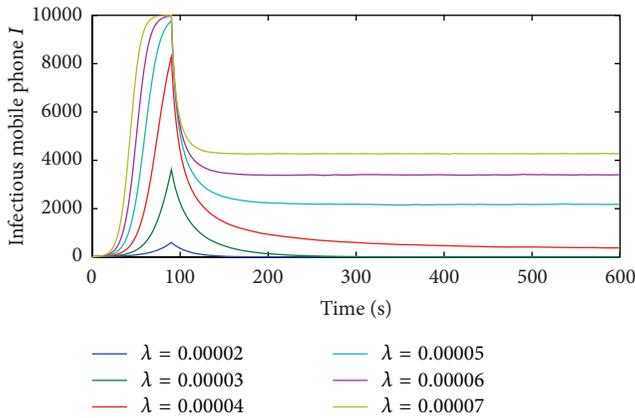


FIGURE 6: Impact of the contact infection rate λ on the M-SIS model.

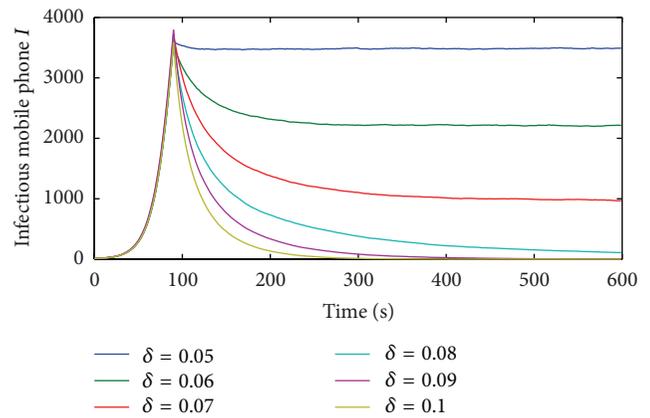


FIGURE 7: Impact of infection recovery rate δ on the M-SIS model.

rapid the propagation speed and wide the propagation scope of mobile user-tricking virus.

The infection recovery rate δ is also discussed while other parameters remain unchanged. δ is set at 0.05, 0.06, 0.07, 0.08, 0.09, and 0.1 respectively, and the propagations trends of the mobile user-tricking virus are given in Figure 7.

With the increase of δ , the number of infected nodes decreases, but all the curves reach the peak at the same time. It means that the infection recovery rate can only affect the spread scope of mobile user-tricking virus. It cannot rapid the propagation speed.

6.2. Experiment for the M-SIR Model. In this experiment, the patching rate γ for infected mobiles is 0.01 based on the research of Wang et al. [35]. The root of mobile vulnerability-exploiting virus existing is software vulnerabilities which are inevitable during the design and implementation process of software and hard to detect. Due to lots of bandwidth consumption the patch cannot be distributed in time, so the patching rate ω for susceptible mobile phones is relatively small and is set as 0.0001. And it is assumed that the probability b that the “new born” mobile phone becomes

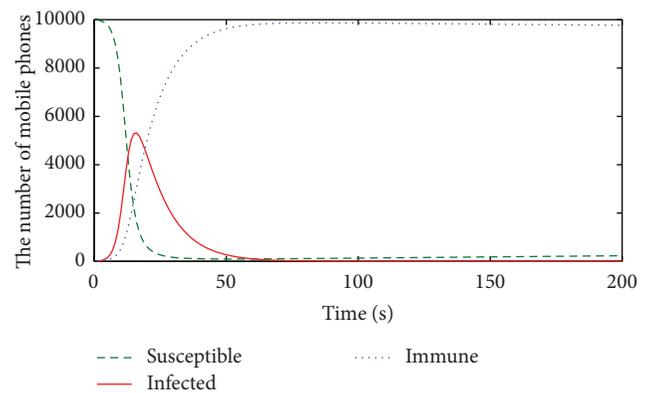


FIGURE 8: Numerical results in the M-SIR model.

susceptible one is 0.6. Other parameters are set the same as the ones in the M-SIS model.

The numerical results of the susceptible, infected, and immune mobile phones in the M-SIR model are given in Figure 8.

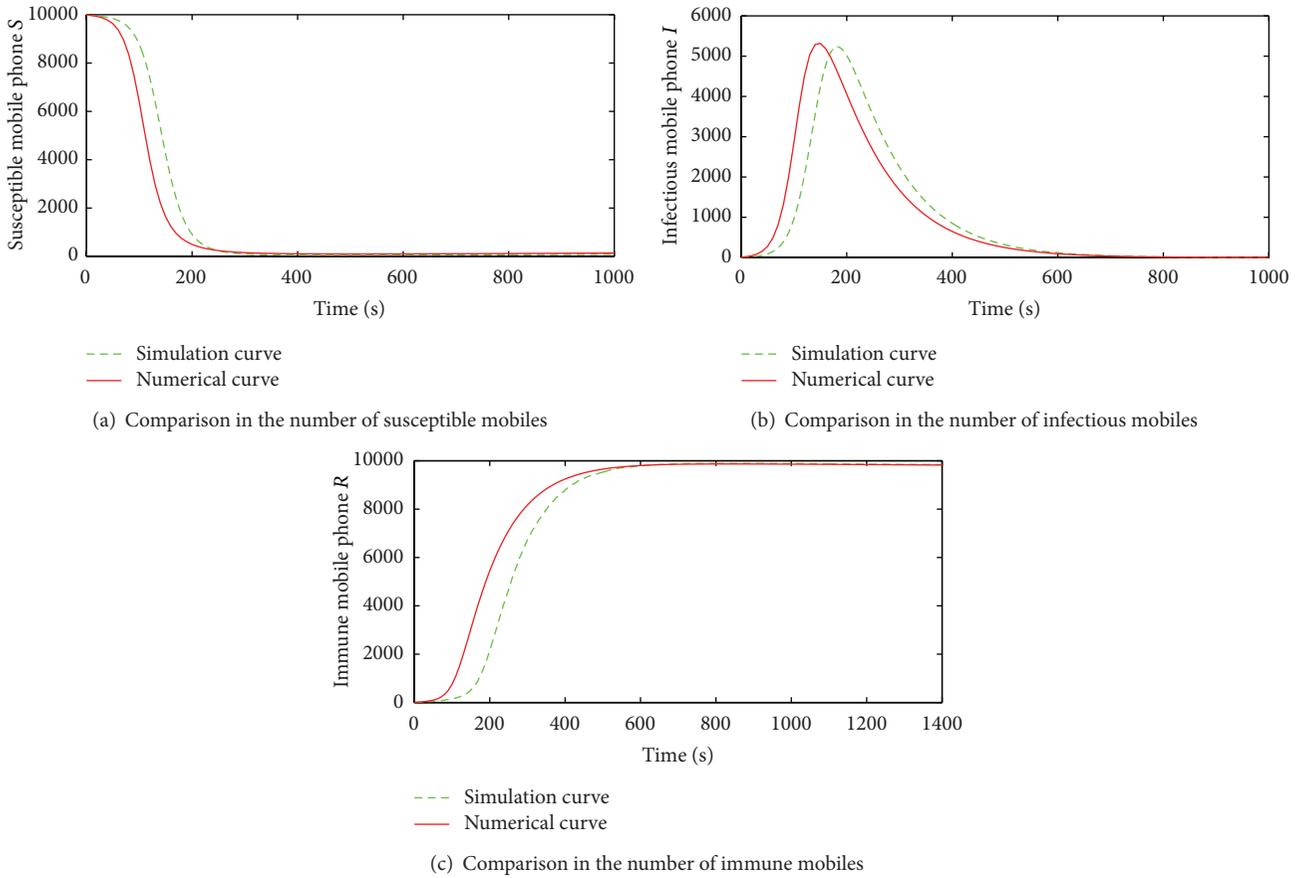


FIGURE 9: Comparisons between numerical and simulation results in the M-SIR model.

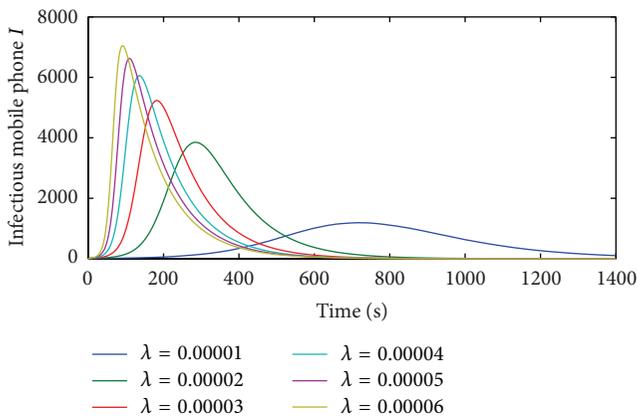


FIGURE 10: Effect of the contact infection rate λ in the M-SIR model.

All infected mobile phones vanish and the population in the long term is in an immune state. According to Theorem 3, the basic reproduction number R_1 is about $0.8 < 1$, which means that the propagation system of mobile phone virus under the M-SIR model will stabilize at its infection-free equilibrium point. In Figure 8 the susceptible, infected, and immune state mobile phones all reach their equilibrium points. This is fully consistent with the conclusions of Theorem 3.

The numerical results and simulation ones in susceptible, infected, and immune mobile phones, respectively, under the M-SIR model are shown in Figure 9.

The simulation curves of all states are almost consistent with the numerical ones which prove the correctness of the M-SIR model. The effect of contact infection rate to the propagation of vulnerability-exploiting mobile phone virus is shown in Figure 10.

Figure 10 shows the propagation trends of vulnerability-exploiting mobile phone virus with six different contact infection rates. With the increasing of the contact infection rate, the spread speed of the vulnerability-exploiting mobile phone virus is promoted, which makes the vulnerability-exploiting mobile phone virus reach the peak with little time. The scope of vulnerability-exploiting mobile phone virus also widens with the higher contact infection rate. The higher contact infection rate is the more nodes are infected. But the impact on the propagation is weakening with λ going up to some extent.

The performance of the immune rate to the propagation of vulnerability-exploiting mobile phone virus is discussed in Figure 11.

Figure 11 gives the propagations of the vulnerability-exploiting mobile phone virus with five different immune rates. The immune rate can affect the speed and scope of propagation. Obviously, the more the immune rate γ

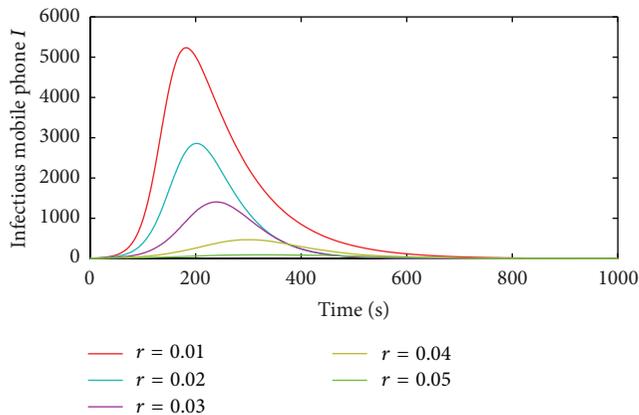


FIGURE 11: Effect of the immune rate γ in the M-SIR model.

is, the weaker the spread capability of the vulnerability-exploiting mobile phone virus is. Therefore, in order to guarantee normal applications of mobile phones and suppress the propagation speed and the propagation scope of mobile phone virus, we should choose a reasonable value for immune rate γ .

7. Conclusions

The objective of this paper is to model two kinds of mobile phone virus under two important factors (viz., the characteristics of mobile phone viruses and the network topology structure) and then to find out certain means to suppress the propagation of mobile phone virus. The M-SIS and M-SIR propagation models for mobile phone viruses are proposed, combining with the structural characteristics of the complex network.

The M-SIS propagation model is effective to predict the propagation of the user-tricking mobile phone virus. It reflects the characteristic of the mobile virus, which is difficult to completely remove, and the removed mobile phone virus can reinfect the same mobile phone.

The M-SIR propagation model is suitable to describe the vulnerability-exploiting mobile phone virus. It reflects the characteristic of the mobile virus, which spreads by exploiting vulnerabilities, and the mobile phone can be immune to the mobile phone virus after virus removal and patching.

Through analysis, the stable infection-free equilibrium point and the stability condition of the two propagation models are derived. The basic reproduction numbers R_0 and R_1 are given, which can determine whether the mobile phone virus extinguishes. When $R_0 < 1$ and $R_1 < 1$, the proposed M-SIS and M-SIR models have only a worm-free equilibrium, respectively, which is globally stable and implies that the worm dies out eventually. Then some numerical and simulation experiments are carried out which prove that our models are correct and fully consistent with the conclusions of our analysis. Our future work will expand this model which can characterize more features of mobile phone virus, for example, taking delay or impulse into consideration.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the Fundamental Research Funds of the Central Universities under Grant no. N120317001, Program for New Century Excellent Talents in University (NCET-13-0113), Natural Science Foundation of Liaoning Province of China under Grant no. 201202059, and Program for Liaoning Excellent Talents in University under LR2013011.

References

- [1] N. Leavitt, "Mobile phones: the next frontier for hackers?" *Computer*, vol. 38, no. 4, pp. 20–23, 2005.
- [2] Ericsson, "Traffic and market data report," 2011, http://www.ericsson.com/res/docs/2012/tmd_report_feb_web.pdf.
- [3] M. Ghaderi and S. Keshav, "Multimedia messaging service: system description and performance analysis," in *Proceedings of the 1st International Conference on Wireless Internet*, pp. 198–205, 2005.
- [4] Z. Zhu, G. Cao, S. Zhu, S. Ranjan, and A. Nucci, "A social network based patching scheme for worm containment in cellular networks," in *Proceedings of the 28th IEEE International Conference on Computer Communications (IEEE INFOCOM '09)*, pp. 1476–1484, Rio de Janeiro, Brazil, April 2009.
- [5] F-SECURE, "F-secure virus information pages: commwarrior," <http://www.f-secure.com/v-descs/commwarrior.shtml>.
- [6] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and S. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pp. 29–42, 2007.
- [7] R. Kumar, J. Novak, and A. Tomkins, "Structure and evolution of online social networks," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 611–617, 2006.
- [8] A. Yong-Yeol, H. Seungyeop, K. Haewoon, M. Sue, and J. Hawoong, "Analysis of topological characteristics of huge online social networking services," in *Proceedings of the 16th International Conference on World Wide Web*, pp. 835–844, 2007.
- [9] C. Fleizach, M. Liljenstam, P. Johansson, G. M. Voelker, and A. Mehes, "Can you infect me now?: malware propagation in mobile phone networks," in *Proceedings of the ACM Workshop on Recurring Malcode*, pp. 61–68, 2007.
- [10] F. Li, Y. Y. Yang, and J. Wu, "CPMC: an efficient proximity malware coping scheme in smartphone-based mobile networks," in *Proceedings of the 29th IEEE Conference on Information Communications (INFOCOM '10)*, pp. 2811–2819, San Diego, Calif, USA, 2010.
- [11] J. O. Kephart and S. R. White, "Directed-graph epidemiological models of computer viruses," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 343–361, 1991.
- [12] J. O. Kephart and S. R. White, "Measuring and modeling computer virus prevalence," in *IEEE Computer Security Symposium on Security and Privacy*, pp. 2–15, 1993.
- [13] R. W. Thommes and M. J. Coates, "Modeling virus propagation in peer-to-peer networks," in *Proceedings of the IEEE 5th*

- International Conference on Information, Communications and Signal Processing*, pp. 981–985, IEEE, Bangkok, Thailand, 2005.
- [14] C. Zou, D. Towsley, and G. Weibo, “On the performance of internet worm scanning strategies,” *Performance Evaluation*, vol. 63, no. 7, pp. 700–723, 2008.
- [15] H. Yuan and G. Chen, “Network virus-epidemic model with the point-to-group information propagation,” *Applied Mathematics and Computation*, vol. 206, no. 1, pp. 357–367, 2008.
- [16] B. K. Mishra and S. K. Pandey, “Dynamic model of worms with vertical transmission in computer network,” *Applied Mathematics and Computation*, vol. 217, no. 21, pp. 8438–8446, 2011.
- [17] C. W. John and J. M. David, “Modelling computer virus prevalence with a susceptible-infected-susceptible model with reintroduction,” *Computational Statistics Data Analysis*, vol. 45, no. 1, pp. 3–23, 2004.
- [18] Y. B. Kafai, “Understanding virtual epidemics: children’s folk conceptions of a computer virus,” *Journal of Science Education and Technology*, vol. 17, no. 6, pp. 523–529, 2008.
- [19] F. Wang, Y. Zhang, C. Wang, J. Ma, and S. Moon, “Stability analysis of a SEIQV epidemic model for rapid spreading worms,” *Computers and Security*, vol. 29, no. 4, pp. 410–418, 2010.
- [20] J. R. C. Piqueira and V. O. Araujo, “A modified epidemiological model for computer viruses,” *Applied Mathematics and Computation*, vol. 213, no. 2, pp. 355–360, 2009.
- [21] W. Fan and K. H. Yeung, “Online social networks-Paradise of computer viruses,” *Physica A: Statistical Mechanics and Its Applications*, vol. 390, no. 2, pp. 189–197, 2011.
- [22] B. Dwan, “The mobile phone virus,” *Network Security*, vol. 7, pp. 14–15, 2004.
- [23] J. Jamaluddin, N. Zotou, R. Edwards, and P. Coulton, “Mobile phone vulnerabilities: a new generation of malware,” in *Proceedings of the IEEE International Symposium on Consumer Electronics*, pp. 199–202, 2004.
- [24] D. Dagon, T. Martin, and T. Starner, “Mobile phones as computing devices: the viruses are coming!,” *Pervasive Computing*, vol. 3, no. 4, pp. 11–15, 2004.
- [25] W. Zhang, H. He, Q. Zhang, and T.-H. Kim, “PhoneProtector: protecting user privacy on the android-based mobile platform,” *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 282417, 10 pages, 2014.
- [26] C. Jin, X. Huang, and S. Jin, “Propagation model of mobile phone virus based on efficiency of immunization,” in *Proceedings of the International Conference on MultiMedia and Information Technology (MMIT ’08)*, pp. 500–502, 2008.
- [27] J. Ren, X. Yang, Q. Zhu, L.-X. Yang, and C. Zhang, “A novel computer virus model and its dynamics,” *Nonlinear Analysis: Real World Applications*, vol. 13, no. 1, pp. 376–384, 2012.
- [28] Y. Yao, W. Xiang, A. Qu, G. Yu, and F. Gao, “Hopf bifurcation in an SEIDQV worm propagation model with quarantine strategy,” *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 304868, 18 pages, 2012.
- [29] P. Wang, M. C. González, C. A. Hidalgo, and A. L. Barabási, “Understanding the spreading patterns of mobile phone viruses,” *Science*, vol. 324, no. 5930, pp. 1071–1076, 2009.
- [30] H. Zheng, L. Dong, and G. Zhuo, “An epidemic model of mobile phone virus,” in *Proceedings of the 1st International Symposium on Pervasive Computing and Applications*, pp. 1–5, IEEE, Urumqi, China, August 2006.
- [31] W. Xuetao, N. C. Valler, M. Faloutsos, I. Neamtiu, B. A. Prakash, and C. Faloutsos, “Smartphone viruses propagation on heterogeneous composite networks,” in *Proceedings of the IEEE 2nd Network Science Workshop*, pp. 106–109, 2013.
- [32] M. E. J. Newman, S. Forrest, and J. Balthrop, “Email networks and the spread of computer viruses,” *Statistical, Nonlinear, and Soft Matter Physics*, vol. 66, Article ID 035101, 2002.
- [33] J. P. Zhang and Z. Jin, “The analysis of an epidemic model on networks,” *Applied Mathematics and Computation*, vol. 217, no. 17, pp. 7053–7064, 2011.
- [34] C. C. Zou, D. Towsley, and W. B. Gong, “Code red worm propagation modeling and analysis,” in *Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS ’02)*, pp. 138–147, 2002.
- [35] F. W. Wang, Y. K. Zhang, and C. G. Wang, “Stability analysis of a SEIQV epidemic model for rapid spreading worms,” *Computers & Security*, vol. 29, no. 4, pp. 410–418, 2010.

Research Article

Fuzzy-Based Trust Prediction Model for Routing in WSNs

X. Anita,¹ M. A. Bhagyaveni,¹ and J. Martin Leo Manickam²

¹ Department of ECE, Anna University, Chennai 600025, India

² Department of ECE, St. Joseph's College of Engineering, Chennai 600119, India

Correspondence should be addressed to X. Anita; anitaaxtee@yahoo.co.in

Received 14 March 2014; Revised 15 June 2014; Accepted 17 June 2014; Published 14 July 2014

Academic Editor: Fei Yu

Copyright © 2014 X. Anita et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The cooperative nature of multihop wireless sensor networks (WSNs) makes it vulnerable to varied types of attacks. The sensitive application environments and resource constraints of WSNs mandate the requirement of lightweight security scheme. The earlier security solutions were based on historical behavior of neighbor but the security can be enhanced by predicting the future behavior of the nodes in the network. In this paper, we proposed a fuzzy-based trust prediction model for routing (FTPR) in WSNs with minimal overhead in regard to memory and energy consumption. FTPR incorporates a trust prediction model that predicts the future behavior of the neighbor based on the historical behavior, fluctuations in trust value over a period of time, and recommendation inconsistency. In order to reduce the control overhead, FTPR received recommendations from a subset of neighbors who had maximum number of interactions with the requestor. Theoretical analysis and simulation results of FTPR protocol demonstrate higher packet delivery ratio, higher network lifetime, lower end-to-end delay, and lower memory and energy consumption than the traditional and existing trust-based routing schemes.

1. Introduction

Wireless sensor networks (WSNs) have attractive wider range of application from civil sector to military [1–5]. A WSN consists of large number of resource constraint sensor nodes (SNs) deployed in hostile environment which makes it feasible for adversaries to perform varied types of attack [6, 7]. Due to the limited communication range [8], the SNs communicate with the sink in multihop. This cooperative nature of WSNs makes it vulnerable to insider attack which requires a trust management scheme. Most of the trust management schemes proposed in the literature were dependent on direct and indirect observations.

For direct trust computations, the promiscuous mode of operation was used in most of the trust-based routing protocols for neighbor monitoring to compute direct trust. It demands that nodes should be in wakeup state for longer duration which incurs more energy. The indirect trust was computed by receiving recommendation from all the neighbors which also consumes more energy. Moreover, the malicious nodes were identified based on their historical trust. Hence, in order to reduce the damages caused due to malicious activities in mission critical applications, the

behavior of a node should be predicted in advance based on the historical trust and the tendency of that node to maintain it consistently with the neighboring nodes in the network.

To address these issues, in this paper we proposed a novel fuzzy-based trust prediction model for routing (FTPR) in WSNs. FTPR was designed with the following objectives:

- (i) To minimize the energy consumption by avoiding promiscuous mode of operation for neighbor monitoring and by reducing the number of recommendations collected from neighbors to compute indirect trust.
- (ii) To reduce packet loss by identifying and eliminating malicious nodes by trust prediction. The trust of a neighboring node is predicted based on direct trust, number of trust fluctuations, and recommendation inconsistency.
- (iii) To thwart black hole attack, on-off attack, bad-mouthing attack, and conflicting behavior.

This paper is organized as follows: Section 2 discusses the related work. In Section 3, we described the framework of the proposed FTPR protocol. Simulation results and theoretical

analysis are discussed in Section 4, and Section 5 concludes the paper along with the future scope of the work.

2. Related Works

Several trust-based routing schemes proposed recently in the literature were designed not only to meet the security requirements but also considered the resource constraint nature of WSNs.

Paris et al. proposed a novel routing protocol to eliminate the selfish behavior of a neighbor [9]. The scheme used a novel routing metric called expected forwarding counter (EFW) that was used to thwart selective forwarding attack in wireless mesh networks. EFW was a cross-layer metric updated based on the observation of network layer and MAC layer. Mohi et al. proposed an intrusion detection scheme to eliminate denial-of-service (DoS) attack using Bayesian game approach in WSNs [10]. It was an incentive-based approach that motivates the nodes to behave properly. DoS attack was prevented based on the past behavior of the nodes in the Bayesian game formulation. Fuzzy-based detection and prediction system (FBDPS) [11] was proposed to detect distributed DoS (DDoS) attack. FBDPS compared the actual energy consumed by a neighbor with the normal value. When the energy consumed by that node was abnormal, then the node was considered as malicious. The drawback of these schemes was their ability to identify only a specific attack which may not be suitable for realistic applications.

Group-based trust management scheme (GTMS) [12] was designed to overcome black hole attack. The trust was dependent on direct and indirect monitoring. A distributed trust management scheme was adopted in intragroup level by collecting recommendations from all its group members to compute trust. A centralized trust management approach was used in intergroup level as each cluster head (CH) collected recommendations of other CHs directly from the sink. In order to reduce the memory consumption, the trust was represented as unsigned integers in the range from 0 to 100. The drawback of the GTMS was in the requirement of high energy CHs to directly communicate with the sink. Ambient trust sensor routing (ATSR) [13] was proposed to thwart black hole attack, bad-mouthing attack, and conflicting behavior. It was a geographic routing protocol and trust was computed based on direct and indirect observations. The trust values were represented as real numbers in the range from 0 to 1. Lightweight and dependable trust system (LDTS) [14] designed for hierarchical WSNs thwarted black hole and bad-mouthing attacks. The trust was computed based on direct and indirect observations. A centralized trust management scheme was used in intracluster and intercluster level. The trust value was assigned in the range of 0 to 10. All the above-mentioned schemes use promiscuous mode of operation for direct observation. The malicious nodes were identified only based on the past experience of a node.

In order to improve the network security through trust prediction, trust-based source routing protocol (TSR) [15] was proposed for mobile ad hoc networks (MANET). Fuzzy logic-based approach was used to predict the future behavior of a node from the knowledge of past behaviors. Trust was

derived from the direct observations and TSR was able to thwart black hole attack and grey hole attack. Ad hoc on-demand trusted multipath distance vector routing protocol (AOTMDV) [16] was proposed for MANET to eliminate modification attack, black hole attack, and grey hole attack. It derived the trust based on direct and indirect observations. It used all the received recommendations to compute the historical trust of node which made it vulnerable to bad-mouthing attack. Trust-aware secure routing framework (TSRF) [8] proposed for WSNs was based on direct and indirect observations. It was designed to thwart grey hole, tampering, on-off and bad-mouthing attack. As the trust value was represented as real numbers in the range from 0 to 1, TSRF consumed more memory. The malicious nodes were identified only based on historical trust of a node. Two-way acknowledgment-based trust (2-ACKT) [17] framework did not use promiscuous mode of operation for trust derivation and thwarted the black hole attack in WSNs. It used acknowledgments to derive the trust on the neighboring nodes. The scheme assumed that the malicious node drops data packets alone and not the acknowledgments. The scheme depends only on direct trust. As the recommendations were not gathered from the neighboring nodes, the decisions derived might not be fully consistent with the actual state of the network.

3. Fuzzy-Based Trust Prediction Routing Protocol

In this section, we discussed the detailed framework of our proposed FTPR protocol for WSNs. The assumptions made for the protocol design and the threat model employed for evaluating the performance of the protocol were also presented. FTPR protocol derived the trust based on direct and indirect observations.

3.1. Assumptions and Threat Model. In WSNs, each node forwards the data to the sinks with the help of other intermediate nodes. The number of sinks did not have any impact on the FTPR protocol. Hence, for simplicity we assumed there was only one sink in the network. We assumed a hierarchical topology that consists of CHs and cluster members (CMs). The FTPR protocol maintains intracluster and intercluster topologies. The intragroup topology consists of group of CMs which were attached to a CH. Intergroup topology comprises CHs and sink. The proposed FTPR framework consisted of two stages, namely, route discovery stage and data forwarding stage. During route discovery stage, each node discovered a route to the sink using a routing protocol and it was assumed that all nodes behaved legitimately during this stage. In data forwarding stage, each CM forwarded the data to the CH and CH in turn forwarded the data to the sink using multihop communication link. We assumed that some of the intermediate nodes in the multihop communication link behaved maliciously while forwarding the data packets. Basically trust is a relationship associating between two nodes for a specific action. In FTPR, we derived the trust between any two communicating nodes based on packet forwarding action. An adversary can modify the contents of data packets or various control packets exchanged

between the neighboring nodes in FTTPR protocol. In order to prevent fabrication of control or data packets, a secure communication channel which can be established with the help of any key management schemes [18–21].

We assumed malicious nodes manifested black hole attack [8], on-off attack [8], bad-mouthing attack [8], and conflicting behavior attack [8]. The black hole attack and on-off attack are manifested in the data forwarding plane. In black hole attack, a malicious node drops all the received packets instead of forwarding. It behaves well and bad alternatively in on-off attack, hoping that it can remain undetected while misbehaving. The bad-mouthing attacks and conflicting behavior are manifested in the trust evaluation plane. In bad-mouthing attack, a malicious node provides dishonest feedback to recommend good node as bad node and bad node as good node. In conflicting behavior attack, a malicious node behaves differently to nodes in different groups.

3.2. Network Topology. Consider the topology shown in Figure 1. Let node S be the subject wanted to evaluate the trust on its neighbor node target T . Node S forwards the data packet to its neighbor node T which in turn forwards the packet to its neighbor node sponsor R . On receiving the data packet, sponsor R forwards the data packet to its neighbor X as well as transmits an acknowledgment to node S through third party P as shown in the Figure 1. 2-ACKT protocol [17] was used for routing and determination of third party for the transmission of acknowledgment. A transaction was considered to be successful when the subject receives the acknowledgment for the data packet sent to target T . The higher the number of successful transactions, the higher the trust on the target.

In FTTPR, the trust was computed in intracluster level and intercluster level. In intracluster level, the CH aggregates all the data packets transmitted by the CMs. Some of the intermediate CMs in the communication link were malicious nodes. In FTTPR protocol, when a CM “ x ” wanted to communicate with the CH through the intermediate CM “ y ,” then node x would check the trust of node y in its trust table. If node y was legitimate, then node x would transmit the data packet to node y ; otherwise node x would find another route to the CH. Within the cluster, the trust was based on direct observation only as discussed in Section 3.3. In order to reduce the overhead involved in gathering recommendations, the indirect trust was not considered at the intracluster level. In intercluster level, the CH sends all the aggregated data to the sink through the multihop communication link which may contain malicious nodes. In FTTPR protocol, when a CH “ x ” wanted to communicate with the sink through the intermediate CH “ y ,” then node x would check the trust of CH “ y ” in its trust table. If the CH “ y ” was legitimate, then CH “ x ” would transmit the data packet to node y ; otherwise node x would find another route to the sink. The trust was computed based on direct trust and indirect trust as discussed in Section 3.4. Indirect trust was considered to maintain trust consistency within the network.

3.3. Direct Trust Computation. In trust-based routing schemes, direct trust was calculated based on direct

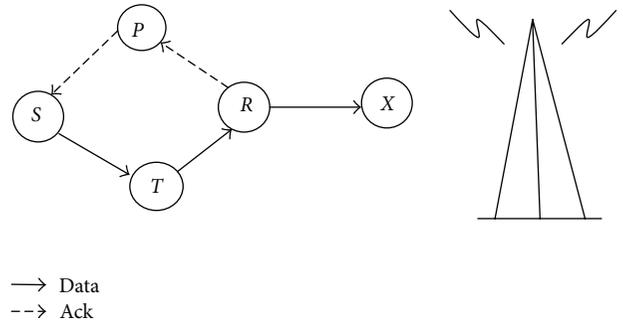


FIGURE 1: Network topology.

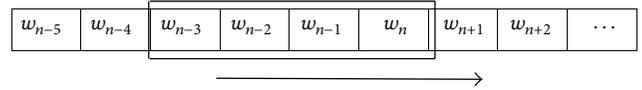


FIGURE 2: Sliding time window.

interaction with the neighbors. It must ensure that the neighbor had successfully received the packet and then forwarded the packet honestly by following the underlying routing protocol. The packet forwarding behavior of a CM was monitored by two-hop group acknowledgment scheme as discussed in [17].

In order to identify the inconsistent behavior of a node, the historical trust of a node should be considered to compute trust. To address these issues, a sliding time window scheme for trust calculation was used as shown in Figure 2. The time scale was divided into equal sized observation windows such as $w_{n-5}, w_{n-4}, w_{n-3}, w_{n-2}, w_{n-1}, w_n, w_{n+1}, \dots$, where w_n was the n th observation window. The numbers of successful and failed transactions were calculated for each observation window.

The sliding time window consisted of four observation windows as shown in Figure 2. The details of interactions in each observation window were recorded separately. Trust of a node was computed based on the numbers of successful and failed transactions. For each unit of time, the sliding time window slides one observation window to the right, thereby dropping the older experience by one unit and adds up the newer experience. Hence, the trust on the target during the n th observation window w_n depends on the numbers of successful and failed transactions during four observation windows, namely, w_n, w_{n-1}, w_{n-2} , and w_{n-3} . In order to store the details gained during direct interaction, we introduced a transaction table in the routing protocol. The observed successful and failed transactions were stored in the transaction table.

The transaction table of the CM consisted of the following fields:

{node id, number of successful transactions, number of failed transactions, trust level},

where *node id* was the address of the target, the *number of successful transactions* field was incremented by one whenever it received a link-layer acknowledgment from the target and a third-party acknowledgment within a timeout period,

the *number of failed transactions* was incremented by one whenever it received a link-layer acknowledgment from the target and not the third-party acknowledgment within a given timeout period, and *trust level* can take an integer value that lies in the range from 0 to 7. The computed trust value that lies in the range from 0 to 100 was mapped to a trust level which lies in range from 0 to 7 as discussed in [17].

As the CH computed trust based on direct observations as well as from the neighbor's recommendations, the transaction table of CH consists of the following fields.

(node id, number of recent transactions, number of successful transactions, number of failed transactions, trust level, peer recommendation, number of fluctuations, recommendation inconsistency, predicted trust),

where *peer recommendation* field was updated based on the recommendations received from the neighbors, *number of recent transactions* was incremented by one when it received a data packet from the neighbor *node id*, and *number of fluctuations* was used to monitor frequency of change in the trust level of neighboring node. A good node maintains a constant trust level and hence the number of fluctuations was low. The number of fluctuations (T_{NF}) can be updated as given by

$$T_{NF} = \begin{cases} T_{PNF} - T_{CTL} + T_{MTL}, & T_{CTL} < T_{PTL}, \\ T_{PNF} - 1, & T_{CTL} \geq T_{PTL}, \end{cases} \quad (1)$$

where T_{PNF} was the previous number of fluctuations and T_{MTL} was the maximum trust level; T_{PTL} and T_{CTL} were the previous and current trust level.

The equation was designed in such a way that T_{NF} increases rapidly when the trust level decreases.

The *recommendation inconsistency* was used to monitor whether the target was behaving in a consistent manner with all its neighbors and it is equal to the variance (σ^2) of the received recommendations. The variance of the recommendations received for a target that behaves uniformly with all its neighbors is lesser than for a target that behaves inconsistently with its neighbors. It was used to identify the conflicting behavior of node.

The *predicted trust* was updated using the fuzzy-based trust prediction model discussed in Section 3.6.

The direct trust between the subject S and target T based on the number of successful and failed transactions can be derived as follows. Let $T_{DV}(S, T)$ be the direct trust value of T computed by S . It was initially assumed to be 100 since all nodes were considered legitimate during network setup. Let T_{s_i} and T_{f_i} be the number of successful and failed transactions during the i th observation window, respectively; then

$$T_{DV}(S, T) = \left(\frac{\sum_{i=0}^{n-1} \alpha_i P_i (1 - P_i)}{\sum_{i=0}^{n-1} \alpha_i (1 - P_i)} \right) 100, \quad (2)$$

where n is the total number of observation windows and P_i represents positive trust value given by

$$P_i = \left(\frac{T_{s_i} + 1}{T_{s_i} + T_{f_i} + 2} \right) \left(1 - \frac{1}{T_{s_i} + 2} \right). \quad (3)$$

The term $(T_{s_i} + 1)/(T_{s_i} + T_{f_i} + 2)$ in (3) simply gives the ratio of number of successful transactions to the total number of transactions during the i th observation window. In order to give more importance to the number of successful transactions, the ratio $(T_{s_i} + 1)/(T_{s_i} + T_{f_i} + 2)$ was multiplied by the term $1 - (1/(T_{s_i} + 2))$. As the bad behavior of a node should be remembered for longer duration and the recent transactions must carry more significance than older transactions, P_i was multiplied by $\alpha_i(1 - P_i)$ while calculating the trust value, where α_i is an aging factor and $1 - P_i$ represents negative trust value for i th observation window. The value of α_i can take any values in the range from 0 to 1 subject to the condition that $\alpha_1 < \alpha_2 < \dots < \alpha_n$.

3.3.1. Trust Counselor. The two-hop acknowledgments depend on an alternate path through sponsor and third party. So, the trust of the target might be reduced due to the malicious activity of the sponsor or target or both. In order to identify the malicious activity of sponsor and target, a trust counselor component was introduced.

When the trust level of the target dropped below a warning threshold (T_w), then the subject initiated the counseling process by unicasting a warning packet to the target. T_w was not a constant and it was determined based on the trust level of the trusted neighboring nodes. The warning packet comprises the following fields:

(warning identifier, subject address, packet category, node address_1, node address_2),

where *warning identifier* and *subject address* uniquely identified the warning packet and *warning identifier* was defined by the subject; *packet category* was assigned 0 or 1 if the *node address_2* had not forwarded data or acknowledgment packets, respectively. It was assigned 0 when the subject initiated the counseling process as it assumed target had not forwarded the data packet. When the subject unicasted a warning packet to target, *node address_1* and *node address_2* were set as subject and target addresses, respectively.

On receiving the warning packet, target modified the packet category to 1 as it assumed that the sponsor had not forwarded the acknowledgment back to the subject through the alternate path. In this way the warning packet reached the third party through the sponsor. On receiving the warning packet, the third party unicasts a response packet back to the sponsor. The response packet consists of the following fields:

(response identifier, subject address, node address_1, node address_2, status),

where *response identifier* and *subject address* were assigned with the *warning identifier* and *subject address* as mentioned in the warning packet, respectively. *node address_1* was assigned with its own address (third party address) and *node address_2* was assigned with the sponsor address. The status field can be 0, 1, 2, 3, 4, or 5 as mentioned in the Table 1.

Status 0 denoted link failure and not yet rectified. Hence, the node was not ready to forward any packet. Status 1 referred to the condition that the node had not forwarded the packet due to link failure; but later the failure was

TABLE 1: Status field in response packet.

Status	Description
0	Link failure and not ready to forward
1	Link failure but now ready to forward
2	Insufficient resource and not ready to forward
3	Insufficient resource but now ready to forward
4	Non co-operative neighbor and no alternate path
5	Non co-operative neighbor but new alternate path available

rectified and ready to forward. Link failure could be due to network traffic conditions. Status 2 implied the inability of a node to participate in data forwarding activity due to energy or bandwidth unavailability. Status 3 referred to the condition that the node had not forwarded the packets due to insufficient resources; but later the resources were available and ready to forward the packets. Status 4 was used to indicate the existence of noncooperative malicious neighbor and the unavailability of an alternate path. Status 5 was used when the node had identified another alternate path due to the existence of noncooperative neighbor.

In this way the response travelled back to the subject through the sponsor and target. If the response packet did not reach the subject within the response wait time, then the subject reinitiated the route discovery process. The time interval between the transmission of warning packet to the target and the reception of response generated by any one of the three entities, namely, target, sponsor, and third party is called response wait time. The upper bound for the response wait time is equal to the three times the sum of the propagation delay and the processing delay experienced by the packets in the network.

3.4. Indirect Trust Computation. The indirect trust for the target was computed based on the recommendations obtained from neighbors. Recommendations also help in building a trust consistent with the network. In this section, we discussed the procedures for requesting recommendations and responding to such requests in WSNs. For requesting recommendations, subject broadcasts a trust request (TREQ) message to its neighbors in the transmission range. The TREQ message contains the following fields:

$$\langle TReqId, subject, target, I_{\min}, timestamp \rangle.$$

TReqId is the trust request identifier used to uniquely identify the TREQ message and *timestamp* indicated the issuing time. I_{\min} denotes the minimum number of interactions a recommender must have with the subject and it is given by

$$I_{\min} = \frac{1}{n} \sum_{i=0}^n T_{s_i} + T_{f_i}, \quad (4)$$

where n is the number of neighboring CHs.

Algorithm 1(a) describes the procedure for transmitting the TREQ message in detail. Upon receiving the TREQ message, the nodes that had prior trust relationship with target processed the TREQ message as given in the Algorithm 1(b) and unicast trust reply (TREP) message back to the subject. The TREP message contains the following fields:

$$\langle recommender, subject, target, T_{DL}(r_i, T) \rangle,$$

which indicates that a recommender (r_i) unicasts the TREP message back to the subject which had a trust level of $T_{DL}(r_i, T)$ with target.

Let us assume that ψ is the set of recommenders for the subject defined as

$$\psi = \{r_i : 0 \leq i \leq J - 1\}, \quad (5)$$

where J is the total number of recommenders. Then the indirect trust between the subject S and target T can be defined as

$$T_{IL}(S, T) = \frac{1}{J} \sum_{i=0}^{J-1} T_{DL}(r_i, T), \quad (6)$$

where $T_{DL}(r_i, T)$ is the trust level associated between the i th recommender and the target T . It was an integer value represented in the range from 0 to 7. It can be noted that a malicious node can manifest bad-mouthing attacks while responding to the TREQ message.

In order to detect such outliers from the received recommendations, we used empirical rule [22] with mean (μ) plus or minus one standard deviation (σ) as the recommendations were represented as integer values in the range from 0 to 7. Only those recommendations that lie in the range from $(\mu - \sigma)$ to $(\mu + \sigma)$ were considered consistent and used for calculating $T_{IL}(S, T)$ from (6). Let us assume that there are l number of outliers in the J number of received recommendations, and then (6) can be rewritten as

$$T_{IL}(S, T) = \frac{1}{J-l} \sum_{i=0}^{J-l-1} T_{DL}(r_i, T), \quad (7)$$

where $l \leq J$. A target was considered to be trusted when $T_{IL}(S, T)$ is greater than or equal to T_{TL} .

3.5. Trust Level. The trust level $T_{TL}(S, T)$ was computed based on $T_{DL}(S, T)$ and $T_{IL}(S, T)$.

$$T_{TL}(S, T) = \alpha T_{DL}(S, T) + \beta T_{IL}(S, T), \quad (8)$$

where $\alpha > \beta$ and $\alpha + \beta = 1$.

The trust level was predicted based on direct trust level, number of fluctuations, and recommendation inconsistency.

3.6. Fuzzy-Based Trust Prediction Model. The fuzzy trust prediction model has three inputs, namely, direct trust value ($T_{DTV}(S, T)$), number of fluctuations (T_F), and recommendation inconsistency (R), and one output, trust level (T). The fuzzy membership functions for direct trust value are LOW (L), MEDIUM (M), and HIGH (H). The fuzzy

(a) Algorithm for transmitting TREQ message

Subject broadcasts the following TREQ message to its neighbors in its transmission range
 $\langle TReqId, S, T, I_{min}, timestamp \rangle$

where S and T be the subject and target respectively.

Subject waits for recommendation messages until a predetermined time.

(b) Algorithm for processing the TREQ message

Neighbor r_i receives the TREQ message broadcast by the subject

if ($TReqId$ is new and not processed earlier)

r_i checks its trust table for trust relationship with target

if ($T_{DL}(r_i, T)$ is available in the transaction table) and (*number of recent transactions* $\geq I_{min}$) then

r_i unicasts the TREP message to S

$\langle r_i, S, T, T_{DL}(r_i, T) \rangle$

else

discards the TREQ packet

end if

else

discards the TREQ packet

end if

ALGORITHM 1: Algorithm for processing TREQ and TREP.

membership functions for the number of fluctuations are LOW (L), MEDIUM (M), and HIGH (H). The fuzzy membership functions for the recommendation inconsistency are LOW (L) and HIGH (H). The fuzzy membership functions for the fuzzy output predicted trust are VERY LOW (VL), LOW (L), MEDIUM (M), HIGH (H), and VERY HIGH (VH). The rule bases of the evaluator are shown in Table 2. The bases of functions are chosen so that they result in optimal value of performance measures. To illustrate one rule, the first rule can be interpreted as, "If the direct trust value is LOW and number of fluctuations is LOW and recommendation inconsistency is LOW, the predicted trust is medium." Similarly the other rules are framed.

4. Performance Analysis

The performance of FTPR protocol was evaluated using ns-2 simulator. The simulation parameters are listed in Table 3. We took a simulation area of 300 m \times 300 m, with six hundred nodes placed at random. The transmission range was 45 m. IEEE 802.15.4 was the MAC layer protocol used to evaluate the performance of the proposed trust model under attack conditions.

4.1. Metrics. The performance of FTPR routing protocol was evaluated using the following metrics.

- (i) *Packet Loss.* The total number of data packets lost legitimately or through malicious action without any notification.
- (ii) *Packet Delivery Ratio (PDR).* The ratio of total number of data packets delivered to the total number of data packets sent.

(iii) *Control Overhead.* The ratio of total number of control packets generated in the network to the total number of data packets received.

(iv) *Energy Consumption.* The average energy consumed by each node during the given simulation time and expressed in Joules (J).

(v) *Network Lifetime.* Time taken for the energy of the first node to fall from 0.5 J to zero and expressed in seconds.

(vi) *End-to-End Delay.* The delay experienced by the data packet during transmission from source to sink, including processing, queuing, and propagation delay.

(vii) *Communication Overhead.* The total number of packets generated for trust establishment in the network. It included TREQ, TREP, acknowledgments, and warning and response packets.

(viii) *Memory Consumption.* The total memory space exclusively used for trust derivation and representation, expressed in bits.

4.2. Simulation Results and Discussion. The performance of FTPR protocol was compared with 2-ACKT [17], GTMS [12], and AODV [23] protocols under varying number of malicious nodes as shown in Figure 3.

Among the total number of malicious nodes, 40 percent performed black hole attack, 30 percent performed on-off attack, 15 percent performed bad-mouthing attack, and 15 percent performed conflicting behavior attack. The FTPR, 2-ACKT, GTMS, and AODV routing protocols were tested against exactly the same scenario and connection pattern. The packet loss of FTPR, 2-ACKT, GTMS, and AODV protocols was plotted against varying percentage of malicious attacks as

TABLE 2: Fuzzy rule base.

Rule number	Direct trust value	Number of fluctuations	Recommendation inconsistency	Predicted trust
1	L	L	L	M
2	L	L	H	L
3	L	M	L	M
4	L	M	H	L
5	L	H	L	VL
6	L	H	H	VL
7	M	L	L	H
8	M	L	H	M
9	M	M	L	M
10	M	M	H	L
11	M	H	L	M
12	M	H	H	L
13	H	L	L	VH
14	H	L	H	M
15	H	M	L	H
16	H	M	H	M
17	H	H	L	H
18	H	H	H	L

TABLE 3: Simulation parameters.

Simulation time	800 secs
Simulation area	300 m × 300 m
Number of nodes	600
Frequency of operation	2.4 GHz
Node placement	Random
Transmission range	45 m
Propagation model	Two-ray
Movement model	Static
Traffic type	CBR (UDP)
Packet size	50 bytes
Packet interval	10 secs
Maximum number of malicious nodes	180
Type of attack	Black hole, on-off attack, conflicting behavior attack, and bad-mouthing attack
Initial energy	2 Joules
T_{TL}	4

shown in Figure 3(a). The AODV is a traditional routing protocol which cannot thwart any malicious attacks and hence resulted in higher packet loss compared to FTPR, 2-ACKT, and GTMS. The GTMS and 2-ACKT were designed to thwart only black hole attack. The presence of on-off attack, bad-mouthing attack, and conflicting behavior attack resulted in higher packet loss in GTMS and 2-ACKT protocols.

In GTMS and 2-ACKT, a node forwarded the data packets to its malicious neighbor until the trust level of that neighbor dropped below the T_{TL} . But in FTPR, the node transmitted a packet to its next hop neighbor based on the predicted trust level and as a result, the packet loss in FTPR protocol is 43.53 percent and 45.24 percent lower than GTMS protocol and 2-ACKT protocol, respectively. As the malicious nodes were identified only based on direct trust in 2-ACKT, the packet loss was slightly higher when compared to GTMS. It has a positive effect on the PDR of FTPR protocol as shown in Figure 3(b). The PDR of FTPR routing protocol is augmented by 43.91 percent, 19.78 percent, and 18.18 percent when compared to AODV, 2-ACKT, and GTMS protocols, respectively.

In FTPR routing protocol, only direct observation was considered to compute trust in intracluster level and in intercluster level; the recommendations were collected only from the most interacted neighbors. In GTMS, the recommendations were considered both in the intracluster level and in the intercluster level. As the promiscuous mode of operation was not used for neighbor monitoring, the control overhead of FTPR protocol is 13.99 percent lower than the GTMS protocol as shown in the Figure 3(c). As the recommendations were not gathered, the control overhead of 2-ACKT protocol is 15.04 percent lower than FTPR.

The lower control overhead and the effective trust prediction mechanism in FTPR reduce the energy consumption by 17.26 percent when compared to GTMS protocol as shown in Figure 3(d). The energy consumption of GTMS is higher as the nodes use promiscuous mode for neighbor monitoring and also as the CHs use high powered transmitters to communicate with the BS. The simulation was performed

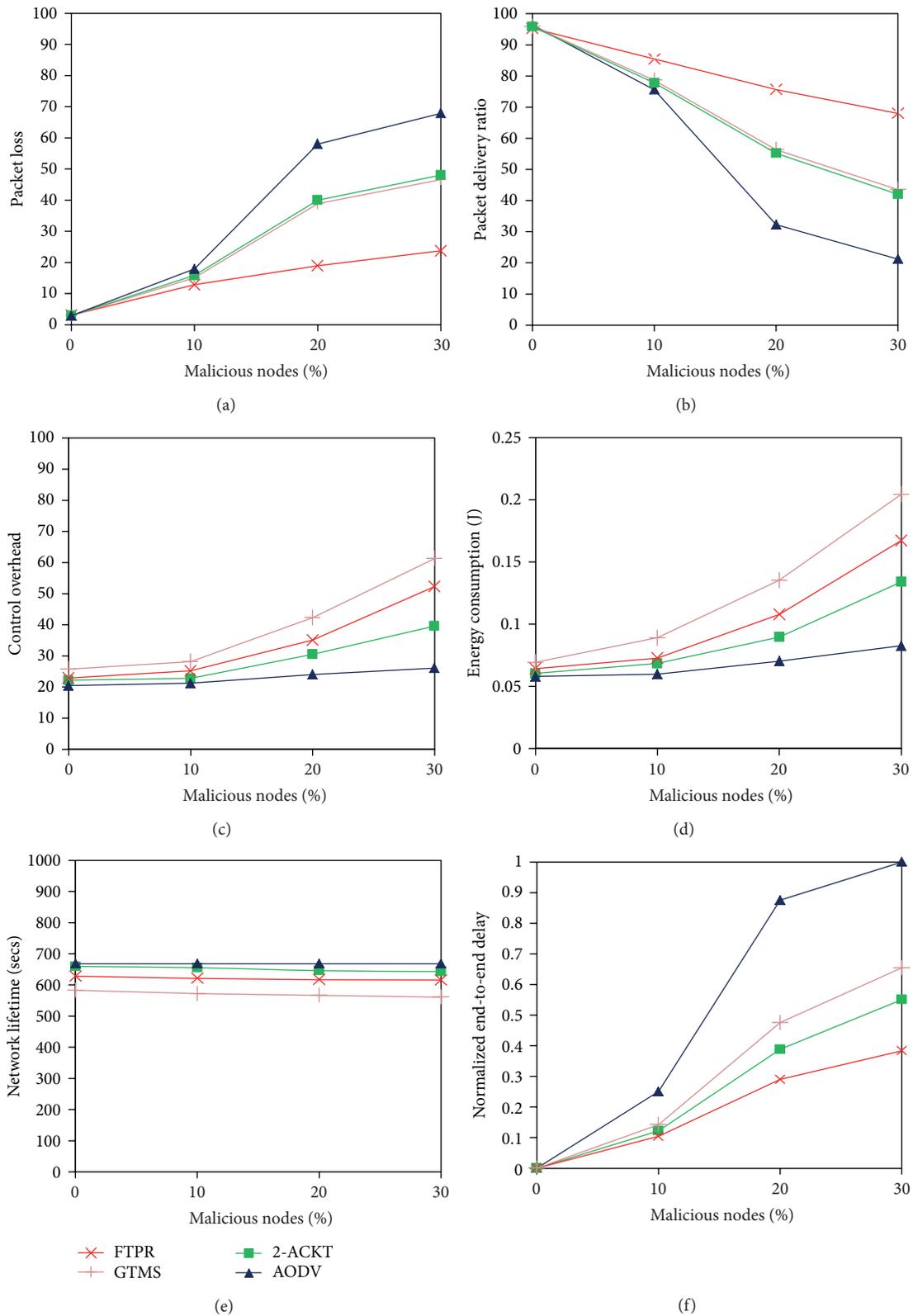


FIGURE 3: Performance comparison of FTPR, 2-ACKT, GTMS and AODV routing protocols under varying percentage of malicious attacks.

with an initial energy of 0.5 J to calculate the network lifetime. The lower energy consumption improves the network lifetime of FTPR protocol by 8.72 percent higher than GTMS routing protocol as shown in Figure 3(e). Even though the control overhead of 2-ACKT and AODV was lower than FTPR, the presence of malicious nodes resulted in higher end-to-end delay in AODV, 2-ACKT, and GTMS as most of the data packets had not reached the destination as shown in Figure 3(f).

4.3. Theoretical Analysis. Let us assume that “ N ” is the total number of SNs in the network and let “ h ” be the average number of hops between a CM and the sink. For this analysis, we assumed that all nodes in the network wanted to communicate with the sink using “ h ” hops and did not have any prior trust relationship between their neighbors. In this section, the performance of FTPR protocol was compared with the GTMS [12] protocol in terms of communication overhead and memory consumption.

4.3.1. Communication Overhead. In FTPR, when a node from i th cluster wants to communicate with the sink through its CH, the total number of acknowledgments generated for trust computation was $2(h - 1)$. Assuming a maximum of 30 percent malicious nodes, the maximum number of warning packets generated in the network was $0.3 \times 3(h - 1)$ and the maximum number of response packet generated was $0.3 \times 3(h - 1)$. Therefore, the communication overhead incurred by direct observation for one node to communicate with the sink is

$$2(h - 1) + (0.3 \times 6(h - 1)). \quad (9)$$

The indirect observation was considered to compute trust between clusters. FTPR broadcasts one recommendation request and receives recommendation only from a set of neighbors and let σ be the number of received recommendations. Therefore, the communication overhead incurred by indirect observation for one CH to communicate with the sink is

$$1 + \sigma(h - 2). \quad (10)$$

So the total communication overhead incurred by one node to communicate with the sink is

$$1 + \sigma(h - 2) + 2(h - 1) + (0.3 \times 6(h - 1)). \quad (11)$$

When all the N nodes in the network wanted to communicate with the sink, the communication overhead is given by

$$C_{\text{FTPR}} = N [1 + \sigma(h - 2) + 2(h - 1) + (0.3 \times 6(h - 1))]. \quad (12)$$

As described in [19], the communication overhead of GTMS [12] protocol can be derived as

$$C_{\text{GTMS}} = 2N(\delta + h - 4). \quad (13)$$

The communication overhead was plotted against the number of communicating nodes by setting $N = 144$ and $h = 5$ as shown in Table 4. The GTMS protocol with cluster

size of 9, 12, and 18 nodes was represented as GTMS-9, GTMS-12, and GTMS-18, respectively. Communication overhead of GTMS protocol increases with cluster size as shown in Table 4, whereas the communication overhead of FTPR protocol was same throughout as it was not dependent on cluster size.

In GTMS, the recommendations were collected even in intracluster level and so, when the cluster size was large, more numbers of recommendations were received in GTMS protocol. As a result, the communication overhead of GTMS-18 is 17.9 percent higher than our proposed FTPR protocol.

In GTMS protocol, the CH employed a high power transmitter to directly communicate with the sink for requesting and gathering recommendations about the state of neighboring CHs. But in FTPR protocol, all the nodes used similar low power transmitters and communicated with the sink using a multihop link. The exchange of acknowledgments, warning and response packets in the multihop link increases the overhead of FTPR protocol. As a result, the communication overhead of GTMS-9 was 35.8 percent lower when compared to that of FTPR. Hence, FTPR is more suitable for large cluster sized homogenous WSNs.

4.3.2. Memory Consumption. In FTPR, CMs and CHs maintained a transaction table to monitor and store the trust level of their neighbors. The fields in the transaction table and its memory size of CMs are shown in the Table 5. The node id occupied 2 bytes, number of successful transactions and number of failed transactions occupied 2 bytes each for each observation window present in the sliding time window, and trust level required 3 bits. Therefore, the memory required to store a record in the transaction table that represented the trust relationship with a neighbor was $2.375 + 4n$ bytes, where n is the number of observation windows.

The fields in the transaction table and its memory size of CHs are shown in Table 6. The CH contains 3 more additional fields than CM, namely, peer recommendations that occupy 3 bits and number of fluctuations and recommendation inconsistency occupying 4 bits each. Therefore, the memory required by a CH to store a record in the transaction table that represented the trust relationship with a neighbor was $3.75 + 4n$ bytes, where n is the number of observation windows.

The total size of the transaction table that represented the trust relationship between a CM and all its neighbors was

$$M_{\text{FTPR(CM)}} = (2.375 + 4n)(\delta - 1) \text{ bytes}, \quad (14)$$

where δ is the average size of cluster.

The total size of the transaction table that represented the trust relationship between a CH and all its neighbors was

$$M_{\text{FTPR(CH)}} = (6.125 + 4n) \left(\frac{N}{\delta} - 1 \right) \text{ bytes}, \quad (15)$$

where η_{av} is the average number of CHs.

TABLE 4: Communication overhead of FTTPR and GTMS.

Number of communicating nodes	Communication overhead			
	GTMS-9	GTMS-12	GTMS-18	FTPR
36	720	936	1368	1123
72	1440	1872	2736	2246
108	2160	2808	4104	3369
144	2880	3744	5472	4492

TABLE 5: FTTPR CM trust table.

Node id	Number of successful transactions			Number of failed transactions			Trust level
	w_1	...	w_n	w_1	...	w_n	
2 bytes	2 bytes		2 bytes	2 bytes		2 bytes	3 bits

TABLE 6: FTTPR CH trust table.

Node id		2 bytes
Number of recent transactions		2 bytes
Number of successful transactions	w_1	2 bytes
	⋮	
	w_n	2 bytes
Number of failed transactions	w_1	2 bytes
	⋮	
	w_n	2 bytes
Trust level		3 bits
Peer recommendations		3 bits
Number of fluctuations		4 bits
Recommendation inconsistency		4 bits
Predicted trust		3 bits

TABLE 7: Memory consumption of FTTPR and GTMS.

Number of neighboring nodes	Memory consumption (bytes)	
	GTMS	FTPR
9	26478	25998
12	32026.5	31762.5
16	41283	41139
18	46221	46109

5. Conclusions and Future Scope

In this paper, we proposed FTTPR protocol to effectively thwart black hole attack, on-off attack, conflicting behavior attack, and bad-mouthing attack. It employed a fuzzy-based trust prediction model to predict the future behavior of a neighboring node based on its historical behavior, trust fluctuations, and recommendation inconsistency. It derived the trust based on the direct and indirect observations. It reduces the energy consumption significantly by avoiding the promiscuous mode of operation for direct trust derivation and by gathering recommendations only from a subset of neighbors for indirect trust derivation. The memory consumption is significantly reduced by representing 8 bit trust values as 3 bit trust levels.

By considering the historical behavior of node using sliding time window scheme, the on-off attack was identified and eliminated. The bad-mouthing attack was avoided effectively by eliminating outliers from the received recommendations. The conflicting behavior was thwarted by considering recommendation inconsistency in the fuzzy-based trust prediction. The novel trust prediction model significantly improved the packet delivery ratio in the network. As the recommendations were received only from a subset of neighbors, there was a significant reduction in control overhead. Theoretical and simulation results of FTTPR protocol demonstrate higher packet delivery ratio, lower end-to-end delay, higher network lifetime, and lower memory consumption than the traditional and existing trust-based routing schemes. The limitation of this research work was that the nodes were assumed to have unique identity which is not suitable for some applications.

Therefore, the total memory consumed for the entire network which consisted of N number of CMs and N/δ number of CHs was

$$M_{\text{FTPR}} = (2.375 + 4n)(\delta - 1)N + (6.125 + 4n)\left(\frac{N}{\delta} - 1\right)\left(\frac{N}{\delta}\right) \text{ bytes.} \quad (16)$$

As described in [21], the memory consumption of GTMS [12] protocol can be derived as

$$M_{\text{GTMS}} = (6 + 4n) \left\{ N(\delta - 1) + \left(\frac{N}{\delta}\right) \left(\frac{N}{\delta} + \delta - 2\right) \right\} \text{ bytes.} \quad (17)$$

The memory consumption for FTTPR and GTMS protocols was plotted against the number of neighboring nodes and setting the size of the observation window $n = 4$ as shown in Table 7.

It was found that the memory consumption in FTTPR protocol is 19.9 percent lower than the GTMS protocol. It was achieved due to the use of 3 bits to represent trust levels of the neighboring nodes in the transaction table and also direct trust was only considered in intercluster level.

So, we plan to design a trust-based routing protocol for applications that require anonymous identity in WSNs.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] E. Fontana, J. F. Martins-Filho, S. C. Oliveira et al., "Sensor network for monitoring the state of pollution of high-voltage insulators via satellite," *IEEE Transactions on Power Delivery*, vol. 27, no. 2, pp. 953–962, 2012.
- [2] J. Valverde, V. Rosello, G. Mujica, J. Portilla, A. Uriarte, and T. Riesgo, "Wireless sensor network for environmental monitoring: application in a coffee factory," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 638067, 18 pages, 2012.
- [3] J. Wu and S. Shimamoto, "Usage control based security access scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, pp. 1–5, May 2010.
- [4] T. Watteyne and K. S. J. Pister, "Smarter cities through standards-based wireless sensor networks," *IBM Journal of Research and Development*, vol. 55, no. 1-2, pp. 7–10, 2011.
- [5] Y. Cao, C. Xu, J. Guan, F. Song, and H. Zhang, "Environment-aware CMT for efficient video delivery in wireless multimedia sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 381726, 12 pages, 2012.
- [6] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, pp. 52–73, 2009.
- [7] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Communication Surveys and Tutorials*, vol. 8, no. 2, pp. 2–23, 2006.
- [8] J. Duan, Y. Dong, H. Zhu, S. Zhang, and J. Zhao, "TSRF: a trust-aware secure routing framework in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 209436, 14 pages, 2014.
- [9] S. Paris, C. Nita-Rotaru, and F. Martignon, "EFW: a cross-layer metric for reliable routing in wireless mesh networks with selfish participants," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 576–580, 2011.
- [10] M. Mohi, A. Movaghar, and P. M. Zadeh, "A bayesian game approach for preventing DoS attacks in wireless sensor networks," in *Proceeding of the WRI International Conference on Communications and Mobile Computing, (CMC '09)*, pp. 507–511, Yunnan, China, January 2009.
- [11] C. Balarengadurai and S. Saraswathi, "Fuzzy based detection and prediction of DDoS attacks in IEEE 802.15.4 low rate wireless personal area network," *International Journal of Computer Science Issues*, vol. 10, no. 6, pp. 293–301, 2013.
- [12] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, and Y. Song, "Group-based trust management scheme for clustered wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 11, pp. 1698–1712, 2009.
- [13] T. Zahariadis, P. Trakadas, H. C. Leligou, S. Maniatis, and P. Karkazis, "A novel trust-aware geographical routing scheme for wireless sensor networks," *Wireless Personal Communications*, vol. 69, no. 2, pp. 805–826, 2013.
- [14] X. Li, F. Zhou, and J. Du, "LDTS: a lightweight and dependable trust system for clustered wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 924–935, 2013.
- [15] H. Xia, Z. Jia, X. Li, L. Ju, and E. H.-M. Sha, "Trust prediction and trust-based source routing in mobile ad hoc networks," *Ad Hoc Networks*, vol. 11, no. 7, pp. 2096–2114, 2013.
- [16] H. Xia, Z. Jia, L. Ju, X. Li, and E. H.-M. Sha, "Impact of trust model on on-demand multi-path routing in mobile ad hoc networks," *Computer Communications*, vol. 36, no. 9, pp. 1078–1093, 2013.
- [17] X. Anita, J. Martin Leo Manickam, and M. A. Bhagyaveni, "Two-way acknowledgment-based trust framework for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 952905, 14 pages, 2013.
- [18] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.
- [19] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pp. 62–72, October 2003.
- [20] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security And Privacy*, pp. 197–213, May 2003.
- [21] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 162–175, November 2004.
- [22] D. J. Rumsey, *Statistics For Dummies*, John Wiley & Sons, 2nd edition, 2011.
- [23] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing, Systems and Applications (WMCSA '99)*, pp. 90–100, New Orleans, La, USA, February 1999.

Research Article

Towards Dynamic Remote Data Auditing in Computational Clouds

**Mehdi Sookhak,¹ Adnan Akhunzada,¹ Abdullah Gani,¹
Muhammad Khurram Khan,² and Nor Badrul Anuar³**

¹ Center for Mobile Cloud Computing Research (C4MCCR), University of Malaya, 50603 Kuala Lumpur, Malaysia

² Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh 92144, Saudi Arabia

³ Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

Correspondence should be addressed to Mehdi Sookhak; m.sookhak@ieee.org

Received 14 March 2014; Accepted 8 May 2014; Published 9 July 2014

Academic Editor: Fei Yu

Copyright © 2014 Mehdi Sookhak et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cloud computing is a significant shift of computational paradigm where computing as a utility and storing data remotely have a great potential. Enterprise and businesses are now more interested in outsourcing their data to the cloud to lessen the burden of local data storage and maintenance. However, the outsourced data and the computation outcomes are not continuously trustworthy due to the lack of control and physical possession of the data owners. To better streamline this issue, researchers have now focused on designing remote data auditing (RDA) techniques. The majority of these techniques, however, are only applicable for static archive data and are not subject to audit the dynamically updated outsourced data. We propose an effectual RDA technique based on algebraic signature properties for cloud storage system and also present a new data structure capable of efficiently supporting dynamic data operations like append, insert, modify, and delete. Moreover, this data structure empowers our method to be applicable for large-scale data with minimum computation cost. The comparative analysis with the state-of-the-art RDA schemes shows that the proposed scheme is secure and highly efficient in terms of the computation and communication overhead on the auditor and server.

1. Introduction

Despite being a promising business concept, cloud computing is also becoming the fastest growing segment of the IT industry. It is all about moving services, computation, and/or data off-site to an internal or external, location-transparent facility or contractor. It is the way to increase the capacity or to add capabilities without investing in new infrastructure, licensing new software, or training new personnel. Despite many existent cloud definitions, they all agree that this paradigm aims at offering every network-accessible computing resource “as-a-service” (XaaS); however, the most highly structured definition comes from the National Institute of Standards and Technology (NIST) [1–3]. Thus, cloud computing is a key technology for empowering convenient, on-demand network access to a shared pool of configurable computing resources with negligible

service provider interaction or management effort. Therefore, enterprise and businesses tend to outsource their data on the cloud storage without investing in extra hardware, software, and the maintenance [4].

Despite the fact that cloud offers noticeable services for data owners, storing data to a remote server and entrusting management of data to a third party result in losing the physical control over the data [5, 6]. Though cloud has a promising, resilient, and reliable architecture, the data in the cloud is still susceptible to many threats and encounters many security challenges. It might lead to compromise the confidentiality, integrity, and availability of data. Examples are included to be able to delete less frequently accessed data to make available disk space or to hide data damage in order to protect the reputation. Recently, owners may lose their outsourced data on the cloud due to service and data disruptions in servers with major cloud infrastructure providers such as Amazon

S3 breakdown [7], Gmail email mass deletion [8], Sidekick Cloud Disaster [9], and Amazon EC2 service's outage [10, 11]. Besides, more than 535 data breaches in 2011 were reported with breaching of a cloud-based email service provider in Epsilon [12], stealing 3.3 million patients' medical details of Sutter Physicians Services, a major compromise of Sony PlayStation Network, Sony Online Entertainment, and Sony Pictures, stealing customers' information on EMC's RSA, exposing 150 million user accounts of Adobe Company, and most importantly leaking more than 104,000 employees' and contractors' private information of the Department of Energy in the U.S. that are the good examples of the data breach in 2013 [13]. Subsequently, the owners may have a high tendency to lose their outsourced data on the cloud.

To address this important issue, several researchers have proposed remote data auditing (RDA) protocols (e.g., [14–16]) to securely, frequently, and efficiently verify the integrity of the data stored over a cloud. RDA schemes mainly fall into three main categories: (1) integrity-based: it actually enables a cloud user to verify the integrity of data, (2) recovery based: data recovery is performed by leveraging error correction and erasure codes; however, normal integrity verification provides a way for recovering data in case of any possible corruption, and (3) deduplication-based: it is meant to improve the efficiency of data storage and mitigate the communication overhead of data outsourcing.

However, designing a proper remote data auditing mechanism, a set of noteworthy properties must be taken into consideration. These properties are as follows. (1) Efficiency: it is to verify data with a minimum possible amount of computational time, storage space, and communication between client and server. (2) Mode of verifiability (public/private): private verification methods exclusively work on the client's computer; however, in a public verification, the intricate task of verification is delegated to a third party often called third party auditor (TPA). The rationale behind this delegation is to take advantage of expertise and large capabilities of TPA as compared to limited computing power of client machine. (3) Frequency: it is the maximum number of times a user can verify his data. (4) Probability of detection: it represents the probability by which a potential data loss is discovered. (5) Dynamic update: it is the ability to verify the integrity of data without downloading the whole data when data is liable to different kinds of update operations, including insert, delete, modify, and append.

This paper proposes an efficient remote data auditing method based on an algebraic signature which allows the client to check data possession in cloud storage efficiently while incurring less computation overhead on cloud side and client side compared to homomorphic cryptosystem. Furthermore, we extend our data auditing scheme by designing an efficient data structure to support dynamic data update feature with minimum computation overhead on client and cloud server. The contribution of this paper is summarized as follows.

- (1) We propose an efficient remote data auditing scheme for data storage in cloud computing based on algebraic signature. Our data auditing scheme incurs the

minimum computation and communication cost on client and cloud server.

- (2) We design a new data structure to efficiently support dynamic data operations, such as insert, append, delete, and modify operations. This data structure empowers our method to be applicable for large scale data with least computation cost on client and server.
- (3) We implement our scheme to prove the security, justify the performance of our method, and compare with the state of the art data auditing methods.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 introduces the preliminaries and the fundamental concepts which are used in the construction of our method. In Section 4, we introduce the details of our remote data auditing scheme. We describe the security analysis of our scheme in Sections 5 and 6 gives the performance analysis in terms of computation overhead. Finally, the conclusion of this paper is presented in Section 7.

2. Related Work

Recently, a great deal of attention has been paid to the RDA schemes that are used to check the correctness of outsourced data in cloud computing [15–22].

Ateniese et al. [15] were the first to propose the provable data possession (PDP) scheme to check the correctness of the outsourced data statically in the cloud storage without having to retrieve the data. They used the RSA-based homomorphic verifiable tag to combine the tags and to build a proof message that permits the client to check whether the server has specific blocks, even when the client has no access to the blocks. However, the PDP scheme incurs high computation and communication cost on the server side due to the usage of RSA numbering over the whole file. It also has linear storage for the user and fails to provide secure data possession when the server has a malicious intent [23, 24]. In [16], Ateniese et al. considered static data update issue in the original PDP method [15] and developed a semidynamic data auditing method based on symmetric-key operations. This method allows a user to modify, delete, or append the stored data in the cloud. However, the data owner needs to regenerate all remaining challenges during the update operation, which makes it inapplicable for huge files.

Jules and Kaliski [25] defined a type of the RDA techniques, namely, proof of retrievability (POR) in which an auditor has also the capability to recover and mitigate data corruption by using forward error-correcting codes when data is stored in untrusted cloud. To achieve this goal, the data owner needs to create a set of sentinel blocks by using a one-way function and inserts the sentinels randomly on the data blocks before uploading to the server. If the server modifies a small portion of the file, the verifier can find it and check the integrity of a file easily due to the effect of file modification on the sentinels. However, the number of queries in such method depends on the number of inserted sentinel blocks. Moreover,

the POR method incurs high computation overhead on the client side because of the error recovery and data encryption processes. The work proposed by Shacham and Waters [26] improved the efficiency and security of the original POR based on the data fragmentation concept. The authors used the BLS homomorphic authentication [27] to generate a fixed size tag by aggregating all of the tags to minimize the network computation cost and used the Reed Solomon code to recover the corrupted blocks. The main disadvantage of this method is supporting static data update. Furthermore, during public verification process, the privacy of data cannot be protected against a trusted third party. The majority of POR methods failed to efficiently support dynamic data update because the server is unable to realize the relation between the data blocks and encrypted code words. Cash et al. [28] addressed this issue and designed a first dynamic POR scheme by using the ORAM technique [29]. The dynamic POR method also incurs high computation overhead on the client and server side.

The work by Erway et al. [18] addressed the dynamicity issue in the PDP schemes by combining the skip list [30], rank-based information, and authentication dictionary. Each node in this data structure needs to store the number of reachable nodes from this node as a rank. Although the dynamic PDP method ensures the integrity of variable-sized data blocks, it is unable to verify the integrity of individual block [31].

Wang et al. [19] employed a combination of the Merkle hash tree (MHT) [32] and bilinear aggregate signature [27] to propose a dynamic remote data auditing in cloud computing. The main contribution of this method is in manipulating the classic MHT construction by sorting the leaf nodes from left to right in order to support dynamic update and determine the insert, delete, or modify positions by following this sequence and computing the root in MHT. However, the method leaks the data content to the auditor and incurs heavy computation cost on the auditor.

Yang and Jia [17] implemented an efficient data auditing scheme to overcome the privacy issue in [19]. The authors used the bilinearity property of the bilinear pairing for generating an encrypted proof such that the auditor is only able to verify it. They also design a new data structure to support dynamic operations in which data owner needs to store a row, including block index and block logical location for each block of outsource file. During the delete and insert operations, the auditor has to find the position of the required block (i) and shift the remaining blocks ($n - i$) to create or delete a row in such data structure. However, by increasing the number of blocks in the data structure, the auditor needs to shift a huge number of blocks, which incur the high computation overhead on the auditor. The other drawback of this method is that deleting or inserting a large data block imposes high computational overhead on the auditor side. Furthermore, the bilinear pairing computation is more expensive than the algebraic structure that is used in our method [33, 34].

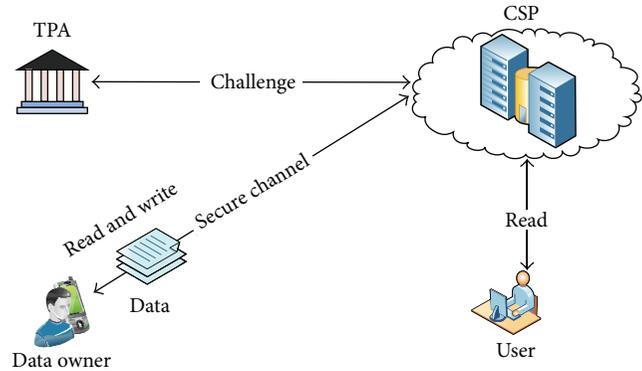


FIGURE 1: The network architecture of RDA in cloud computing.

3. Preliminaries

This section provides an overview on the background of our dynamic remote data auditing method. We first describe the general architecture of the remote data auditing protocol. Then, we state the fundamental technique of this method that is called algebraic signature in order to audit the outsourced data efficiently.

3.1. System Model. The architecture of RDA protocols in a network comprises four key entities: (1) user: it represents an enterprise or individual having permission to read the stored data in the cloud, (2) data owner (DO): it is enterprise or businesses which store their data in the cloud storage having the ability to do update operations (modify, delete, and insert), (3) cloud storage provider (CSP): this entity is responsible to back up the user data and generates a proof as a response of the received challenges, and (4) third party auditor (TPA): auditing the outsourced data and its verification is done by TPA. It actually ensures whether the data remains intact over the passage of time in public auditing models. Private auditing schemes, however, cannot support the TPA and DOs in order to check the integrity of the data. Figure 1 clearly depicts the typical RDA components and their interactions.

3.2. Algebraic Signatures. Algebraic signature is a type of hash functions with algebraic properties that allows computing the signatures of unseen messages in a limited way. The fundamental feature of algebraic signature schemes is to take a signature of the sum of some random blocks giving the same result as taking the sum of the signatures of the corresponding blocks [35].

Let an element γ in the Galois field be composed of a vector of various nonzero elements $\gamma = (\gamma_0, \gamma_2, \dots, \gamma_{n-1})$. An algebraic signature of file F including n data block $(f[1], f[2], \dots, f[n])$ is computed by

$$S_\gamma(F) = \sum_{i=1}^n f[i] \cdot \gamma^{i-1}. \quad (1)$$

In the following, a number of algebraic signature properties are listed.

Proposition 1. Litwin and Schwarz [36] have also shown that the algebraic signature of concatenation of two blocks $b[1]$ with length r and $b[2]$ is computed by

$$S_\gamma(f[i] \parallel f[j]) = S_\gamma(f[i]) \oplus r^\gamma S_\gamma(f[j]). \quad (2)$$

Proposition 2. The algebraic signature of summation of blocks of a file F is equal to summation of signature of each of the blocks

$$\begin{aligned} S_\gamma(f[1]) + S_\gamma(f[2]) + \dots + S_\gamma(f[n]) \\ = S_\gamma(f[1] + f[2] + \dots + f[n]). \end{aligned} \quad (3)$$

Proof. Assume that the file F is divided into n blocks and each of the block consists of r sectors. Then,

$$\begin{aligned} S_\gamma(f[1]) + S_\gamma(f[2]) + \dots + S_\gamma(f[n]) \\ = \sum_{j=1}^r f[1][j] \cdot \gamma^{j-1} + \sum_{j=1}^r f[2][j] \cdot \gamma^{j-1} \\ + \dots + \sum_{j=1}^r f[n][j] \cdot \gamma^{j-1} \\ = \sum_{j=1}^r \gamma^{j-1} (f[1][j] + f[2][j] + \dots + f[n][j]) \\ = S_\gamma(f[1] + f[2] + \dots + f[n]), \end{aligned} \quad (4)$$

where $f[i][j]$ indicates the j th bit of block i in file F . \square

Proposition 3. The algebraic signature of summation of two files F, G is equal to summation of signature of the files

$$S_\gamma(F + G) = S_\gamma(F) + S_\gamma(G). \quad (5)$$

Proof. Assume that the files F and G include n blocks. Then, the summation of signature of such files can be computed by

$$\begin{aligned} S_\gamma(F) + S_\gamma(G) &= \sum_{i=1}^n f[i] \cdot \gamma^{i-1} + \sum_{i=1}^n g[i] \cdot \gamma^{i-1} \\ &= \sum_{i=1}^n \gamma^{i-1} (f[i] + g[i]) \\ &= S_\gamma(F + G). \end{aligned} \quad (6) \quad \square$$

4. The Proposed Scheme

This section presents the applied techniques and algorithms of our dynamic remote data auditing scheme. We also show the correctness proof of our method by using the characteristics of the algebraic signature technique.

4.1. Data Auditing Algorithm. Suppose that file F includes n data blocks and each of the block is divided into r sectors by using the data fragment technique. If the last block has less number of sectors, we increase the size of the block by setting $f_{i,j} = 0$ for $j \leq r$. Our data storage auditing scheme consists of the following phases.

Setup. The DO firstly generates the public and secret key by executing the keygen algorithm ($\text{KeyGen}(1^k) \rightarrow (pk, sk)$). Then, the unique tag (metadata) for each block of input file is computed based on the algebraic signature of the block using the following formula:

$$T_i = S_\gamma(f[i] \parallel (\text{ID}_F \parallel i \parallel L_i \parallel V_i)), \quad (7)$$

where $f[i]$ is i th block of file F , ID_F is unique identity of the file, L_i is the logical number of file in the DCT table, and V_i indicates the version of data block. Also, the DO computes $C_i = S_\gamma(\text{ID}_F \parallel i \parallel L_i \parallel V_i)$ for each data block to prevent the replay attack. When all of the tags are generated, the DO outsources the data blocks along with the considering tags to the cloud $\{f[i], T_i, C_i\}_{i=1}^n$.

Challenge. When the DO decides to check the correctness of the outsourced data, it selects c data blocks randomly as a challenge message ($\text{chal} = \{cs_i\}_{i=1}^c$) by using pseudorandom permutation [37] keyed with a fresh randomly chosen key in order to prevent the server from anticipating the block indices.

Proof. Upon receiving the challenge message, the cloud computes a linear combination of the blocks (σ) and the aggregate authenticator tags (μ) as a proof message based on the received challenge and the corresponding tags by using

$$\mu = \sum_{i=cs_1}^{cs_c} T_i \oplus C_i \quad \sigma = \sum_{i=cs_1}^{cs_c} f[i]. \quad (8)$$

Verification. Upon receiving the pair (μ, σ) , the DO uses the algebraic signature of the block tags to verify the correctness of the blocks by using the following formula:

$$S_\gamma(\sigma) \stackrel{?}{=} \mu. \quad (9) \quad \square$$

4.2. Dynamic Data Operations. To support dynamic data update, we propose a data structure that is called Divide and Conquer Table (DCT). The DCT prevents the server from using the previous version of the stored data instead of the updated one to pass the verification phase (replay attack). The DCT consists of two components: logical index (L_i) and version number (V_i). The L_i indicates the original index of data block and the V_i indicates the current version of block on the basis of number of updates. When a data block is updated, the considered V_i in DCT must be incremented by 1. The index of each block in DCT also denotes the physical position of the outsourced data block.

This data structure must be created by the DO before outsourcing a data block to the cloud. The DO is in charge

of managing the DCT during update operation. Therefore, by increasing the size of file, a huge computation overhead is imposed on the owner side. For example, to insert a new data block after the i th block, the data owner must shift $n-i$ blocks, which waste the time and impose additional computation overhead. To overcome this issue, we reduce the size of the DCT by dividing it into k data structures in which each of them is able to store $\lceil n/k \rceil$ of the data blocks. As a result, when the DO decides to insert a new block after the i th block, the data owner only needs to shift the $\lceil n/k \rceil - i$ data block. The experimental results show that the proposed data structure is able to support the large scale data efficiently. In the rest of this section, we discuss how our scheme performs dynamic data operations, such as modify, insert, delete, and append.

Data Modification. One of the important requirements of remote data auditing techniques is to support the data modification operation in which the DO has capability to replace the specified blocks with new ones. Suppose that the DO wants to modify the i th block of the file F ($f[i]$) to $f'[i]$. The DO executes the modification algorithm to perform the following modifications:

- (1) finding the specific DCT that has the required block on the basis of the ranges of DCTs and then updating $V_i = V_i + 1$;
- (2) generating a new block tag for modified data block by

$$\begin{aligned} T'_i &= S_\gamma (f' [i] \parallel (\text{ID}_F \parallel i \parallel L_i \parallel V'_i)), \\ C'_i &= S_\gamma (\text{ID}_F \parallel i \parallel L_i \parallel V'_i); \end{aligned} \quad (10)$$

- (3) sending the modification request message to the CSP, which includes $(\text{ID}_F, i, f' [i], T'_i, C'_i)$.

Upon receiving the modification request message, the CSP replaces the block $f[i]$ with $f'[i]$ and updates the version of data block by replacing the tag (T_i, C_i) with (T'_i, C'_i) . Figure 2 shows that the data owner modifies block $f[7]$ when the number of entities in each of table is 5.

Data Insert. To insert a new data block ($f^* [i]$) after block ($f[i]$), the DO needs to run insert algorithm to perform the following modifications:

- (1) finding the i th block of the file F by comparing its index with the range of DCTs;
- (2) constructing a new row in the DCT after i th block and shifting the subsequent blocks $(\lceil n/k \rceil - i)$ one position down; the DO also sets the original index of data block $L^*_{i+1} = n + 1$ and the version number of the block $V^*_{i+1} = 1$ where n is number of blocks;
- (3) the Do needs to increase maximum and minimum ranges of subsequent DCTs;
- (4) generating a block tag (T^*_{i+1}, C^*_{i+1}) for the new data block by

$$\begin{aligned} T^*_{i+1} &= S_\gamma (f^* [i + 1] \parallel (\text{ID}_F \parallel i + 1 \parallel L^*_{i+1} \parallel V^*_{i+1})), \\ C^*_{i+1} &= S_\gamma (\text{ID}_F \parallel i + 1 \parallel L^*_{i+1} \parallel V^*_{i+1}); \end{aligned} \quad (11)$$

- (5) sending the insert request message to the CSP, which includes $(\text{ID}_F, i + 1, f^* [i], T^*_{i+1}, C^*_{i+1})$.

When the CSP receives such message, the new data block and the considering tag are inserted after position i in the file. For example, Figure 3 illustrates that the data owner only needs to shift 3 entities down to insert a new block ($\text{DCT}_2[3] = \{16, 1\}$) after block $f[7]$ in the second table and increases all of range of next tables and the uprange of DCT_2 .

Data Append. The append operation refers to the insertion of a new data block into the end of data blocks. Therefore, the Do only needs to insert a new row to the end of the last DCT without having to shift any entities of the DCTs. For instance, Figure 4 shows that to append a new block, the data owner only needs to create a free row for the last table and increase its uprange ($\text{UR}_3 = \text{UR}_3 + 1$).

Data Delete. The delete operation is the opposite of the insert operation in which the i th block of the file of F ($f[i]$) is removed. To achieve this goal, the DO finds the CDT that contains the required block on the basis of the DCT ranges. Then, the block is removed by shifting all of the subsequent blocks $(\lceil n/k \rceil - i)$ one position up. The DO sends a request to delete the i th block of the file of F . As it is shown in Figure 5, to delete a 4th data block ($f[4]$), the data owner only needs to shift up 1 row ($f[5]$) and the range of next tables will be reduced along with the uprange of the first table ($\text{UR}_1 = \text{UR}_1 - 1$).

5. Security Analysis

In this section, we evaluate the surety of our remote data auditing construction in term of security and correctness.

In the first step, we analyze the correctness of the verification algorithm. Upon receiving the challenge message $(\{cs_i\}_{i=1}^c)$, the CSP generates a pair (μ, σ) as a proof message. We extend (8) by using the properties of algebraic signature as follows:

$$\begin{aligned} \mu &= \sum_{i=cs_1}^{cs_c} T_i \oplus C_i \\ &= \sum_{i=cs_1}^{cs_c} S_\gamma (f [i] \parallel (\text{ID}_F \parallel i \parallel L_i \parallel V_i)) \\ &\quad \oplus S_\gamma (\text{ID}_F \parallel i \parallel L_i \parallel V_i) \\ &= \sum_{i=cs_1}^{cs_c} S_\gamma (f [i]) \oplus r^\gamma S_\gamma (\text{ID}_F \parallel i \parallel L_i \parallel V_i) \\ &\quad \oplus S_\gamma (\text{ID}_F \parallel i \parallel L_i \parallel V_i) \\ &= \sum_{i=cs_1}^{cs_c} S_\gamma (f [i]). \end{aligned} \quad (12)$$

When the DO obtains the proof message from the server, it verifies the proof message to ensure the storage correctness by

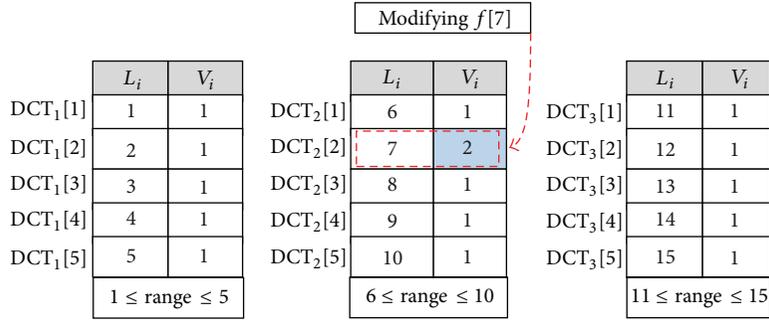


FIGURE 2: Performing modify operation on $f[7]$ when the number of blocks in each table is 5.

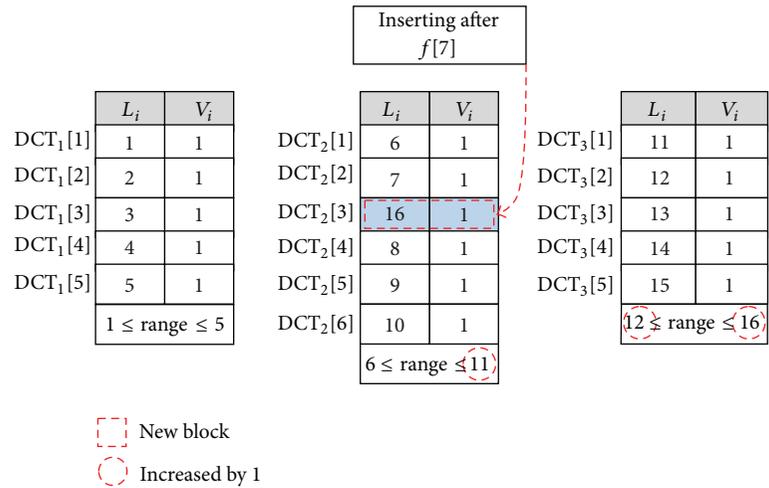


FIGURE 3: Inserting a new data block after $f[7]$ when the number of blocks in each table is 5.

using (9). We rewrite the equation on the basis of the algebraic signature properties to show why it is true:

$$\begin{aligned}
 S_\gamma(\sigma) &= S_\gamma\left(\sum_{i=cs_1}^{cs_c} f[i]\right) \\
 &= S_\gamma(f[cs_1] + \dots + f[cs_c]) \\
 &= S_\gamma(f[cs_1]) + \dots + S_\gamma(f[cs_c]) \\
 &= \sum_{i=cs_1}^{cs_c} S_\gamma(f[i]) = \mu.
 \end{aligned}
 \tag{13}$$

Our scheme relies on the algebraic signature that generates a small entity as a signature for each block and is able to show any modifications in the original block. The algebraic signature also has the capability to verify a large amount of stored data on the distributed storage systems with minimum computation and communication overhead [35]. On the other hand, probability of collision in the algebraic signature is negligible [36]. For example, if the length of signature is 64 bits, the probability of collision is very small (2^{-64}). As a result, the algebraic signature technique is useful for verifying

the correctness of outsourced data specially by using the resource restricted devices.

6. Performance Analysis

In this section, we assess the performance of the proposed remote data auditing method. We also analyze the probability of misbehavior detection of this scheme. We give the computation complexity during the insert, delete, append, and modify operations and compare the results with the state-of-the-art remote data auditing methods proposed by Yang and Jia [17] and Wang et al. [19].

6.1. Probability of Misbehavior Detection. Our remote data auditing scheme is constructed on the basis of a random sampling strategy to reduce the workload on the server. In the sampling technique, the input file (F) is divided into several blocks (n) and a random number of blocks (c) are used to perform batch processing. We analyse the probability of misbehaviour detection of our scheme based on the block sampling.

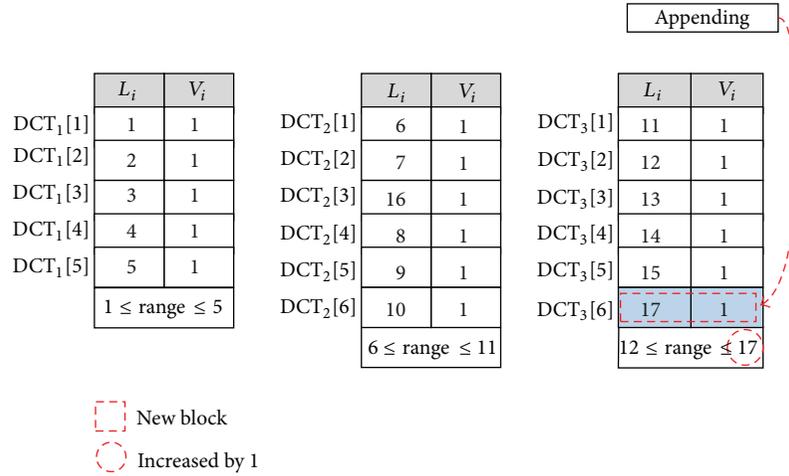


FIGURE 4: Appending a new data block.

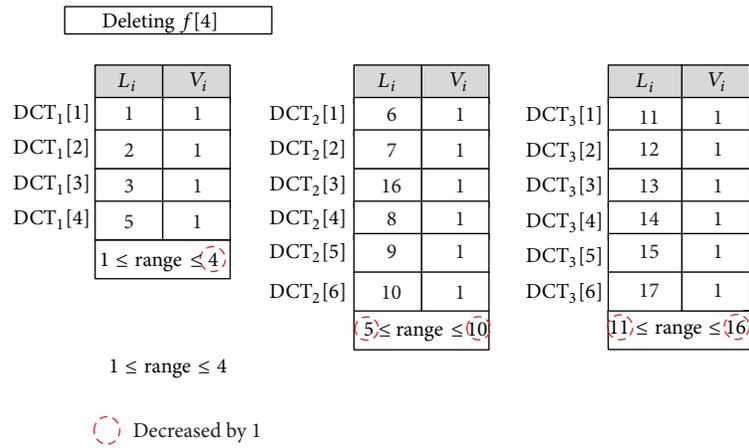


FIGURE 5: Deleting the $f[4]$ when the number of blocks in each table is 5.

Suppose the CSP modifies m blocks out of the n outsourced blocks. Then, the probability of corrupted blocks is equal to $p_m = m/n$. Let c be the number of blocks that the DO asks to verify the outsourced data in the challenge step and let r be the number of sectors in each block. Let x be a discrete random variable that indicates the number of blocks chosen by the DO that matches the blocks modified by the CSP. We compute the probability that at least one of the blocks picked by the DO matches one of the blocks modified by the server, namely, $P_x (x \geq 1)$, as follows:

$$\begin{aligned}
 P_x (x \geq 1) &= 1 - P_x (x = 0) \\
 &= 1 - \left(\frac{n-m}{n}\right) \\
 &\quad \cdot \left(\frac{n-m-1}{n-1}\right) \cdots \left(\frac{n-m-c+1}{n-c+1}\right)
 \end{aligned}$$

$$\begin{aligned}
 &= 1 - \left(1 - \frac{n}{m}\right) \\
 &\quad \cdot \left(1 - \frac{m}{n-1}\right) \cdots \left(1 - \frac{m}{n-c+1}\right) \\
 &= 1 - \prod_{i=0}^{c-1} \left(1 - \frac{m}{n-i}\right).
 \end{aligned} \tag{14}$$

On one hand,

$$\begin{aligned}
 \left(1 - \frac{m}{n-i}\right) &\leq \left(1 - \frac{m}{n}\right) \implies \prod_{i=0}^{c-1} \left(1 - \frac{m}{n-i}\right) \leq \left(1 - \frac{m}{n}\right)^c \\
 \implies 1 - \prod_{i=0}^{c-1} \left(1 - \frac{m}{n-i}\right) &\geq 1 - \left(1 - \frac{m}{n}\right)^c.
 \end{aligned} \tag{15}$$

Therefore,

$$\stackrel{(14),(15)}{\implies} P_x(x \geq 1) \geq 1 - \left(1 - \frac{m}{n}\right)^c = 1 - (1 - p_m)^c. \quad (16)$$

Since, each of the blocks consists of r sectors, such probability on the basis of sector corruption p_s is computed by

$$\begin{aligned} p_m &\geq 1 - (1 - p_s)^r \implies (1 - p_m)^c \leq ((1 - p_s)^r)^c \\ &\implies 1 - (1 - p_m)^c \geq 1 - (1 - p_s)^{rc} \\ &\implies P_x(x \geq 1) \geq 1 - (1 - p_s)^{rc}. \end{aligned} \quad (17)$$

On the other hand,

$$\begin{aligned} \left(1 - \frac{m}{n-i}\right) &\geq \left(1 - \frac{m}{n-c+1}\right) \implies \prod_{i=0}^{c-1} \left(1 - \frac{m}{n-i}\right) \\ &\geq \left(1 - \frac{m}{n-c+1}\right)^c \implies 1 - \prod_{i=0}^{c-1} \left(1 - \frac{m}{n-i}\right) \\ &\leq 1 - \left(1 - \frac{m}{n-c+1}\right)^c. \end{aligned} \quad (18)$$

Therefore,

$$P_x(x \geq 1) \leq 1 - \left(1 - \frac{m}{n-c+1}\right)^c. \quad (19)$$

Then, we can conclude that the probability of misbehavior detection is in

$$1 - \left(1 - \frac{m}{n}\right)^c \leq P_x(x \geq 1) \leq 1 - \left(1 - \frac{m}{n-c+1}\right)^c. \quad (20)$$

Suppose the DO divides 1 GB file into 125000 blocks 8 KB and outsources the blocks in the cloud. Figure 6 shows the required number of challenge blocks (c) that are used to detect the different number of corrupted blocks (m) when the probability of misbehaviour detection is collected from a set of $P_x = \{0.7, 0.8, 0.9, 0.99, 0.99999\}$. For example, if the server modifies 0.1 of the outsourced blocks (n), the DO needs to randomly select 98 block as a challenge to achieve P_x of at least 0.99999. As it is clear, by increasing the number of corrupted blocks, the least number of challenge blocks is required to achieve such a probability of detection.

Figure 7 illustrates the number of challenge blocks when the probability of misbehavior detection is between 0.5 and 1 with variable rate of data corruption. For example, if the server modifies 0.01% out of the n outsourced blocks, the DO needs to randomly select 520 data blocks as a challenge for detecting the corrupted blocks with probability of 0.9899. It also can be seen that when the rate of corrupted blocks is more than 0.3%, the minimum numbers of challenge blocks are used to audit the outsourced data.

6.2. Evaluation and Experimental Results. Table 1 shows a comparison of our scheme and state-of-the-art remote data

TABLE 1: Comparison of different remote data auditing scheme.

Metric	Scheme		
	[19]	[17]	Our scheme
Communication	$O(c \log n)$	$O(c)$	$O(c)$
Computation Auditing			
Server	$O(c \log n)$	$O(cs)$	$O(cs)$
Verifier	$O(c \log n)$	$O(c)$	$O(c)$
Computation modify			
Verifier	$O(c \log n)$	$O(c)$	$O(c)$
Computation insert			
Verifier	$O(c \log n)$	$O(n)$	$O(n/k)$
Computation delete			
Verifier	$O(c \log n)$	$O(n)$	$O(n/k)$
Computation append			
Verifier	$O(c \log n)$	$O(c)$	$O(c)$

auditing protocols based on the communication and computation overhead through dynamic data update, where n is the number of blocks, s is the number of sectors of a block, c indicates the number of challenge blocks in each auditing query, and k indicates the number of the DCTs.

From the table, we can find that the Wang et al. method [19] has the maximum computation overhead during dynamic data update. In the Yang scheme [17], to insert a block after i or delete a specific block ($f[i]$), the verifier must shift $(n-i)$ entities in the data structure. Therefore, the computation overhead of such method during insert and delete operations is $O(n)$. We improve our auditing scheme by designing a new data structure (DCT) to reduce the computation overhead. As mentioned earlier in Section 4.2, the verifier only needs to shift $(n/k - i)$ blocks that incurs $O(n/k)$ computation overhead on the verifier. It is important to mention that to find a block ($f[i]$) in DCT structure, the verifier only needs to divide the location of block into k and find the appropriate DCT that incurs negligible overhead on verifier.

The first step to perform insert, delete, append, and modify operations is to identify that the i th data block of the file is going to be a part of which DCTs. The auditor is able to find the i th data block by computing the quotient of a division of the requested block index (i) by the number of data block in each DCT structure (k). Such quotient shows the DCT number and the remaining of the division shows the position of block in the found DCT. To insert a new data block after j th data block or delete the j th data block, the auditor has to find the considered DCT and the position of the block in it (i) and then moves forward or backward the remaining blocks of the DCT ($n/k - i$). Since each DCT contains (n/k) blocks, performing insert and delete operations incurs $O(n/k)$ computation overhead on the auditor. The modification operation incurs $O(C)$ as a computation overhead on the auditor. It is because the auditor only requires finding the position of i th data block in the

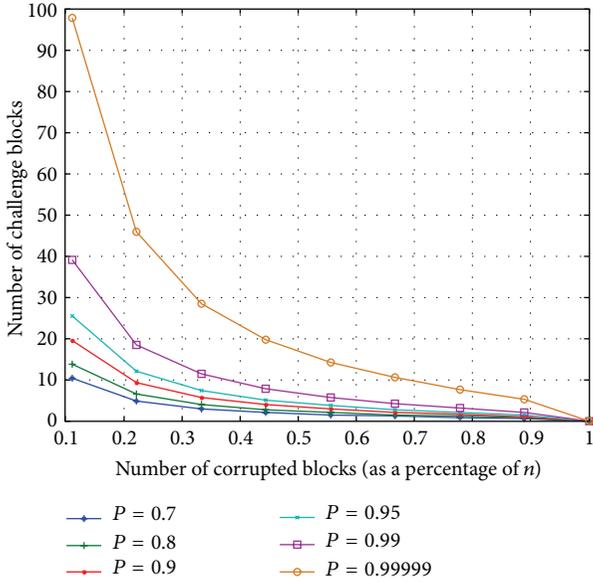


FIGURE 6: Number of required blocks as a challenge message under different number of data corruptions.

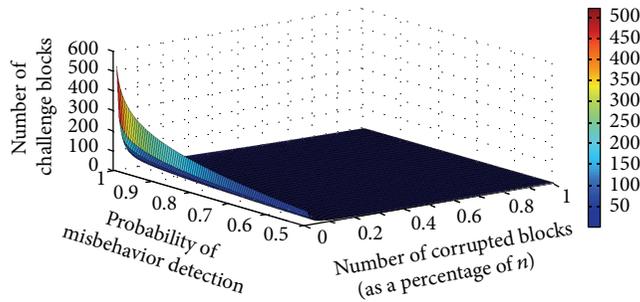


FIGURE 7: Number of required blocks as a challenge message under probability of misbehavior detection are from 0.5 to 1.

DCTs and modifying the content. Finally, to append to an operation, the auditor must inset a new data block after the last data block of the last DCT which imposes $O(C)$ as a computation cost.

We set up our own Eucalyptus private infrastructure as a service (IaaS) cloud in order to conduct this experiment using the existing IT infrastructure of center for mobile cloud computing (C4MCC). Eucalyptus is an acronym for “Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems” and is actually a Linux-based open-source software architecture that can be installed without modification on all major Linux operating systems such as RHEL, Centos, Ubuntu, and Debian. The reason why we choose eucalyptus is due to its compatibility with Amazon AWS APIs [38] which means that we can use Eucalyptus commands to manage Amazon or Eucalyptus instances and move freely between an Eucalyptus private cloud and the Amazon Public cloud making it a hybrid cloud. Secondly, Eucalyptus cloud computing architecture is highly scalable because of its distributed nature and is flexible enough to

support businesses of any size. Thirdly, it allows you to make your apps in-house on Eucalyptus and then migrate them to AWS; however, it was designed initially at the University of California, Santa Barbara, to support high performance computing (HPC) research [39]. The main components having their own Web-service interface that comprises our Eucalyptus installation are as follows.

- (1) Cloud controller (CLC) is actually the entry point into the cloud for administrators, managers, developers, and end-users and is accountable for satisfying the request of node managers. CLC is also responsible for making and implementing high level scheduling decisions with the help of cluster controllers.
- (2) Cluster controller (CC) generally executes on a computer system that has network connectivity to the systems running node controllers (NCs) and to the machine running the CLC. It actually manages a number of VMs and schedules their execution on particular NCs.
- (3) Node controller (NC) is executed on every system that is selected for hosting VM instances. It manages the life cycle of instances by making interaction with the OS and the hypervisor running on the same system and the CC.
- (4) Storage controller (SC) essentially implements block-accessed network storage such as EBS (Amazon Elastic Block Storage). Subsequently, it has the ability to send disk traffic across the local network to a remote storage site.
- (5) Walrus permits different users to store persistent data. It set access control policies for users to allow certain operations such as delete and create. Its interface is, however, compatible with Amazon’s S3 to store and access both the virtual machine images and user data. It is actually a file-level storage system while essentially representing a block-level storage system.

We calculated the signature on the basis of defining multiplication in $GF(2^l)$ as polynomial multiplication modulo a generator polynomial. The multiplication by the unknown X is carried out by a left shifting and XORing with a parameter corresponding to the generator polynomial. As a result, a γ can be identified with the unknown so that multiplication by γ includes a left shift operation followed by a conditional XOR. Broder [40] proposed a technique to perform several shift operations in one time, by creating a table consisting of a number of decisions that are used as the XOR-operand. In this simulation, we assume that the length of a bit string (l) is 16 bits and the length of each block is 8 KB. We also divide each of the blocks into equal bit strings to compute the algebraic signature of each block.

We conduct the experiments for updating an outsourced file (F) with length 1GB, including 125,000 data blocks, and demonstrate the efficiency of the proposed scheme in Figure 8, where the numbers of updated (inserted or deleted) blocks are increasing from 100 to 1000 with intervals of 8. To insert or delete a block in the Wang scheme, the auditor needs

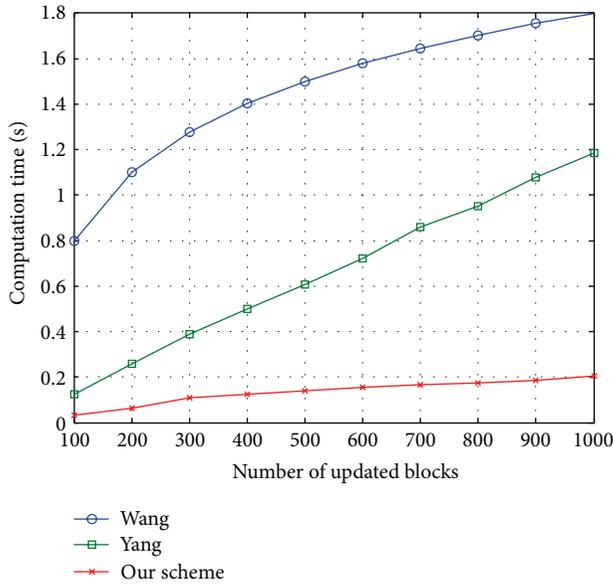


FIGURE 8: Comparison of computation cost under different number of update requests.

to find the position of the block (i) in the MHT tree. Moreover, inserting or deleting a block needs to recalculate the hash of the root each time that incurs the huge computation overhead on the auditor. Similarly, in the Yang method, after finding the position of the block (i), as a precondition, the auditor has to shift the remaining ($n-i$) blocks for every insert or delete operations. Subsequently, repeating this process multiple (100–1000) times results in a significant computation overhead on the auditor. The proposed method considers 10 DCTs with size 12500 instead of a single array with size 125000 in the Yang scheme. Consequently, the number of shifts reduced in our method results in the minimum computation overhead on the client side. Figure 8 shows the performance in terms of computation cost under different number of update (insert or delete) operations. The analysis of the results shows the efficiency of our scheme.

We also show the impact of the size of the file on the computation overhead of the auditor by Figure 9, where the DO updates the different size of outsource data by inserting or deleting 100 blocks in random positions, respectively, from 1 to 10 GB file. The computation overhead of the Wang method is dramatically increasing from 0.8 to approximately 2.3 by increasing the size of file because the auditor encounters a huge number of data block in MHT. Similarly, in the Yang scheme, when the size of input file is enhancing from 1 GB to 10 GB with the same size of data block (8 kB), a number of data blocks are also increasing. Consequently, the auditor requires shifting a huge number of blocks to insert or delete a data block. As it is shown in Figure 9, our method incurs the minimum overhead on the auditor (maximum 0.2 sec when the size of file is 10 GB) due to using 10 DCTs instead of one while applying the algebraic signature. Therefore, our method can be applicable for auditing large scale files dynamically.

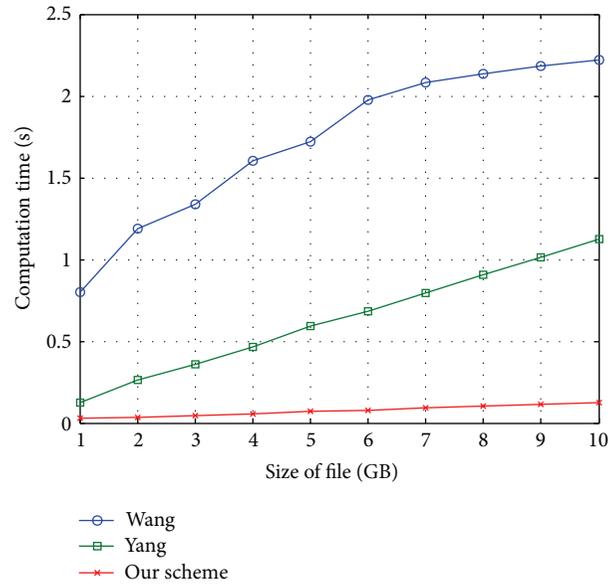


FIGURE 9: Comparison of computation cost under different file size from 1 GB to 10 GB when the number of update requests is 100.

Figures 8 and 9 clearly show the performance and efficiency of our scheme in terms of computation overhead. The comparative analysis shows that our scheme is more efficient than Wang and Yang schemes, respectively.

7. Conclusion

In this paper, we present an efficient remote data auditing scheme to ensure the data storage security in cloud computing. To achieve this goal, we employed algebraic signature properties that empower our scheme to verify the integrity of outsourced data and reduce the computation overhead on the client and server side of the cloud. We also design a new data structure, namely, divide and conquer table, to support dynamic data update, including insert, delete, append, and modify operations. The divide and conquer table also allows the verifier to audit the large scale data and perform a large number of insert and delete operations with minimum computation overhead on the verifier and server. The security and performance analysis shows the efficiency and provability of our scheme.

As a part of future work, we extend our scheme to find the optimized number of divisions in the divide and conquer table. We also improve our scheme to be applicable for distributed cloud servers.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was carried out as part of the Mobile Cloud Computing Research Project funded by the Malaysian Ministry of Higher Education under the High Impact Research Grant of University of Malaya, reference number UM.C/HIR/MOHE/FCSIT/03. This work was partly supported by the National Natural Science Foundation of China under Grant no. 61300220.

References

- [1] P. Mell and T. Grance, *The NIST Definition of Cloud Computing (Draft)*, NIST Special Publication 800, 2011.
- [2] Y.-J. Chen and L.-C. Wang, "A security framework of group location-based mobile applications in cloud computing," in *Proceeding of the International Conference on Parallel Processing Workshops (ICPPW '11)*, pp. 184–190, Taipei, Taiwan, September 2011.
- [3] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer Applications*, vol. 40, pp. 325–344, 2014.
- [4] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [5] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," *IEEE Network*, vol. 24, no. 4, pp. 19–24, 2010.
- [6] L. Wei, H. Zhu, Z. Cao et al., "Security and privacy for storage and computation in cloud computing," *Information Sciences*, vol. 258, pp. 371–386, 2014.
- [7] M. K. Khan and J. Zhang, "Multimodal face and fingerprint biometrics authentication on space-limited tokens," *Neurocomputing*, vol. 71, no. 13-15, pp. 3026–3031, 2008.
- [8] M. Khan and J. Zhang, "An efficient and practical fingerprint-based remote user authentication scheme with smart cards," in *Information Security Practice and Experience*, K. Chen, R. Deng, X. Lai, and J. Zhou, Eds., vol. 3903 of *Lecture Notes in Computer Science*, pp. 260–268, Springer, Berlin, Germany, 2006.
- [9] N. Gohring, Amazon's S3 down for several hours, 2008, <http://status.aws.amazon.com/s3-20080720.html>.
- [10] M. Arrington, "Gmail disaster: reports of mass email deletions," 2006, <http://techcrunch.com/2006/12/28/gmail-disaster-reports-of-mass-email-deletions/>.
- [11] Cellan-Jones, "The Sidekick Cloud Disaster," http://www.bbc.co.uk/blogs/technologycloud_disaster.html.
- [12] D. Storm, Epsilon breach: hack of the century?, 2011, http://blogs.computerworld.com/18079/epsilon_breach_hack_of_the_century.
- [13] M. J. Schwartz, "6 worst data breaches of 2011," 2012, <http://www.informationweek.com/news/security/attacks/232301079>.
- [14] G. Ateniese, R. Burns, R. Curtmola et al., "Remote data checking using provable data possession," *ACM Transactions on Information and System Security*, vol. 14, no. 1, article 12, 2011.
- [15] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 598–609, Virginia, Va, USA, November 2007.
- [16] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (SecureComm '08)*, pp. 1–10, ACM, Istanbul, Turkey, September 2008.
- [17] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1717–1726, 2012.
- [18] C. Erway, A. K p c , C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 213–222, ACM, Chicago, Ill, USA, November 2009.
- [19] Q.-A. Wang, C. Wang, K. Ren, W.-J. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2011.
- [20] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," *IEEE Transactions on Services Computing*, vol. 5, no. 2, pp. 220–232, 2012.
- [21] H. Wang, "Proxy provable data possession in public clouds," *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 551–559, 2012.
- [22] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.
- [23] Ł. Krzywiecki and M. Kutylowski, "Proof of possession for cloud storage via lagrangian interpolation techniques," in *Network and System Security*, L. Xu, E. Bertino, and Y. Mu, Eds., vol. 7645 of *Lecture Notes in Computer Science*, pp. 305–319, Springer, Berlin, Germany, 2012.
- [24] F. Seb , J. Domingo-Ferrer, A. Mart nez-Ballest , Y. Deswarte, and J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1034–1038, 2008.
- [25] A. Juels and B. S. Kaliski Jr., "Pors: proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 584–597, ACM, Alexandria, VA, USA, November 2007.
- [26] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Advances in Cryptology (ASIACRYPT)*, pp. 90–107, Springer, Berlin, Germany, 2008.
- [27] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Advances in Cryptology (EUROCRYPT)*, pp. 416–432, Springer, Berlin, Germany, 2003.
- [28] D. Cash, A. K p c , and D. Wichs, "Dynamic proofs of retrievability via oblivious RAM," *Cryptology ePrint Archive: Report 2012/550*, IACR, 2012.
- [29] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious RAMs," *Journal of the ACM*, vol. 43, no. 3, pp. 431–473, 1996.
- [30] W. Pugh, "Skip lists: a probabilistic alternative to balanced trees," *Communications of the ACM*, vol. 33, no. 6, pp. 668–676, 1990.
- [31] Y. Zhang and M. Blanton, "Efficient dynamic provable possession of remote data via update trees," *IACR Cryptology ePrint Archive*, vol. 2012, p. 291, 2012.
- [32] R. C. Merkle, "Protocols for public key cryptosystems," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 122–134, April 1980.

- [33] D. Catalano, D. Fiore, R. Gennaro, and K. Vamvourellis, "Algebraic (trapdoor) one-way functions and their applications," in *Theory of Cryptography*, A. Sahai, Ed., vol. 7785 of *Lecture Notes in Computer Science*, pp. 680–699, Springer, Berlin, Germany, 2013.
- [34] M. Sookhak, H. Talebian, E. Ahmed, A. Gani, and M. K. Khan, "A review on remote data auditing in single cloud server: taxonomy and open issues," *Journal of Network and Computer Applications*, 2014.
- [35] T. S. J. Schwarz and E. L. Miller, "Store, forget, and check: using algebraic signatures to check remotely administered storage," in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS '06)*, p. 12, 2006.
- [36] W. Litwin and T. Schwarz, "Algebraic signatures for scalable distributed data structures," in *Proceedings of the 20th International Conference on Data Engineering (ICDE '04)*, pp. 412–423, April 2004.
- [37] R. Impagliazzo, L. A. Levin, and M. Luby, "Pseudo-random generation from one-way functions," in *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing Seattle*, pp. 12–24, Washington, DC, USA, May 1989.
- [38] AWS, "Amazon Simple Storage Service (S3)," <http://aws.amazon.com/s3/>.
- [39] D. Nurmi, R. Wolski, C. Grzegorzczak et al., "The eucalyptus open-source cloud-computing system," in *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID '09)*, pp. 124–131, Shanghai, China, May 2009.
- [40] A. Broder, "Some applications of Rabin's fingerprinting method," in *Sequences II*, pp. 143–152, Springer, New York, NY, USA, 1993.

Research Article

Combining Digital Watermarking and Fingerprinting Techniques to Identify Copyrights for Color Images

Shang-Lin Hsieh,¹ Chun-Che Chen,^{1,2} and Wen-Shan Shen¹

¹ Department of Computer Science and Engineering, Tatung University, Taipei 10452, Taiwan

² Taipei College of Maritime Technology, New Taipei 25172, Taiwan

Correspondence should be addressed to Shang-Lin Hsieh; sunny6677@gmail.com

Received 15 March 2014; Accepted 20 May 2014; Published 8 July 2014

Academic Editor: Fei Yu

Copyright © 2014 Shang-Lin Hsieh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a copyright identification scheme for color images that takes advantage of the complementary nature of watermarking and fingerprinting. It utilizes an authentication logo and the extracted features of the host image to generate a fingerprint, which is then stored in a database and also embedded in the host image to produce a watermarked image. When a dispute over the copyright of a suspect image occurs, the image is first processed by watermarking. If the watermark can be retrieved from the suspect image, the copyright can then be confirmed; otherwise, the watermark then serves as the fingerprint and is processed by fingerprinting. If a match in the fingerprint database is found, then the suspect image will be considered a duplicated one. Because the proposed scheme utilizes both watermarking and fingerprinting, it is more robust than those that only adopt watermarking, and it can also obtain the preliminary result more quickly than those that only utilize fingerprinting. The experimental results show that when the watermarked image suffers slight attacks, watermarking alone is enough to identify the copyright. The results also show that when the watermarked image suffers heavy attacks that render watermarking incompetent, fingerprinting can successfully identify the copyright, hence demonstrating the effectiveness of the proposed scheme.

1. Introduction

Many researchers [1–17] have been engaged in finding the solution to protecting copyrights of digital images, which may be duplicated and distributed over the Internet without the authors' permission. Generally speaking, there are two approaches to discovering image copyright infringement. One is watermarking [1–11] and the other is fingerprinting [12–17]. The main idea of watermarking is to embed a piece of information (i.e., watermark) in the host image. If a similar watermark can be retrieved from a suspect image, it is then considered a duplicated one. On the other hand, the principle of fingerprinting is to extract unique features (i.e., fingerprints) from both the host image and the suspect one for comparison. If their fingerprints are similar, the ownership of the image can then be confirmed.

There are some general considerations on the two techniques, including the processing time and robustness. In terms of processing time, watermarking is more efficient

because fingerprinting needs extra time to compare the image's fingerprint with those stored in the database. If the database is large, it will be very time consuming. On the other hand, fingerprinting is generally more robust [18, 19] because when a watermarked image suffers some image processing operations that modify the content of the image, the embedded watermark will usually be damaged or even destroyed. On the contrary, since normal image processing does not destroy the features of an image, the fingerprint of the image can therefore be preserved. In summary, fingerprinting is more robust whereas watermarking is more efficient. If the complementary natures of two approaches can be utilized properly, a robust and efficient scheme can then be developed to identify copyrights.

This paper proposes a novel scheme that combines the two techniques to identify copyrights for color images. The proposed scheme generates from the image a fingerprint, which also serves as the watermark. The watermark is then embedded in the host image to produce a watermarked

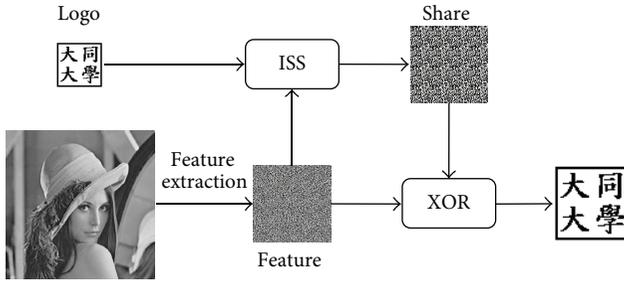


FIGURE 1: The process of the ISS.

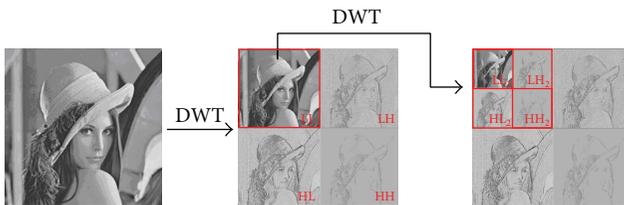


FIGURE 2: 2D DWT.

image. When there is a dispute over the copyright of a suspect image, the suspect image will first be processed by watermarking, which tries to retrieve the watermark from the suspect image. If the watermark is identified, the copyright is confirmed at this stage; otherwise, the image will then be processed by fingerprinting, which utilizes the retrieved watermark as the fingerprint and compares it with those stored in the database. If a match is found, then the suspect image will be considered a duplicated one.

2. Related Background

The proposed scheme utilizes a special technique called image secret sharing (ISS), whose details can be found in the paper [20] we published in 2008. The following briefly describes the main idea of the technique utilized by the proposed scheme.

The ISS generates a share image from two images. In the proposed scheme, the two images are the logo image and the feature image (as depicted in Figure 1). The logo image can be any identifiable image. The feature image is generated from the input image as follows. First, the input image is split into nonoverlapping 8×8 blocks. Then, the 2D DWT is applied to each block to generate four subbands, LL_2 , LH_2 , HL_2 , and HH_2 . An example of 2D DWT is shown in Figure 2. Finally, the coefficients in the LL_2 subband of each DWT block are used to generate the feature image. The ISS then generates a share image from the logo image and feature image. The share image will be used as the fingerprint of the input image by the scheme. The share image also serves as the watermark to be embedded in the host image. The benefit of the ISS scheme is that performing the XOR operation on the feature image and the share image will restore the logo image, which can then be used to identify the copyright.

3. The Proposed Copyright Identification Scheme

The proposed scheme contains two phases: the *fingerprint and watermarked image generation* phase and the *authentication logo detection* phase. The former phase extracts features from the host image, which, along with a logo image, is used to generate the fingerprint. The fingerprint also serves as the watermark, and the phase embeds it in the host image to produce a watermarked image. On the other hand, the latter phase extracts features and retrieves the watermark from the suspect image. The extracted features and the retrieved watermark are utilized to restore the logo image, which is used to identify the copyright. If it fails, the retrieved watermark then serves as the fingerprint and is compared with those in the database to determine if the suspect image is a duplicated one.

The *fingerprint and watermarked image generation* phase (shown in Figure 3) works as follows. In the beginning, *feature extraction* extracts the features of the host image and then *logo scrambling* disarranges the authentication logo to a scrambled logo image. After that, *fingerprint generation* takes as input the extracted features and the scrambled logo to generate the fingerprint. Finally, the fingerprint serves as a watermark and is embedded in the host image, which becomes a watermarked image. The fingerprint is also stored in a database for later use in the next phase.

The *authentication logo detection* phase (shown in Figure 3) checks the watermark first and, if necessary, the fingerprint next. In the beginning, *watermark retrieval* regains the watermark from the suspect image. Next, the features of the suspect image are extracted by *feature extraction*. After that, *logo restoration* takes as input the retrieved watermark (the expected fingerprint of the suspect image) and the extracted features to recover and rearrange the scrambled logo to restore the authentication logo. The phase ends if the accuracy rate of the restored logo determined by *logo comparison* is high enough; otherwise, the process proceeds to retrieve the next available fingerprint from the database and then returns to *logo restoration*, which takes as input the retrieved fingerprint instead of the extracted watermark. The phase restores the logo from the retrieved fingerprint as well as the extracted features and proceeds to *logo comparison*. The looping process continues until the authentication logo is discovered or no fingerprint is available.

3.1. Fingerprint and Watermarked Image Generation Phase. The following paragraphs detail the stages in the *fingerprint and watermarked image generation* phase, including *feature extraction*, *logo scrambling*, *fingerprint generation*, and *watermark embedding*.

3.1.1. Feature Extraction. The *feature extraction* stage takes a color image as input and then extracts its features. The stage has two substages, *sampling* and *feature generation*. During *sampling*, the stage first transforms the RGB image to the YCbCr color space [21, 22]. Then, it partitions each of the three channels into several nonoverlapping blocks of size 8

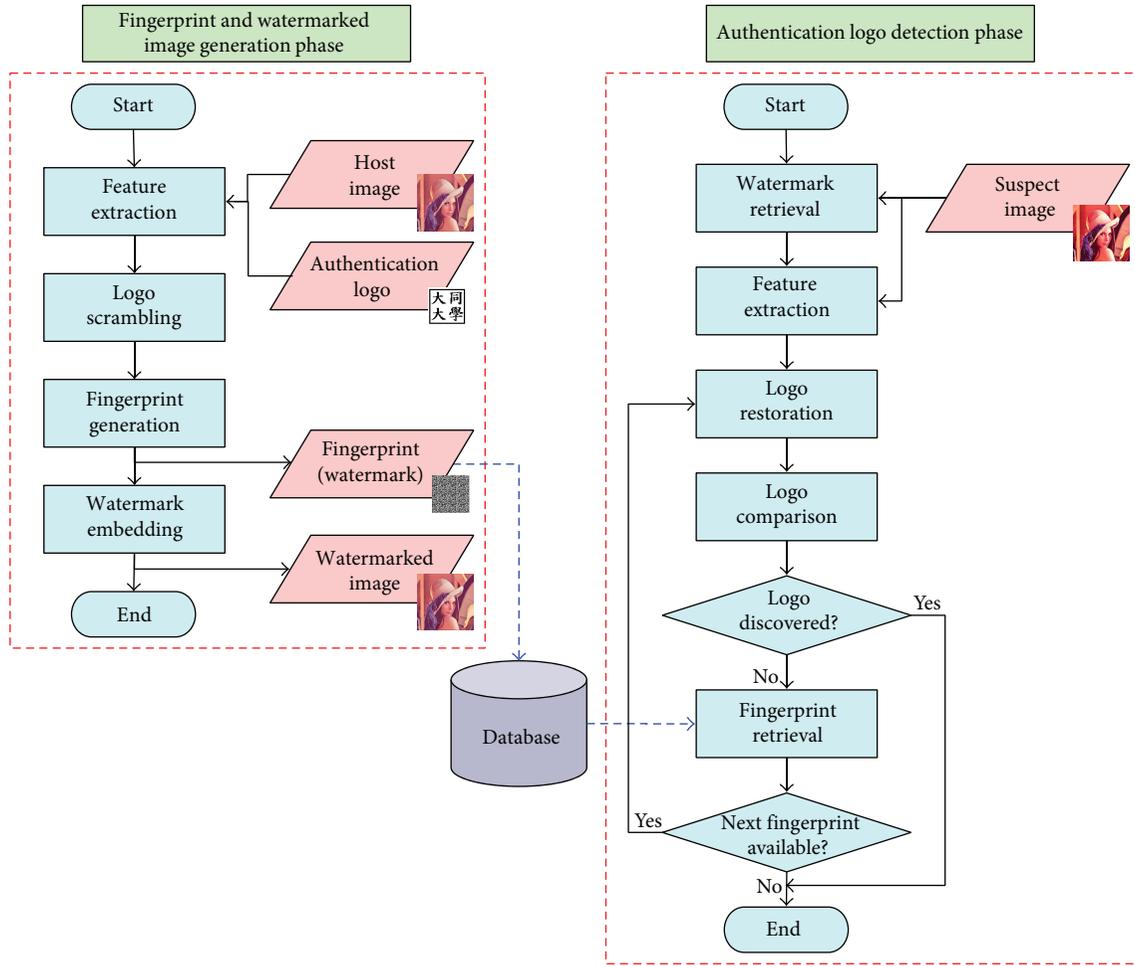


FIGURE 3: The phases of the proposed scheme.

$\times 8$ (hence, a color image of size $N \times N$ will result in $3 \times N/8 \times N/8$ nonoverlapping blocks). After partition, for each row of the corresponding blocks in the three channels, the stage takes the first four samples from the Y channel, the next two from the Cb, and the last two from the Cr (as depicted in Figure 4) to generate new packed blocks, each of size 8×8 .

After *sampling*, the stage enters its second substage, *feature generation*. For each of the packed blocks, the stage applies 2D DWT to the block, resulting in four coefficients in the LL_2 subband. Next, the stage computes the average (denoted by A) of the four coefficients and then obtains a feature type T according to the relationship of the four coefficients and the average A as expressed in (1). Consider

$$T = \begin{cases} 1, & \text{if only one coefficient is smaller than } A. \\ 2, & \text{if only two coefficients are smaller than } A. \\ 3, & \text{if only one coefficient is greater than } A. \\ 4, & \text{if all of the four coefficients are the} \\ & \text{same and hence all equal } A. \end{cases} \quad (1)$$

According to T , A , and the mapping table shown in Table 1, a feature share (called FT-share) of size 2×2 is

determined for each block. The FT-shares represent the features of the input color image. They are assembled to form the feature image.

The steps of the *feature extraction* stage are listed in Algorithm 1.

3.1.2. Logo Scrambling. In order to disperse the intensity of attacks, the proposed scheme adopts Torus automorphism [23] to scramble the authentication logo. The stage uses a predetermined key, k , and the following equation to scramble the logo. Consider

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \end{pmatrix} \bmod T, \quad (2)$$

where (x_t, y_t) are the coordinates in state t and T is the coordinate size of the given image. Figure 5 shows an example of the authentication logo scrambled four times with $k = 1, T = 128$.

3.1.3. Fingerprint Generation. The proposed scheme uses ISS (mentioned in Section 2) to generate the fingerprint. It determines a FP-share for each FT-share (generated in *feature*

Input: A color image $H (N \times N)$.
Output: A feature image FT ($N/8 \times N/8$).
 Convert H to the YCbCr color space
 Partition each of the Y, Cb, and Cr channels into $N/8 \times N/8$ non-overlapping blocks of size 8×8
 For each corresponding block of the three channels
 Take the first 4 samples from the Y channel, the next 2 from the Cb, and last 2 from the Cr as depicted in Figure 4 to form a packed block of size 8×8
 End For
 Apply 2D-DWT to each packed block to obtain $N/8 \times N/8$ LL_2 blocks of size 2×2
 For each of the LL_2 blocks
 Compute the average A of the four coefficients
 Obtain the feature type T according to (1)
 Determine the FT-share according to T, A , and Table 1
 End For
 Assemble the FT-shares to form the feature image FT

ALGORITHM 1: Feature extraction.

TABLE 1: The mapping table.

Feature type T	Mean value position	White logo pixel		FT-share XOR	Black logo pixel		FT-share XOR
		FT-share	FP-share	FP-share	FT-share	FP-share	FP-share
1	$a < A < b, c, d$						
	$b < A < a, c, d$						
	$c < A < a, b, d$						
	$d < A < a, b, c$						
2	$a, b \leq A < c, d$						
	$c, d \leq A < a, b$						
	$a, d \leq A < b, c$						
	$b, c \leq A < a, d$						
	$a, c \leq A < b, d$						
	$b, d \leq A < a, c$						
3	$b, c, d \leq A < a$						
	$a, c, d \leq A < b$						
	$a, b, d \leq A < c$						
	$a, b, c \leq A < d$						
4	$A = a = b = c = d$						



The coefficients of the LL_2 subband (a is at the top left position, b the top right, c the bottom left, and d the bottom right)

FT-share: feature share; FP-share: fingerprint share.

TABLE 2: Partial table of Table 1.

Feature type T	Mean value position	White logo pixel		FT-share XOR	Black logo pixel		FT-share XOR
		FT-share	FP-share	FP-share	FT-share	FP-share	FP-share
2	$b, c \leq A < a, d$						

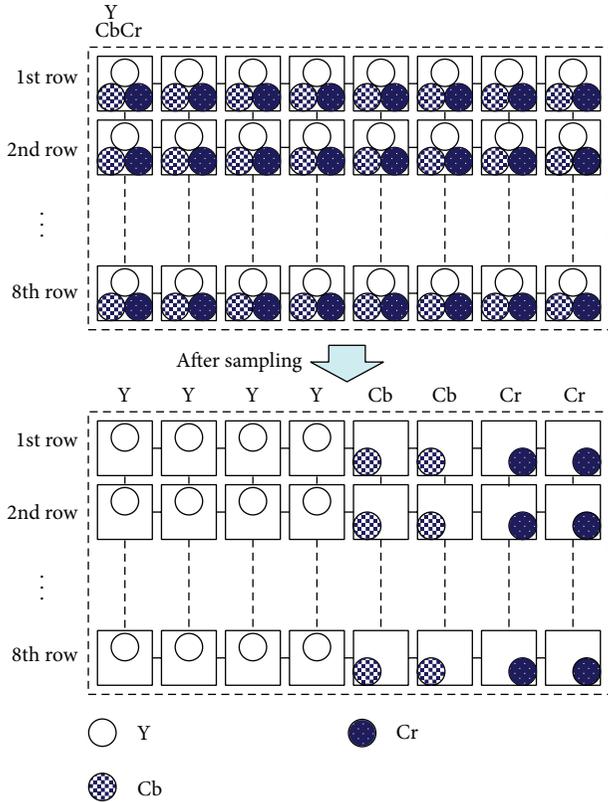


FIGURE 4: Illustration of *sampling*.

extraction) according to the color of the corresponding pixel in the scrambled logo by looking Table 1 up. For example, if the feature type is 2 and the FT-share is the same as the one in Table 2, then the FP-share will be either one of the two FP-shares in Table 2 according to the color of the corresponding pixel in the scrambled logo. After every FP-share is determined, the stage gathers all the FP-shares to form the fingerprint image. The fingerprint also serves as the watermark in the next stage.

3.1.4. *Watermark Embedding.* The watermark embedding stage uses the resulting fingerprint image as a watermark and then embeds it in the host image. Generally speaking, for the RGB color space, the human visual system is more sensitive to the G channel than to the other two [22–28]. Therefore, the proposed scheme embeds the watermark into the less

sensitive R and B channels of the host image. To be more specific, there are four areas: two in the R channel (I_1 and I_2) and two in the B channel (I_3 and I_4), used for *watermark embedding* (see Figure 6). The stage applies the 2D DWT to the R and B channels of the host image and next applies the 1D DWT to the resulting HL_2 and LH_2 to obtain the four blocks, $I_1, I_2, I_3,$ and I_4 .

The proposed scheme embeds the watermark into the image by adjusting the coefficients in the I_1 to I_4 blocks according to a predefined M . The value of M affects the robustness and the quality. The larger the M is, the more robust the embedded watermark is, but the worse the quality of the watermarked image is. Figure 7 illustrates the adjustment of the coefficients. If the watermark bit is 1 and the current coefficient is between $M \times (i - 1)$ and $M \times i$, then the coefficient will be adjusted to be $M \times (i - 1) - M/4$ or $M \times i - M/4$, whichever is closer to the current coefficient. On the other hand, the adjusted coefficient will be $M \times (i - 1) + M/4$ or $M \times i + M/4$ if the watermark bit is 0.

The embedding process is described as follows. First, the stage uses the current coefficient c and the predefined M to calculate S , sign , and (r_0, r_1) according to (3), (4), and (5), respectively. In the equations, sign indicates that the c is positive or negative; r_0 is the remainder for watermark bit value 0 and, similarly, r_1 for watermark bit value 1. Then, the stage determines the C_{Low} and C_{High} according to the value of the watermark bit. If the value is 0, (6) will be used, otherwise, (7). Finally, the stage adjusts the coefficient c to c' according to (8). Consider

$$S = \frac{|c|}{M}, \tag{3}$$

$$\text{sign} = \begin{cases} 1, & \text{if } c \geq 0, \\ -1, & \text{otherwise,} \end{cases} \tag{4}$$

$$(r_0, r_1) = \begin{cases} \left(\frac{M}{4}, \frac{3M}{4} \right), & \text{if } c \geq 0, \\ \left(\frac{3M}{4}, \frac{M}{4} \right), & \text{otherwise,} \end{cases} \tag{5}$$

$$C_{\text{Low}} = \text{sign} \times (S \times M + r_0), \tag{6}$$

$$C_{\text{High}} = \text{sign} \times ((S + \text{sign}) \times M + r_0),$$

$$C_{\text{Low}} = \text{sign} \times ((S - \text{sign}) \times M + r_1), \tag{7}$$

$$C_{\text{High}} = \text{sign} \times (S \times M + r_1),$$

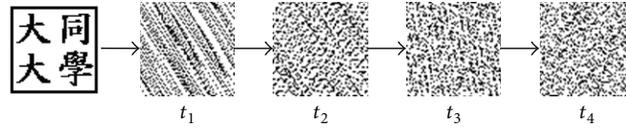


FIGURE 5: Example of scrambling.

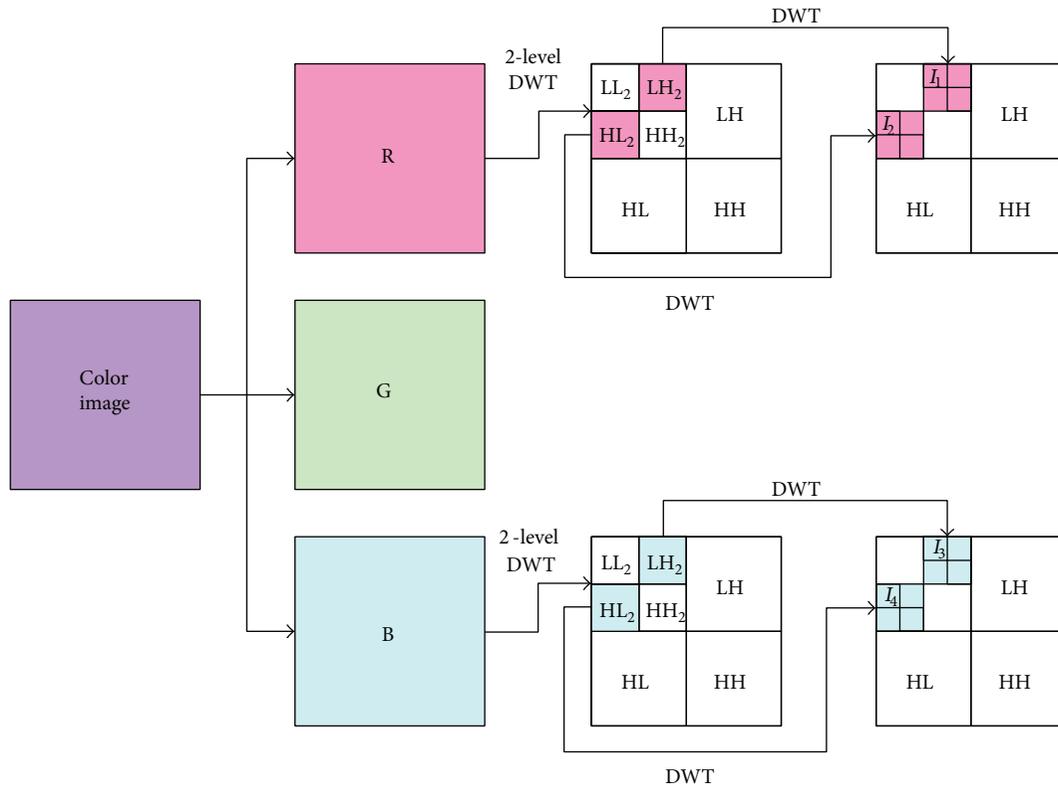


FIGURE 6: The $I_1, I_2, I_3,$ and I_4 blocks used for watermark embedding.

$$c' = \begin{cases} C_Low, & \text{if } |C_Low - c| \leq |C_High - c| \\ C_High, & \text{otherwise.} \end{cases} \quad (8)$$

of the I_1-I_4 blocks, it obtains the watermark bit w according to

$$w = \begin{cases} 1, & \text{if } \left(c \geq 0, c \bmod M \geq \frac{M}{2} \right), \\ & \text{or } \left(c < 0, |c| \bmod M < \frac{M}{2} \right), \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

The steps of the watermark embedding stage are listed in Algorithm 2.

3.2. Authentication Logo Detection Phase. The phase is activated when a dispute over the copyright of a suspect image occurs. The following details the stages in the phase, including watermark retrieval and logo restoration.

3.2.1. Watermark Retrieval. The stage regains the watermark from the suspect image. First, the stage applies the DWT to the R and B planes of the suspect image in the same way as that in watermark embedding to obtain the embedding blocks, I_1 to I_4 (refer to Figure 6). Then, for each coefficient c

where M is the same as that in (3).

Finally, the watermark (which is supposed to be the fingerprint of the image) is restored by assembling every w for each coefficient c in the I_1-I_4 blocks.

Algorithm 3 lists the steps of the watermark retrieval stage.

3.2.2. Logo Restoration. The stage restores the authentication logo. As shown in Figure 8, it has four substages: scrambled

Input: A color image $H (N \times N)$ and a fingerprint image $FP (N/4 \times N/4)$.
Output: A watermarked image $(N \times N)$.
 Apply DWT to the R and B channels of H to obtain the four embedding blocks I_1, \dots, I_4
 For each coefficient c of the I_1-I_4 blocks and the corresponding pixel of FP
 Compute the $sign, S,$ and (r_0, r_1) by (3), (4), and (5), respectively
 Compute the C_Low and C_High by (6) if watermark bit is 0, or (7) if otherwise
 Adjust the coefficient to c' by (8)
 End For
 Perform inverse DWT to produce the watermarked image

ALGORITHM 2: Watermark embedding.

Input: A suspect color image $S (N \times N)$.
Output: A fingerprint (watermarked) image $FP' (N/4 \times N/4)$.
 Apply DWT to the R and B channels of S and obtain the embedding blocks I_1-I_4
 For each coefficient c of the I_1-I_4 blocks
 Obtain w according to (9)
 End For
 Assemble every watermark bit w to restore the fingerprint image FP'

ALGORITHM 3: Watermark retrieval.

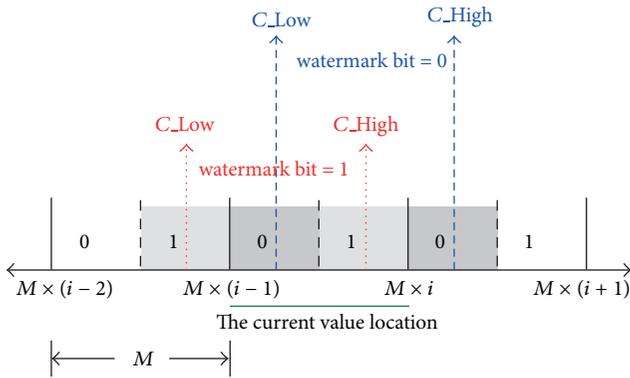


FIGURE 7: Coefficient adjustment.

logo recovery, logo unscrambling, logo enhancement, and logo resizing.

Scrambled Logo Recovery. After retrieving the fingerprint image (i.e., watermark) by *watermark retrieval* and extracting the feature image by *feature extraction*, the substage performs XOR operation on each pixel of the fingerprint image and the corresponding pixel of the feature image to retrieve the scrambled logo. Because both of the two images are black-and-white, each pixel is either 0 (black) or 1 (white). Therefore, the substage simply performs bitwise XOR operation on the two images and obtains a scrambled logo.

Logo Unscrambling. As mentioned in Section 3.1.2, the logo was scrambled before it is used to generate fingerprint in the *fingerprint and watermarked image generation* phase. The

substage adopts (10), which is the inverse equation of (2), to rearrange the scrambled logo and restore the logo. Consider

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}^{-1} \begin{pmatrix} x_{t-1} \\ y_{t-1} \end{pmatrix} \pmod T. \quad (10)$$

Logo Enhancement. The substage enhances the restored logo by erosion and dilation. Erosion removes pixels on object boundaries in an image and therefore can be used to remove smaller islands in the image; dilation, on the other hand, adds pixels to the boundaries of objects in an image and hence can be used to remove bright areas from the image. The substage performs erosion followed by dilation once, which is illustrated in Figure 9.

Erosion is performed by resetting each of the pixels according to (11). That is, if one of the neighbors is black, then the current pixel will be set to black; otherwise, it will be set to white. Let $L(x, y)$ be the current pixel value to be determined and $L(x+1, y), L(x+1, y+1),$ and $L(x, y+1)$ its neighbors; then

$$L(x, y) = \begin{cases} 0 \text{ (black)}, & \text{if } L(x+1, y) \text{ or } L(x+1, y+1) \\ & \text{or } L(x, y+1) \text{ is black} \\ 1 \text{ (white)}, & \text{otherwise.} \end{cases} \quad (11)$$

Dilation, on the contrary, resets the neighbors of each pixel rather than the pixel itself. If $L(x, y)$ is black; the substage sets all of its neighbors, $L(x+1, y), L(x+1, y+1),$ and $L(x, y+1),$ to black; otherwise, the neighbors remain unchanged.

Logo Resizing. The proposed scheme adopts ISS to retrieve the authentication logo, which causes *pixel expansion* because one logo pixel is mapped to a share of four pixels (mentioned

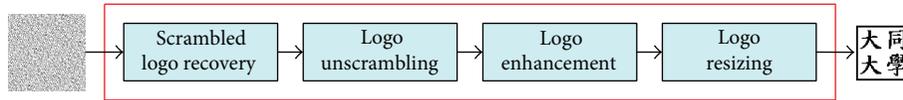


FIGURE 8: The four substages of logo restoration.

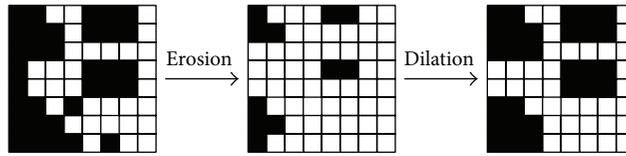


FIGURE 9: An example of erosion and dilation.



FIGURE 10: The authentication logo (64 × 64).

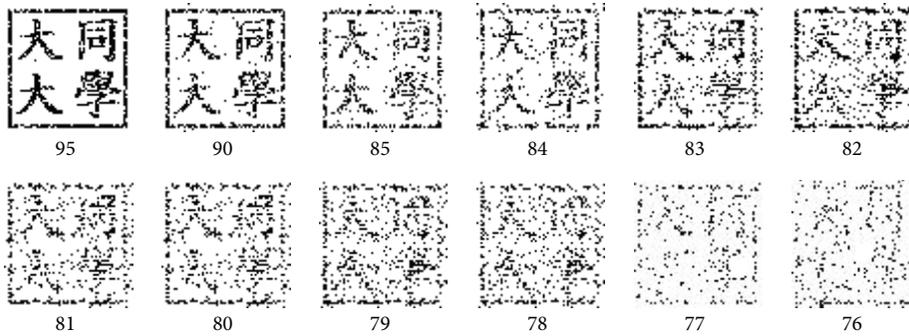


FIGURE 11: The restored logo with their AR values (%).

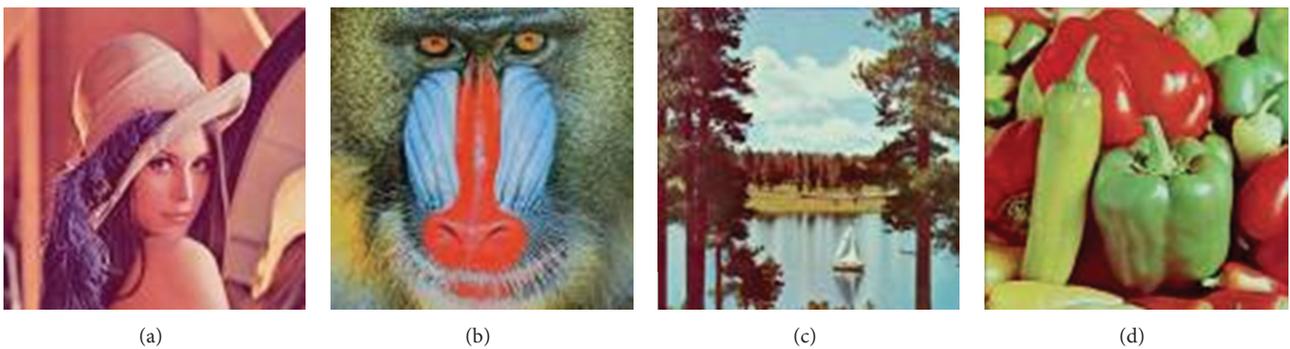


FIGURE 12: The test images used in the experiments: (a) Lena, (b) Mandrill, (c) Sailboat, and (d) Peppers.

in Section 3.1.3). As a result, the retrieved logo from the previous stage will be larger than the original one. To resize the logo to its original size, the substage partitions the

enhanced logo into several blocks of size 2×2 , each of which is then reduced into one pixel with value $L(x, y)$ according to the following rule:

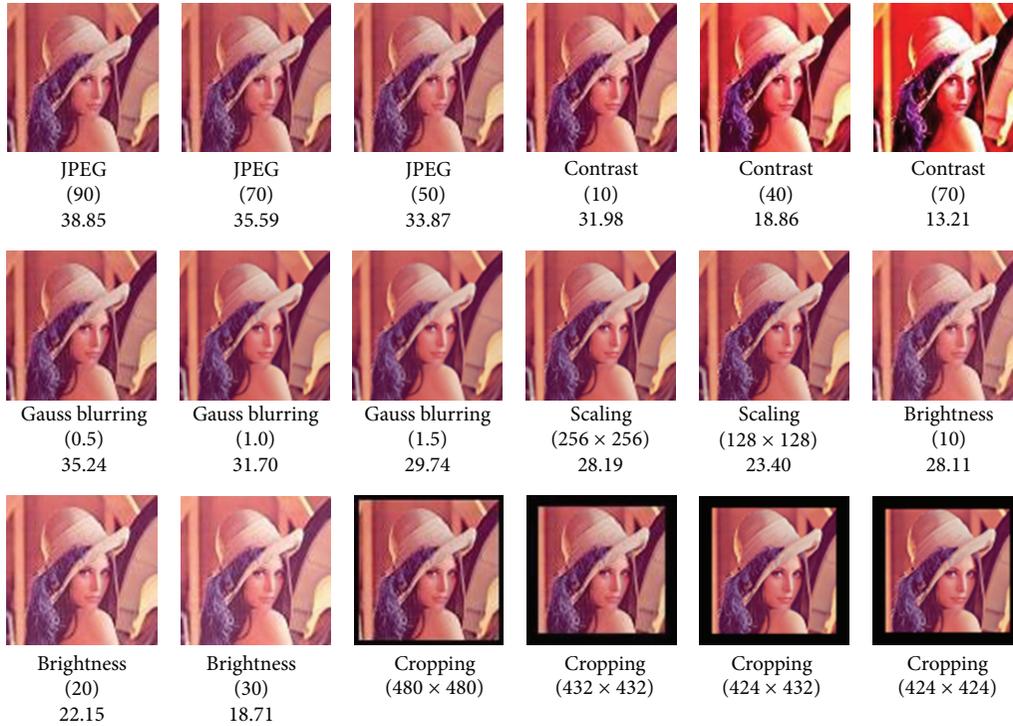


FIGURE 13: Some examples of "Lena" after different image attacks (PSNRs are listed in the last rows).

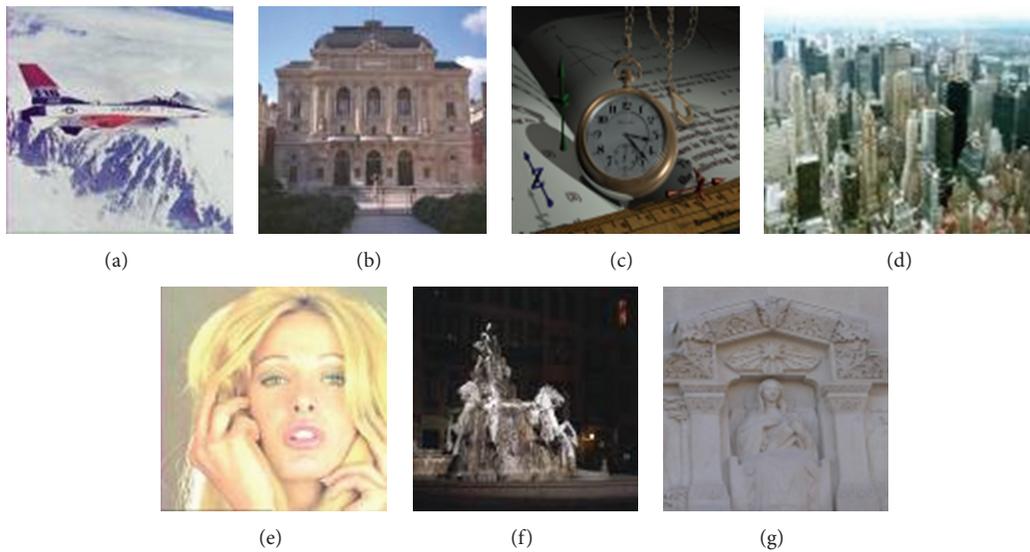


FIGURE 14: The other seven color images used for testing uniqueness.

$$L(x, y) = \begin{cases} 0 \text{ (black),} & \text{the number of the black pixels} \\ & \text{in each block} \geq 3, \\ 1 \text{ (white),} & \text{otherwise.} \end{cases} \quad (12)$$

4. Experimental Results

Two kinds of experiments were conducted to prove the effectiveness of the proposed scheme. The first experiment shows

the robustness of our scheme and the other demonstrates the capability of unique identification. In the experiments, the authentication logo used to generate the watermark (fingerprint) is shown in Figure 10.

Two common measurements used to estimate the robustness of our scheme are described as below.

(1) *Peak Signal to Noise Ratio (PSNR)*. The measurement to estimate the color image quality after image processing is a

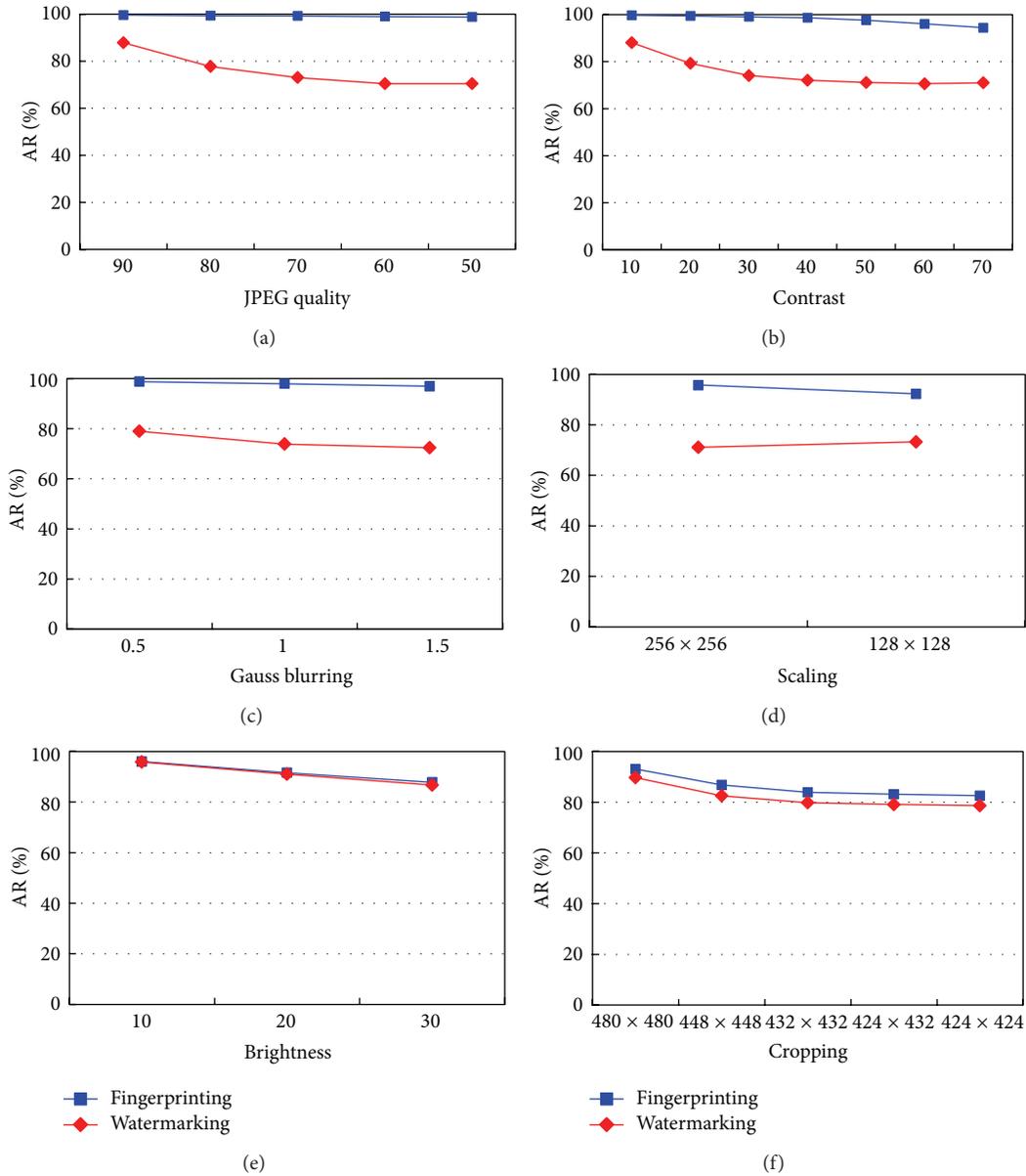


FIGURE 15: The average AR values of test images: (a) JPEG quality, (b) contrast, (c) Gauss blurring, (d) scaling, (e) brightness, and (f) cropping.

variant version of normal PSNR [29]. The variant PSNR listed below does not consider the influence of the green channel because the channel is not modified by our scheme. Consider

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{(\text{MSE}(R) + \text{MSE}(B)) / 2} \text{ dB}, \quad (13)$$

where MSE is the *mean square error* between the original image and the modified image, which is defined as follows:

$$\text{MSE} = \frac{\sum_{i=1}^N \sum_{j=1}^N (x_{ij} - x'_{ij})^2}{N^2}, \quad (14)$$

where x_{ij} represents the original pixel value and x'_{ij} denotes the modified pixel value.

According to the definition of PSNR, the higher the value is, the better the quality of the modified image is. Generally, if the PSNR is greater than 30 dB, the quality of the modified image is acceptable.

(2) *Accuracy Rate (AR)*. The measurement shown below is used to evaluate the correctness of the logo after it has been restored. Consider

$$\text{AR}(\%) = \frac{\text{CP}}{\text{NP}} \times 100, \quad (15)$$

where NP is the number of pixels in the original logo and CP is the number of correct pixels obtained by comparing the pixels of the original logo with the corresponding ones of the restored logo. Figure 11 shows the restored logos with

TABLE 3: The resulting AR values of the test images.

Attacks	AR (%) (Watermarking/fingerprinting)					
	Lena	Mandrill	Sailboat	Peppers	Average	
None	99.93/99.93	99.76/99.76	99.98/99.98	99.63/99.63	99.83/99.83	
JPEG	(90)	88.31/99.71	86.84/99.44	88.04/99.88	88.16/99.49	87.84/99.63
	(80)	79.76/99.73	76.17/99.02	77.29/99.58	77.91/99.12	77.78/99.36
	(70)	74.58/99.46	72.12/98.73	72.61/99.54	73.05/99.17	73.09/99.23
	(60)	71.58/99.63	70.14/98.1	70.19/99.44	69.95/98.83	70.47/99.00
	(50)	71.48/99.41	70.09/97.68	70.43/99.24	69.8/98.88	70.45/98.80
Contrast	(10)	92.6/99.85	80.71/99.78	88.62/99.95	90.31/99.27	88.06/99.71
	(20)	82.86/99.76	73.12/99.56	79.27/99.78	81.81/98.68	79.27/99.45
	(30)	75.85/99.46	70.61/99.07	75.22/99.63	74.63/98.07	74.08/99.06
	(40)	72.61/99.27	70.83/97.92	72.31/99.19	72.51/98.14	72.07/98.63
	(50)	71.7/98.93	70.41/96.19	71.19/98.75	71.09/96.58	71.10/97.61
	(60)	70.39/97.68	70.65/92.7	70.95/98.1	70.53/95.8	70.63/96.07
	(70)	70.8/94.65	70.65/90.31	71.12/97.71	71.36/95	70.98/94.42
Gauss blurring	(0.5)	83.64/99.41	73.44/97.29	78.17/99.17	80.54/99.05	78.95/98.73
	(1.0)	76.1/98.88	70.92/95.78	74.17/98.44	73.88/98.34	73.77/97.86
	(1.5)	73.63/98.17	70.95/94.34	72.44/97.29	72.34/97.73	72.34/96.88
Scaling	w: 256, h: 256	71.66/97.92	70.75/92.53	71.22/96.83	70.73/95.8	71.09/95.77
	w: 128, h: 128	72.83/96.02	73.41/86.04	73.49/94.04	73.41/92.99	73.29/92.27
Brightness	(10)	96.66/96.78	94.41/94.92	97.85/97.85	94.46/94.46	95.85/96.00
	(20)	90.65/91.48	89.06/90.7	95.9/95.9	88.33/88.33	90.99/91.60
	(30)	85.72/87.62	83.98/86.43	93.55/93.58	83.69/83.64	86.74/87.82
Cropping	w: 480, h: 480	89.62/92.92	88.99/92.46	91.48/95.04	88.77/91.87	89.72/93.07
	w: 448, h: 448	82.71/86.82	81.71/85.67	83.79/88.89	82.03/85.82	82.56/86.80
	w: 432, h: 432	79.88/83.08	79.25/83.01	81.13/86.23	79/83.3	79.82/83.91
	w: 424, h: 432	78.88/81.76	78.49/81.96	80.52/85.86	78.59/83.06	79.12/83.16
	w: 424, h: 424	78.22/81.2	78/81.49	80.25/85.33	78.1/82.42	78.64/82.61

different AR values. As can be seen, restored logos with AR higher than 81% (the ones in the upper row) still can be visually recognized whereas the restored logo with AR equal to 76% is hard to identify. However, according to the description in Section 4.2, if AR is higher than 75%, the scheme still can identify the copyrights of the image.

4.1. Robustness Experiments. The experiments proved our scheme is robust to different kinds of attacks. The test images used in the experiment, including “Lena,” “Mandrill,” “Sailboat,” and “Peppers,” are shown in Figure 12. The commercial image processing software “Adobe Image Photoshop CS” was used to simulate several kinds of image attacks, some of which for “Lena” are shown in Figure 13.

The experimental results of the test images are shown in Table 3. As mentioned above, if AR is less than 75%, the scheme cannot identify the copyrights of the image. Table 3 shows that our scheme failed to verify the copyrights for the images suffering the heavier attacks of JPEG, contrast, Gauss blurring, and scaling in *watermark verification*. Nevertheless, the duplications of those images can all be determined in *fingerprint verification*. In summary, our scheme can identify the copyrights of the suspect images under moderate attacks

in *watermark verification* and determine the duplications of those suffering the heavy attacks in *fingerprint verification*.

Moreover, there were 100 attacks in total and 53 of them resulted in an AR value higher than 75% in *watermarking verification*. That is to say, the copyrights of the 53% of the attacked images can be successfully identified in *watermark verification*, and hence only 47% of them need *fingerprint verification*.

4.2. Uniqueness Experiments. The experiment showed that our scheme has the capability of unique identification and is able to distinguish a copyrighted image from different ones. The four copyrighted images (with embedded watermarks) in Figure 12 along with seven unwatermarked images (Figures 14(a)–14(g)) were processed by our scheme to identify the copyright. The stored fingerprints of the watermarked images were used in *fingerprint verification* to restore the logos for all of the images.

The results shown in Table 4 demonstrated the extraordinary unique identification capability of our scheme. It can be clearly seen that all the restored logos except the ones of the copyrighted images (those on the rightmost side) are unrecognizable, which proves that our scheme is actually able to distinguish a copyrighted image from different ones. It will

TABLE 4: Experimental results of the uniqueness experiment.

(a) Lena											
	Mandrill	Sailboat	Peppers	F-16	Opera	Watch	New York	Tiffany	Terraux	Fourviere	Lena
Watermark											
AR (%)	73.07	73.69	73.07	69.80	70.39	70.09	71.34	70.26	69.78	70.26	99.93
Fingerprint											
AR (%)	59.06	56.03	49.24	62.72	59.62	44.34	62.35	60.40	45.65	63.21	99.93
(b) Mandrill											
	Lena	Sailboat	Peppers	F-16	Opera	Watch	New York	Tiffany	Terraux	Fourviere	Mandrill
Watermark											
AR (%)	73.74	73.69	73.07	72.73	73.52	73.79	72.64	74.35	73.82	72.2	99.76
Fingerprint											
AR (%)	58.2	57.81	62.08	57.84	59.45	60.03	59.64	54.35	59.01	60.38	99.76
(c) Sailboat											
	Lena	Mandrill	Peppers	F-16	Opera	Watch	New York	Tiffany	Terraux	Fourviere	Sailboat
Watermark											
AR (%)	73.74	73.07	73.07	72.73	73.52	73.79	72.64	74.35	73.82	72.2	99.98
Fingerprint											
AR (%)	54.32	59.06	53.44	57.91	55.1	52.37	56.47	55.37	53.78	59.4	99.98
(d) Peppers											
	Lena	Mandrill	Sailboat	F-16	Opera	Watch	New York	Tiffany	Terraux	Fourviere	Peppers
Watermark											
AR (%)	73.74	73.07	73.69	72.73	73.52	73.79	72.64	74.35	73.82	72.2	99.63
Fingerprint											
AR (%)	47.02	60.13	46.08	55.59	58.69	49.29	53.26	52.8	54.66	55.77	99.63

not mistakenly identify the copyrights of an unwatermarked image.

Moreover, the resulting AR values of all the unwatermarked images are all lower than 75%. Hence, it is reasonable for our scheme to confirm the copyright when AR is higher than 75%.

4.3. Discussion and Comparison. The fingerprint extracted from an image is more robust than the watermark embedded in the image. This can be seen from Figure 15, which shows the average AR values of the images suffering different attacks for watermarking and fingerprinting. The AR values for fingerprinting are all higher than those for watermarking.

TABLE 5: The processing time (ms) for watermark extraction and logo restoration and comparison.

Stage	Lena	Mandrill	Sailboat	Peppers	Average
Watermark extraction	218	209	200	213	210
Logo restoration and comparison	20	29	39	28	29
Watermark extraction + logo restoration and comparison	238	238	239	241	239

TABLE 6: Comparison of our combined scheme and the other two pure schemes.

	Pure watermarking scheme	Pure fingerprinting scheme	Our combined scheme
Robustness	△	○	○
Efficiency	○	△	○

*○: good; △: fair.

When the image undergoes image processing operations that heavily damage the embedded watermark, the extracted fingerprint still can survive the attacks. However, because linear comparison is a computationally intensive process, fingerprinting is costly in time if there are many fingerprints in the database.

Table 5 shows the processing time for *watermark extraction* and *logo restoration and comparison* in our scheme. The experiments were carried out on a computer equipped with the following hardware and software:

CPU: 3.16 GHz Intel(R) Xeon(R) CPU E3120,

RAM: 4 GB,

OS: Windows 7,

Computing language and environment: MATLAB.

As Table 5 shows, it takes about 0.21 seconds to extract the watermark from the input image and 0.029 seconds to restore and compare the logo. Therefore, the total processing time T_p (in seconds) of our scheme can be expressed by

$$T_p = 0.24 + 0.029 \times n, \quad (16)$$

where n represents the number of the fingerprints retrieved from the database. If the copyright can be identified in *watermark verification*, n is 0. That is, our scheme only needs 0.24 seconds in the best case. Otherwise, our scheme needs additional 0.029 seconds for each retrieved fingerprint in the database.

Because our scheme combines watermarking and fingerprinting techniques, it can be as efficient as pure watermarking schemes and also as robust as pure watermarking schemes. A pure watermarking scheme is very efficient because it only needs to make one watermark comparison to verify the copyright. However, it may not be as robust as a pure fingerprinting scheme when dealing with images that have suffered heavy attacks. Table 6 shows the comparison

of our combined scheme and the other two pure schemes. If the copyright of the input image can be identified by *watermarking verification*, our scheme can be as efficient as pure watermarking schemes. If *watermarking verification* fails to identify the copyright, our scheme is still able to determine duplication in *fingerprinting verification*, which makes our scheme as robust as pure fingerprinting schemes.

5. Conclusion

This paper presented a copyright identification scheme that takes advantage of the complementary nature of digital watermarking and fingerprinting. The experimental results showed that when the watermarked image suffers moderate attacks, *watermarking verification* alone is enough to identify the copyright, and there is no need for *fingerprinting verification*. In other words, the proposed scheme can identify the copyright efficiently in this situation. On the other hand, the experimental results also showed that when the watermarked image suffers heavy attacks that render *watermarking verification* incompetent, *fingerprinting verification*, although more time consuming, can successfully determine the duplication, hence demonstrating the robustness of the proposed scheme.

One distinguishing characteristic of the proposed scheme is that it does not need a separate watermark for *watermarking verification* and a separate fingerprint for *fingerprinting verification*. The proposed scheme extracts features from the input image to generate the fingerprint, which also serves as the watermark. Hence, only one piece of information is needed for both *watermarking* and *fingerprinting verifications*.

To further improve the scheme, retrieving the stored fingerprint image from the database to restore the correct authentication logo more quickly is worth studying. When there are more than one fingerprint image in the database, the original fingerprint of the host image must be correctly retrieved; otherwise, the correct authentication logo cannot be restored. Retrieving every stored fingerprint image to restore an authentication logo for comparison is very time consuming. The scheme should provide a more efficient way that is able to find the proper one in less time, which is the future work of the research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

Financial support of this study by Tatung University, Taipei, Taiwan, under Grant B100-I07-036 is gratefully acknowledged.

References

- [1] G. Coatrieux, H. Huang, H. Shu, and L. Luo, "A watermarking-based medical image integrity control system and an image

- moment signature for tampering characterization," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 6, pp. 1057–1067, 2013.
- [2] K.-C. Liu, "Colour image watermarking for tamper proofing and pattern-based recovery," *IET Image Processing*, vol. 6, no. 5, pp. 445–454, 2012.
- [3] Y.-L. Chen and C.-T. Hsu, "Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 396–406, 2011.
- [4] P.-C. Su, Y.-C. Chang, and C.-Y. Wu, "Geometrically resilient digital image watermarking by using interest point extraction and extended pilot signals," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 1897–1908, 2013.
- [5] A. Piper and R. Safavi-Naini, "Scalable fragile watermarking for image authentication," *IET Information Security*, vol. 7, no. 4, pp. 300–311, 2013.
- [6] M. Pawlak and Y. Xin, "Robust image watermarking: an invariant domain approach," in *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 885–888, May 2002.
- [7] W.-H. Lin, S.-J. Horng, T.-W. Kao, P. Fan, C.-L. Lee, and Y. Pan, "An efficient watermarking method based on significant difference of wavelet coefficient quantization," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 746–757, 2008.
- [8] I.-K. Kong and C.-M. Pun, "Digital image watermarking with blind detection for copyright verification," in *Proceedings of the 1st International Congress on Image and Signal Processing (CISP '08)*, vol. 1, pp. 504–508, May 2008.
- [9] L. A. Elrefaey, M. E. Allam, H. A. Kader, and M. Selim, "Robust blind image-adaptive watermarking," in *Proceedings of the 25th National Radio Science Conference (NRSC '08)*, March 2008.
- [10] M.-H. Lee and O.-J. Kwon, "Color image watermarking based on DS-CDMA using Hadamard kernel," in *Proceedings of the 10th International Conference on Advanced Communication Technology*, pp. 1592–1597, February 2008.
- [11] N.-Y. Lee and C.-C. Wang, "Yet another wavelet watermarking scheme for copyright protection," in *Proceedings of the 9th IEEE International Conference on E-Commerce Technology*, pp. 421–424, July 2007.
- [12] H.-M. Sun, C.-J. Hong, and C.-H. Chen, "A new approach to feature-based copyright protection of images," in *Proceedings of the 3rd International Conference on Information Technology: Research and Education (ITRE '05)*, pp. 233–237, June 2005.
- [13] C.-C. Chang and J.-C. Chuang, "An image intellectual property protection scheme for gray-level images using visual secret sharing strategy," *Pattern Recognition Letters*, vol. 23, no. 8, pp. 931–941, 2002.
- [14] C.-S. Lu, C.-Y. Hsut, S.-W. Sun, and P.-C. Chang, "Robust mesh-based hashing for copy detection and tracing of images," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04)*, vol. 1, pp. 731–734, June 2004.
- [15] J.-S. Lee and K.-S. Yoon, "The system integration of DRM and fingerprinting," in *Proceedings of the 8th International Conference Advanced Communication Technology (ICACT '06)*, vol. 3, pp. 2180–2183, February 2006.
- [16] S.-H. Yang and C.-F. Chen, "Robust image hashing based on SPIHT" in *Proceedings of the 3rd International Conference on Information Technology: Research and Education (ITRE '05)*, pp. 110–114, June 2005.
- [17] F. Ahmed and M. Y. Siyal, "A secure and robust DCT-based hashing scheme for image authentication," in *Proceedings of the 10th IEEE Singapore International Conference on Communication Systems (ICCS '06)*, pp. 1–6, October 2006.
- [18] J. S. Seo, J. Haitsma, T. Kalker, and C. D. Yoo, "A robust image fingerprinting system using the Radon transform," *Signal Processing: Image Communication*, vol. 19, no. 4, pp. 325–339, 2004.
- [19] P. C. Vila, *Content-based audio search: from fingerprinting to semantic audio retrieval [Ph.D. thesis]*, University Pompeu Fabra, Barcelona, Spain, 2007.
- [20] S.-L. Hsieh, I.-J. Tsai, B.-Y. Huang, and J.-J. Jian, "Protecting copyrights of color images using a watermarking scheme based on secret sharing and wavelet transform," *Journal of Multimedia*, vol. 3, no. 4, pp. 42–49, 2008.
- [21] V. NABIYEV and A. GÜNAY, "Towards a biometric purpose image filter according to skin detection," in *Proceedings of the Second International Conference on Problems of Cybernetics and Informatics*, 2008.
- [22] G. Poynton, "Frequently asked questions about color," 2014, <http://www.poynton.com/ColorFAQ.html>.
- [23] G. Voyatzis and I. Pitas, "Applications of torus automorphisms in image watermarking," in *Proceedings of International Conference on Image Processing*, vol. 3, pp. 237–240, 1996.
- [24] G. Voyatzis and I. Pitas, "Chaotic mixing of digital images and applications to watermarking," in *Proceeding of European Conference on Multimedia Applications, Services and Techniques (ECMAST '96)*, vol. 2, May 1996.
- [25] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, New York, NY, USA, 2nd edition, 2002.
- [26] E. J. Stollnitz, T. D. DeRose, and D. H. Salestin, "Wavelets for computer graphics: a primer, part 1," *IEEE Computer Graphics and Applications*, vol. 15, no. 3, pp. 76–84, 1995.
- [27] KeyLawk, "Colors: the human eye is most sensitive to GREEN," 2014, <http://keylawk.blogspot.com/2007/10/colors-human-eye-is-most-sensitive-to.html>.
- [28] S. Zhao, "Wavelength of maximum human visual sensitivity," 2014, <http://hypertextbook.com/facts/2007/SusanZhao.shtml>.
- [29] I. Nasir, W. Ying, and J. Jianmin, "Novel multiple spatial watermarking technique in color images," in *International Conference on Information Technology: New Generations (ITNG '08)*, pp. 777–782, April 2008.

Review Article

A Comprehensive Review on Adaptability of Network Forensics Frameworks for Mobile Cloud Computing

Suleman Khan,^{1,2} Muhammad Shiraz,^{1,2} Ainuddin Wahid Abdul Wahab,^{1,2} Abdullah Gani,^{1,2} Qi Han,^{1,2} and Zulkanain Bin Abdul Rahman²

¹ Centre for Mobile Cloud Computing Research (CAMCCR), Faculty of Computer Science and Information Technology, University of Malaya, 50603 Lembah Pantai, Kuala Lumpur, Malaysia

² University of Malaya, 50603 Lembah Pantai, Kuala Lumpur, Malaysia

Correspondence should be addressed to Suleman Khan; suleman@siswa.um.edu.my

Received 14 March 2014; Accepted 27 May 2014; Published 6 July 2014

Academic Editor: Fei Yu

Copyright © 2014 Suleman Khan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network forensics enables investigation and identification of network attacks through the retrieved digital content. The proliferation of smartphones and the cost-effective universal data access through cloud has made Mobile Cloud Computing (MCC) a congenital target for network attacks. However, confining in carrying out forensics in MCC is interrelated with the autonomous cloud hosting companies and their policies for restricted access to the digital content in the back-end cloud platforms. It implies that existing Network Forensic Frameworks (NFFs) have limited impact in the MCC paradigm. To this end, we qualitatively analyze the adaptability of existing NFFs when applied to the MCC. Explicitly, the fundamental mechanisms of NFFs are highlighted and then analyzed using the most relevant parameters. A classification is proposed to help understand the anatomy of existing NFFs. Subsequently, a comparison is given that explores the functional similarities and deviations among NFFs. The paper concludes by discussing research challenges for progressive network forensics in MCC.

1. Introduction

The latest development in IT has introduced the concept of mobile cloud computing (MCC), in which data are stored and applications are processed in computational clouds. MCC is employed to mitigate problems related to battery life, computational power, memory capacity, and processing delays in smartphone devices [1]. Specifically, computationally intensive applications are offloaded to the computational cloud, which executes and returns the results back to the smartphone device [2]. These applications are executed on remote resources, such as physical and virtual machines, provided by cloud service providers (CSPs). Users are unaware of the location where the offloaded applications are executed. This condition implies that the execution of applications is transparent owing to the concept of virtualization in MCC [3–5]. However, no process in MCC is possible without network links that connect resources within and outside the cloud. Such network communication links in MCC are

called “network positioning,” which is divided into three types, namely, cloud access, data center, and intercloud networks [6]. All network positions in MCC are subject to network attacks that affect various hosts, servers, and data centers. Attackers access network links and perform malicious actions on network packets to propagate adverse effects to cloud resources. Eavesdropping, data modification, IP address spoofing, DoS, DDoS, man-in-the-middle, and packet content modification are examples of such attacks [7]. Researchers have proposed several network forensic frameworks (NFFs) to explore digital evidence and identify the origin of attacks [8, 9], detect malicious code [10], and monitor attackers’ activities in traditional networks [11]. However, MCC networks lack NFFs, which are necessary given the number of attacks that occur in MCC networks.

Attackers access cloud resources through cloud access networks and perform malicious actions inside the cloud [12]. A comprehensive approach is required to investigate such malicious behavior by extracting legal evidence from various

network devices and network positions in MCC, which is only possible when a network forensics investigation (NFI) has access to the networks of MCC. NFI has access only to the cloud access network and not the data center and intercloud networks in MCC [13]. Such limitation restricts the capability of NFI to investigate attacks and identify evidence found in the networks inside the cloud. To address this issue, CSPs must perform their own network forensics and identify legal evidence. Such approach would provide forensics as a service (FaaS) [14] to MCC users [15]. A number of current NFFs can help CSPs adapt to MCC networks to identify vulnerabilities and the origin of the attack [16–18]. However, comprehensive studies on the adaptability of current NFFs to MCC networks are rare. To our knowledge, no study has focused on the implementation and adaptability of current NFFs to MCC network infrastructures.

This study is motivated by the difficulty of addressing numerous attacks [15] and the lack of NFFs for MCC networks. It focuses on three different aspects of forensics for MCCs, namely, (a) adaptability of current NFFs to MCC networks, (b) provision of FaaS to MCC users, and (c) use of current NFFs to convey information on malicious attacks in MCC networks with only a few false-negative results. Two reasons explain this selection. First, this selection saves time and cost that CSPs might spend on reinventing the wheel regarding NFFs. Second, this assortment narrows down the scope of network forensics in MCC for a comprehensive study of the subject. The objective of this comprehensive review is to provide researchers with insights into the latest ideas in current literature, adaptability of current NFFs to MCC networks, and unresolved issues and challenges faced by CSPs. Such review is critical given that calibration and depiction of legal influences have yet to be considered by CSPs and legislators.

The following are the contributions of this study.

- (a) Classification of current NFFs based on their implementation.
- (b) Identification of evaluation parameters for current NFFs based on the MCC context.
- (c) Analysis of current NFFs based on evaluation parameters that highlight similarities and differences among NFFs.
- (d) Identification of issues and challenges in deploying NFFs to investigate cybercrimes in MCC.

The paper is organized as follows. Section 2 presents the theoretical framework by explaining MCC, digital forensics, and the significance of network forensics and positions in MCC. Section 3 presents the overview and classification of current NFFs based on their implementation and explains the anatomy of each current NFF. Section 4 presents the qualitative analysis of current NFFs based on selected evaluation parameters in the context of adaptability to MCC. Existing NFFs are also compared to highlight the similarities and differences of their adaptability to MCC networks. Section 5 discusses related issues and challenges. Section 6 provides the conclusion and future research directions.

2. Background

This section provides background knowledge for a comprehensive understanding of the rest of the paper. Digital forensics is introduced, and its process models and role in MCC are examined. MCC is also discussed in terms of its three entity models, namely, smartphone devices, networking, and cloud infrastructure, to provide readers with knowledge on each part of the MCC infrastructure. The importance of network forensics in the MCC paradigm is also discussed by identifying the significance of NFFs for MCC networks in detecting various vulnerabilities in such networks.

2.1. Mobile Cloud Computing. MCC is a revolutionary model that allows mobile users to connect to computational clouds through their smart mobile devices from anywhere at any time [6]. MCC is a combination of smartphone devices, wireless channels and network links, and clouds, as shown in Figure 1. Smartphones that connect to MCC suffer from constraints, such as limited battery life, memory, and processing unit as well as delays in executing an application. Such constraints prevent users from executing computation-intensive applications on a lightweight smartphone device [1]. To overcome these problems, different offloading mechanisms [19–22] have been proposed to partially or entirely outsource the computational load of smart mobile devices.

Cloud computing also involves the combination of various resources to form data centers; users can then utilize data center resources to compute and store applications [23]. These resources are integrated by the CSP in data centers that merge to form a cloud [24, 25]. Clouds help smartphone users connect to the Internet from any location. Users can connect to cloud resources at any time; CSPs integrate the resources of different organizations and provide a virtual environment to facilitate smartphone use in MCC [6]. Users benefit from the virtual resources assigned by CSP as they experience reduced time delays, increased availability and reliability, and proper load balancing in computation-intensive applications. In addition, users are only charged by CSP for what they have used in the cloud [26]. A cheaper means to utilize the powerful resources of MCC without paying for their infrastructure is thus provided. Cloud resources have high storage capacities, which are utilized by smartphone users to transfer large amounts of important data to MCC. Storing data in the cloud prevents data loss, virus attacks, leakage, and data alteration when an attacker gains access to a smartphone device [6]. Many service providers, such as Amazon S3 [27], Drop Box [28], Google Drive [29], Google Docs [30], and <http://www.SalesForce.com/> [31], allow users to store their data in a cloud. These cloud services can be accessed by connecting to the Internet through different networks.

Smartphones connect to 3G/4G or wireless networks, which serve as a gateway to cloud services. These networks are accessible and allow smartphone users to utilize network services that connect them to MCC from any location. Smartphones also have integrated Wi-Fi chipsets that connect such devices to wireless networks, which are also called 802.11 networks [1]. A smartphone can join a wireless network when it is within range of an access point to detect the signal.

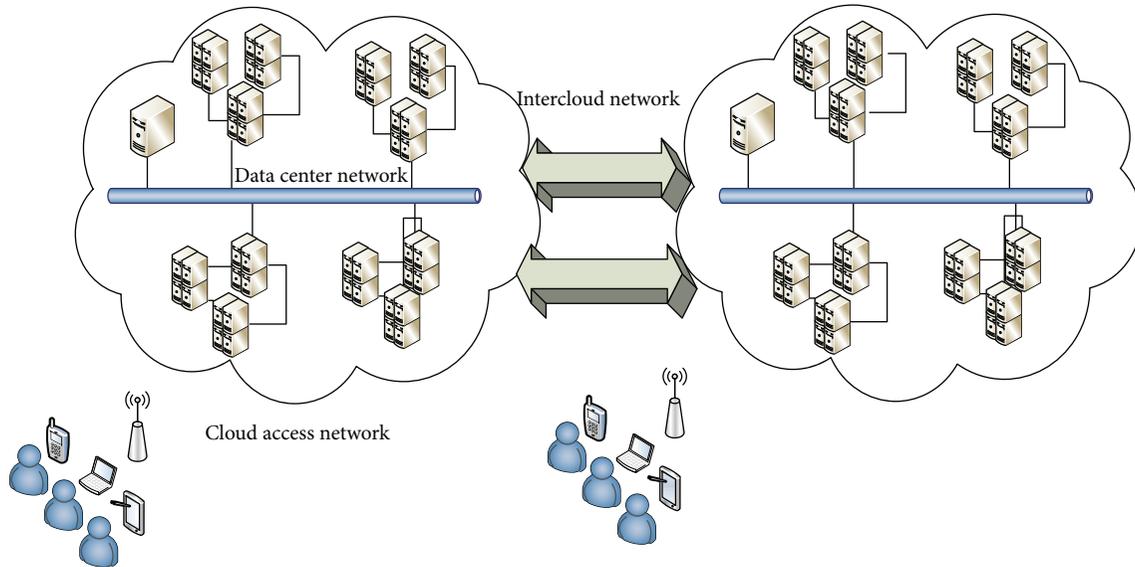


FIGURE 1: Mobile cloud computing basic network architecture.

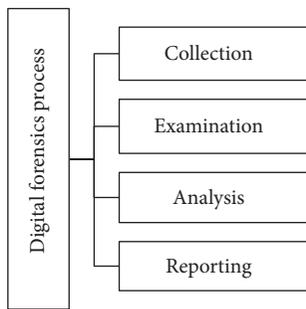


FIGURE 2: Digital forensics process model.

A wireless network is the simplest and most inexpensive means to connect smartphones to MCC via the Internet. However, using wireless networks to connect to the Internet and MCC presents certain constraints related to power assignment [32], load balancing [33], channel assignment [34], and secure communication [35]. Moreover, wireless communication involves weak encryption techniques [35] and thus allows an attacker to exploit networks mostly through DoS and DDoS attacks. Many smartphone users are unaware of such attacks and the vulnerability of wireless networks and are thus likely to be victimized by various attackers.

The MCC model is composed of different types of networks, including cloud access, data center, and intercloud networks. However, data center and intercloud networks are inaccessible to NFI in its investigation of various cybercrimes that occur in an MCC network. Security breaches in MCC networks require comprehensive NFFs to overcome various network susceptibilities. The probability of an attack increases as networks continually evolve, and the network forensics process must be mandatory to identify and prevent cyberattacks. The following section explains the fundamentals of digital forensic mechanisms.

2.2. Digital Forensics. Digital devices are vulnerable to security breaches [36]. Hackers employ malware and spyware to exploit the security flaws of smartphone devices [37]. Malware and spyware are malicious codes that allow an attacker to spy on user activities through smartphone data, such as email, calls, Internet browsing, SMS, and GPS location. Attackers eavesdrop on users' conversation through malware and spyware software [37]. Digital forensics is employed to investigate cyberattacks and malicious codes that affect digital devices, particularly smartphones in the MCC paradigm. Evidence from digital artifacts is identified to investigate the malicious behaviors of attackers [38], including monitoring, altering, deleting, inserting, and copying user data on digital devices. The malicious behaviors of attackers compromise the confidential credentials of users by damaging their privacy [39]. Several models of the digital forensics process have been proposed to conduct digital investigation in different research aspects, such as law enforcement, military operations, and business and industry [38, 40–45].

A digital forensics process model requires four steps to conduct the investigation; these steps are acquisition, identification, evaluation, and admission. These steps are integrated to obtain digital evidence from digital artifacts, which are then presented in court [46]. Alternatively, these steps were modified into five steps, namely, preservation, collection, examination, analysis, and presentation, in [40]. The first harmonized digital forensic investigation model was developed by integrating existing iterative and multitier models [47] to perform efficient digital investigation under legal terms and conditions. The National Institute of Standard and Technology explains the digital forensic process in their report by proposing four steps, namely, collection, examination, analysis, and reporting, as shown in Figure 2 [48].

The collection phase is the initial stage, wherein digital evidence is collected from digital artifacts. This stage is crucial

because collecting incorrect data in this step results in errors in the subsequent stages. The examination phase focuses on the significance of the data to the digital investigation process. The data collected are stored in a memory storage device or a portable drive that allows data to be transferred without altering the format and losing the integrity of the data [49]. The next phase analyzes the data to identify various vulnerabilities and the malicious behavior of the attacker as well as determine the origin of the attack. The analysis is performed with several digital forensic tools, such as EnCase [50], FTK [51], Sleuth Kit [52], and Helix [53]. These tools identify digital evidence extracted from temporary, deleted, and register files as well as from cache, cookies, email, and metadata present in digital devices [54, 55]. Such digital evidence acts as proof against attackers in court by presenting a complete and accurate investigation report. The investigation report is the final phase of digital forensics. All the activities performed by forensic investigators in the preceding stages present a complete picture of the investigation process in terms of a legal document.

Digital forensics is fast becoming a hot topic in MCC. The demand for digital forensics in MCC continues to increase because of the increasing number of attacks, such as DoS, DDoS, and botnets [5, 15, 56, 57]. The origin of attackers must be investigated to stop these attackers from performing further malicious acts. Such investigation in MCC requires a comprehensive digital forensics strategy to perform data collection and examination through virtual situations, live forensics, evidence segregation, and proactive preparation mechanism [58]. The rapid progression and increasing attractiveness of MCC necessitate the establishment of digital forensics in the cloud environment. Novel investigation techniques are required to prevent cybercrimes in MCC. However, several challenges in MCC, such as access to the entire cloud, jurisdictional issues, and technological advancement, create problems in forensics investigation [13]. Nevertheless, such situation also creates opportunities to establish standards, policies, and procedures in cloud-based forensics.

2.3. Network Forensics in Mobile Cloud Computing. Cloud services in MCC are acquired by smartphone users by connecting through Wi-Fi, WLAN, 3G/4G, and long-term evolution networks [1]. These networks have to be fast and secure enough to send user requests to computational clouds and send the results back to smartphone users. However, these networks are targeted by attackers to gain access to the network in the form of a network breach [59]. The current security solution involves the use of intrusion detection systems (IDS) and firewalls to detect and identify attack patterns [12]. However, intelligent attacks circumvent such security solutions to propagate malicious activities in the network [60]. Thus, security solutions should be sufficiently intelligent to detect intelligent attacks that compromise the system [61].

The network forensics process identifies attacks while monitoring and analyzing network traffic. Network forensics delivers two types of services in the investigation process.

First, it detects intrusion and malicious traffic [61]. Second, it collects and analyzes network traffic by performing traceback [16], using attack graphs, and parsing voice over IP (VoIP) [62] in a converged network. Collecting network traffic from high-bandwidth network channels restricts network forensics from capturing large amounts of network traffic, particularly in MCC. High-bandwidth networks pass millions of network packets per second and require a large storage capacity to store network packets for further analysis. One option is to use the storage services of cloud data centers to store and access large amounts of data [63]. Large amounts of data are mostly generated in smartphones through the use of multimedia data applications, such as video chat (Skype), video uploading or downloading (YouTube), audio traffic (iPlayer), and online games. Most of the multimedia traffic are based on real time and must flow quickly and securely to perform accurate communication. To capture multimedia traffic, live forensics technique [64] is required to identify vulnerabilities in the network flow on a real-time basis. Live forensics technique is utilized to capture volatile network traffic that can be eliminated with the power off or on arrival of new network traffic. Volatile data must be captured in network communication for two main reasons. First, network packets pass through different ports and reach their destination without being stored at the destination. Second, attackers reach the compromised system, collect confidential data, and delete trace outs. Network forensics must be conducted at live data communication to address these problems [65]. However, this approach incorporates the overheads of highly computational processors, I/O devices, and large storage devices for collecting, analyzing, and storing real-time network traffic. Such overheads are minimized by collecting data only at peak hours when capturing attacks is also likely to be achieved [66]. The processing time and storage load can then be reduced, and precise results can be obtained in the forensics investigation. The rest of the network links that are inaccessible for real-time network investigation has to be searched to identify attack patterns that would assist in tracing the origin of the attack.

The significance of network forensics is in each part of the MCC's network communication channels. Malicious behavior in network packets needs to be traced through NFF whether a smartphone user is connected to mobile clouds or data centers are interconnected or linked to other cloud data centers. NFIs have limited access to investigate different network susceptibilities [13]; thus, network forensics should become a permanent service to MCC users through cloud network channels and resources.

2.4. Network Positions in Mobile Cloud Computing. Network position shows the location of the network that connects two entities in MCC. Three types of networks generally exist in MCC: cloud access, data center, and intercloud networks [6], as shown in Table 1. Cloud access network is the network that connects smartphone devices and the cloud through the Internet [67]. This network is utilized when a smartphone user connects to the cloud to offload and download an application to the cloud [76]. However, connection difficulties

TABLE 1: Generalized network positions in MCC.

Network positioning	Entities link	Example	Objective	Accessibility
Cloud access network [67, 68]	User ↔ cloud services	Internet, NGN, 4G	Dynamic routing, accessibility to cloud	Possible
Data center network [69]	Data center ↔ data center	Cluster computing	Load balancing, virtualization, intensive computing	CSP
Inter cloud network [70]	Cloud system ↔ cloud system	Cloud resource migration	Cloud collaboration	CSP

TABLE 2: Description of network positioning in MCC.

Cloud access network	Data center network (DCN)	Intercloud network
Connects smartphone user to cloud system through wireless, radio access network (RAN), 3G/4G and LTE networks [4].	DCN connects application software & cluster of computers within data center [70].	Connects two or more cloud systems for cloud collaboration [68]
Public cloud networking makes availability of network applications to users via internet [71].	Its network expands towards connecting two or more data centers with a single cloud system [70]	It not only connects two cloud systems but provides additional functionality such as data format conversions, network virtualization, service availability, address management, intelligent routing, and efficient security [72].
It faces challenges related to security, compliance, privacy, and high availability [71].	It maintains low cost with maximizing efficiency and throughput [71]	It benefits by connecting with one cloud system and acquires its services, dedicated network, and increase transfer speed through protocol optimization [73].
RAN lacks centralize organization for emerging heterogeneous networks, flexibility to drift network services towards network verge for new application utilization and generate revenue from it [74].	Mostly faces two challenges such as scalability and cost effectiveness. Scalability depends on architectural design of DCN while cost depends on its power consumption [75].	

related to security, compliance, privacy, and high availability exist. The cloud itself is composed of data centers, and a data center is a combination of resources. The network that connects all resources within a single data center or the network that combines one or more data centers is called “data center network,” which in turn involves intra- and interdata center networks [69]. Data center networks are employed in the cloud when data are transferred from one resource to another or to other data center resources within the cloud for execution, such as for cluster computing. Scalability and cost effectiveness are the two issues related to this network [69]. Scalability depends on the architectural design, and cost effectiveness depends on the power consumption of a data center network [77].

The connecting network between two or more clouds is called intercloud network [70]. Such network deploys a fiber optics network that has high bandwidth and high-speed line rate. Intercloud network is used when one cloud migrates or sends an application to another cloud for execution or storage. Intercloud network provides a dedicated network and increases transfer speed through protocol optimization. A brief description of each network position in MCC is presented in Table 2. All network positions in MCC are considered vulnerable to attacks in NFFs. No network is safe from an attack because of vulnerabilities, which require further investigation to determine the origin of the attack [8].

Network forensics has a vital function in investigating networks to identify legal evidence on cyberattacks [16]. Network attacks performed on traditional networks can be easily investigated by obtaining legal evidence through current NFFs [71]. Such networks are accessible and allow for immediate investigation. However, data center and intercloud networks in MCC are inaccessible to third-party NFIs because these networks are the sole property of CSPs. The lack of third-party access limits the investigation of NFIs on cloud networks [13]. The best option is for CSPs to conduct their own network investigation and provide FaaS to users, which also generates additional revenue. Almost all networks within the cloud territory are accessible to CSPs; network attacks can be easily investigated, and legal evidence can be obtained. Moreover, MCC users would find it easy to trust the data and believe that such data are safe from third-party NFIs [72].

3. Network Forensic Frameworks

The classification of NFFs based on an exhaustive literature review is presented in the first part of this section. Such classification is derived from the implementation of the architectural frameworks of network forensics, which narrows down the scope and allows for a comprehensive study of the area. NFFs are classified into five categories, namely,

traceback NFFs, converged networks, intrusion detection systems, attack graphs, and distributive frameworks.

In the second part of this section, the structural aspect of current NFFs is discussed in detail by reviewing its frameworks, approach, methods, evaluation, limitations, and output performance. A complete operational overview of each NFF and its implementation objectives is presented.

3.1. Overview and Classification of Network Forensics Frameworks. Network forensics aims to identify legal evidence from network traffic to investigate the origin of the attack and attacker behavior [8]. NFFs capture and analyze network traffic in the network to investigate attacks performed by different attackers [16]. NFFs extract information from network traffic to rebuild emails, messages, FTP traffic, and various other communications. The process helps NFIs reconstruct the attack path and determine the attack's origin [62]. Traditional networks can readily access breached network devices and acquire data for investigation. However, accessing and acquiring legal evidence from breached devices in MCC are difficult because of the CSPs' resource property and virtual resource infrastructure [3]. NFIs are unable to initiate the investigation process without access to network resources in the cloud. Network data integrity in MCC is another concern [49]. Anyone can access cloud resources and alter user data, resulting in the loss of data integrity. Frequent data migration within various clouds can also affect data integrity because of the storage of data in different formats and in different databases. Privacy is also an issue in the investigation of MCC networks because user information could be disclosed when a malicious activity is investigated [72]. User data traveling in the network at the same time further complicates efforts to capture malicious user data among various users, especially in high-speed data rate networks in MCC. Each network link might contain millions of user data simultaneously. Thus, identifying a specific user's data without disturbing that of the others is difficult. Real-time analysis is also a challenging task in MCC because the NFI cannot access cloud networks [74]. Such analysis is important because network data are volatile. Network data might lose their identity after being overwritten by other data, and closing the session could allow attackers to alter the data and delete attack traces from the network. In business, health, and other industries, real-time analysis is necessary to handle malicious situations occurring in the environment. Current network forensic tools lack the capability to capture, record, and analyze high-speed line rate data at various channels of MCC networks [58, 75]. An intelligent network forensics tool that is compatible with the virtual and disseminated environment of MCC networks is thus essential. Moreover, high bandwidth is necessary to acquire legal evidence from different locations in MCC [73]. Acquiring data from a remote cloud data center network requires a dedicated bandwidth to execute the process quickly and respond to user queries in real time. Traditional networks are simpler than MCC networks because of the former's access to resources; such access allows for the use of less bandwidth with quick incident responses. A chain of custody for network evidence is also important for

TABLE 3: Issues in current network forensics and MCC network forensics.

Issues	Current network forensics	MCC network forensics
Data acquisition [79]	No	Yes
Access to artifacts [79, 80]	No	Yes
Bandwidth utilization [73]	No	Yes
Chain of custody [14, 74, 75]	No	Yes
Data Integrity [74]	No	Yes
Privacy [81, 82]	No	Yes
Real time Analysis [74]	No	Yes
Volatile data [83, 84]	No	Yes
Forensics tools [14, 75, 79]	No	Yes

the MCC environment. Each event in the MCC network has to be validated, particularly as to how and where the data are stored on cloud resources while maintaining their integrity [75, 78]. Chain of custody has to be defended in court to present evidence against the attacker. Tracing each event while preserving its integrity and reliability is a challenge because of virtualized and distributed environments in MCC and the incorporation of data migration. A comparison of the status of issues regarding current network forensics and MCC network forensics is presented in Table 3. A comprehensive review is required to explain the aforementioned problems in current NFFs and illustrate deliberate compatibility with MCC networks.

NFFs can be classified into five categories, namely, traceback NFFs, converged networks, intrusion detection systems, attack graphs, and distributive frameworks. This classification is based on an exhaustive literature review. Traceback NFFs [8, 9, 16, 71, 85] identify the origin of the malicious packet in the network generated during a cyberattack. Traceback NFFs are utilized to determine the source of the packet generation [86] and eventually lead to the identification of attackers based on their source IP address. Moreover, NFFs in converged networks distinguish evidence from audio, video, and multimedia data [62, 87, 88]. Converged networks are prone to attacks because of their flaws in communication [89, 90]. In this study, NFFs for converged networks are only demonstrated for VoIP traffic. NFFs identify evidence from voice packets altered by attackers during their attack on the network. Intrusion detection systems (IDSs) are also employed to detect and log malicious packets generated by attackers [91–94]. IDSs are utilized to check packets based on provided rules and knowledge obtained by performing pattern matching at the time of capture. Packets are logged through hash functions and sent to the forensics server for further analysis. Furthermore, NFFs employ attack graphs to identify attack paths in the network. Attack graphs help reconstruct attacks by determining the activities performed by an attacker during the attack [17, 95–99]. Such a graph refines the attack scenario by studying its sequential steps from the origin of the packet to the victim node in the network. Distributive NFFs are utilized to identify malicious packets in disseminated networks by capturing them from

different locations [18, 100–102]. Distributive NFFs eliminate the bottleneck problem, which often results in high bandwidth consumption and long delays. Table 4 shows the classification of NFFs. The function of each NFF is illustrated.

3.2. Structure of Network Forensics Frameworks. This section discusses the structure of NFFs based on their approach, method, evaluation, limitations, and performance. The approach attribute presents the techniques employed by NFFs to conduct the investigation. Such techniques include logging [8], packet marking [9], spread spectrum [85], probabilistic model [17], dynamic forensics intrusion tolerance [92], forensics examination [97], distributive network forensics [100], and visualization and interaction [99] for various NFFs.

Logging is utilized to record network traffic and its patterns in the form of log files to obtain evidence on various attacks [8, 16, 18, 62, 71, 87, 88, 93, 94, 101, 102]. This technique captures network traffic at different routers and performs a hashing function on it. Log files are usually applied by a hash function that validates the files' integrity; such files can be accessed later for the investigation of several attacks. Packet marking is an approach utilized to mark packets at the router to trace packet movement in the network, which is vital for tracing back the origin of an attack [9, 71, 101]. Packet marking is commonly adopted by traceback frameworks [9, 71] to identify the attacker's source address, which they often spoof. Spread spectrum techniques are employed to traceback the origin of the attack, detect attacks, and identify receiver status by spreading signals with a frequency in a domain of a bandwidth [85]. A probabilistic model measures the uncertainty present in the attacked networks to obtain digital evidence [17, 91]. Dynamic forensics intrusion tolerance performs in a situation where a forensics server is affected by an attacker [92]. It provides real-time tolerance to the server and analyzes network traffic. Forensics examination is implemented to examine log files altered by attackers during an attack to guard against investigation [97]. Distributive network forensics collects network logs from distributed agents in disseminated networks and analyzes the logs locally or centrally through forensics servers [100]. Visualization and interaction approach makes the entire attack graph easy to analyze and allows network investigators to examine various attack paths through human-computer interaction interfaces [99].

The method attribute demonstrates how different approaches are employed to identify evidence in the network. Similar to the approach attribute, each NFF uses certain methods as shown in Table 5. Authenticated evidence marking scheme reduces the overhead in the entire network performance by parsing data at edge routers; the authenticity and integrity of network data are thus improved [71]. Lightweight IP traceback method utilizes the time-to-live (TTL) packet field of an IP header to trace packets for investigation [9]. Scalable network forensics traces back the origin of the attack based on attack traffic separated from normal traffic in traces provided over a long period of time [16]. Hopping-based spread spectrum traces back cybercrimes in anonymous network communication by

providing security and accuracy to such communication [85]. The IP traceback method employs a sinkhole router, a compressed hash table, and data mining approaches for network forensics analysis to determine the origin of the attack [8].

VoIP forensics network patterns are utilized in VoIP network traffic to collect and analyze voice packets systematically. The technique helps identify, detect, and trace attacks by generating patterns for such attacks [87]. VoIP network forensics also identifies digital evidence on various attacks by comparing normal and abnormal packets of the network. Such method reduces human intervention by collecting voice packets through sensors installed at various parts of the network [88]. A VoIP evidence model reconstructs attack events by applying secure temporal logic of action (S-TLA+), which works when it lacks sufficient information to investigate an attack [62]. This model also provides reliability and integrity for the collected information, validates the authenticity of the provided evidence, and allows NFIs to capture unknown attacks undetected by other network forensics methods.

Probabilistic discovery and inference method is utilized by IDS to reveal unknown information related to the analysis of network traffic [91]. The method is only applicable to network-based IDS and not to application-based IDS. Such probabilistic method aids in forensics explanation, which is based on unreported signature rules and observed network-based IDS alerts. Moreover, formal methods are employed to show that a forensics server is sufficiently tolerant and thus works even if it is subjected by several attacks [92]. The availability of the forensics server is enhanced by collecting significant evidence. Steganography is also utilized to conceal original log files from an attacker. Logs are converted into images and are hidden from the attacker. The conversion is performed by changing the least significant bit for each pixel without being visible to the naked eye [93]. The original log files are kept in the custody of NFIs, which then trace attacks by monitoring copied log files for alteration. Network forensics architecture is applied to perform multidimensional forensic analysis based on log messages and network data [94]. Network traffic is monitored at the time of network entry, and log data are recorded and analyzed as data move out from various security devices. This twofold monitoring and analysis technique ensures the reliability and credibility of evidence extraction through network traffic.

Scalable analysis [95] is performed to measure the effect of current and future attacks in large-scale networks. Large amounts of network traffic are analyzed in real time by measuring the effect of a single attack on the enterprise and its correlation to other attacks [95]. Moreover, a multilevel and multilayer attack tree identifies the system level risk by analyzing various security threads resulting from various network attacks [96]. It helps NFIs become familiar with future network threats. Antiforensics method can be incorporated with attack graphs to trace attackers based on their activities. Antiforensics method urges attackers to perform malicious actions while being monitored [97]. In addition, a fuzzy cognitive map can be incorporated with a genetic algorithm to identify the worst attack path among a large number of attacks present in the network [98] and help NFIs track and

TABLE 4: Classification of network forensics frameworks.

Frameworks	Functions
Traceback	NFEA [71] Proposes effective tracking range to provide admissible digital evidence with guarantee of integrity and authenticity of track data. Further, it marks packets at edge router which increase efficiency and decrease loss of data.
	LWIP [9] Considers only time to live (TTL) field of IP header to trace out attack path in DDoS attacks. It used three algorithms that address three steps to make proposed scheme efficient, robust, and simple such as (a) embeds TTL value in IP header, (b) performed soon as DDoS attack occur, and (c) attack tree analysis algorithms is executed.
	Scalable NF [16] Proposes scalable network forensics scheme for stealthy self-propagating attacks to traceback the origin of attack. Moreover, scheme is scalable in terms of computational time and space to accurately discover origin of attack. In addition, data reduction mechanism is used to identify deviations of each host and it acts as indication for a potential attack which is further process for forensics investigation.
	HB-SST [85] Presents generic hopping based spread spectrum technique for network forensics traceback in anonymous communication networks. It provides randomized effect to mark network traffic in both time and frequency domains.
	ITP [8] A protocol is design to traceback attacks in real time as well as periodically using compressed hash table in the router. Further, it addresses replay attacks through timestamp attached to the messages and its integrity is verified through using hash function. Moreover, it enhances detection rate of attacks by updating attack list periodically in routers.
Converged network	PBNF [87] Proposes VoIP network forensics patterns that use to collect and analyze voice traffic in a systematic way.
	VoIP-NFDE [88] A digital evidence procedure for VoIP network forensics is proposed especially for internet phone. Evidence is identified by comparing normal and abnormal packets in voice communication.
	VoIPEM [62] Model based forensics method is proposed to identify malicious attacks in VoIP communication that formalize hypothesis through information gathering. Moreover, attack path is reconstructed by adapting secure temporal logic of action (S-TLA+) which provide clear evidence about attacks.
Intrusion detection system	AIDF [91] An analytical intrusion detection framework proposed, based on probability model discovery approach & inference mechanism. It provides forensics explanation not only on intrusion alerts, but also on unidentified signature rules. Moreover, it integrates intrusion alerts from disseminated IDS sensors.
	DFITM [92] Intrusion tolerance base dynamic forensics modeling is performed to enhance availability of forensics server in case of an attack. Modeling is conducted with finite state machine and forensics server availability is analyzed through numerical analysis.
	IIFDH [93] Steganography is applied to identify alteration in log files performed by an intruder after his malicious attack. It maintains reliability and completeness of the evidence for future decisions.
	NFIDA [94] Network forensics based on intrusion detection static and dynamic analysis is performed to provide complete record of data and logs while ensuring credibility and reliability.
Attack graphs	SA [95] Proposes a framework that performs scalable analysis of attack scenarios by analyzing massive amount of alerts in real-time situation. Moreover, it also addresses individual attacks and its impacts on the enterprise.
	MLL-AT [96] It identifies multistage network attacks and analyzes system risk by evaluating various security threads that occurs due to attack sequences.
	AGFE [97] Integrates antforensics mechanism with attack graph to fully observe intruders while deleting certain traces after attack performed.
	FCM [98] Generate fuzzy cognitive map from attack graph with the help of genetic algorithm to find a worst attacks in the network. It simulates a situation for network investigator to tackle such attacks with great concern.
	CSBH [17] A probabilistic approach is proposed that integrates attack graph with hidden Markov model for exploring system states and its observation. It identifies the root cause of attack with providing automation, adaptability, and scalability in large network for cost benefit security hardens.
	AGVI [99] RAVEN framework is proposed that reduces sophistication in large attack graphs by providing interactive visualize interfaces for user to illustrate attack graphs easily.

TABLE 4: Continued.

Frameworks	Functions
ForNet [100]	Proposes distributive framework to collect network logs from different network devices in disseminated network. It analyze IP packet header for IP connection, ports, and various sessions through bloom filter tracking.
DRNIFS [101]	It captures network packets soon as an attack is detected in a real-time situation. Moreover, it collects potential evidences that are deleted in most of the cases by intruders after its malicious attacks. It uses centralize network forensics server with disseminative detective agents.
Distributive	Proposes framework that identifies potential risk, misbehavior of packets, and origin of attack with having distributed cooperative network forensics system. The system is comprised of client server architecture, with client agents installed on different system to capture network traffic logs from different network artifacts.
DCNFM [102]	It proposes artificial intelligence immunity theory to address network forensics in real time with keeping evidence in a safe way. It provides validity, integrality, and authenticity for evidence in a real time situation.
DNF-IA [18]	

reveal attacks that have the worst effect on the network. Raven architecture is utilized to visualize and interact with attack paths in the attack graph [99]. The situation of attack graphs is simplified; thus, NFIs can easily understand several attack paths in the network.

Likewise, distributive architecture is utilized to investigate network packets according to the IP connections, port records, and various session creations between hosts; bloom filter tracking is adopted in this architecture to collect evidence against attackers [100]. This investigation procedure provides comprehensive information against attackers by extracting evidence from various events in the network. Other distributive network forensic architectures are utilized to produce rapid responses in the generation of the attacks while investigating network traffic and logs dynamically [101]. The problem of static analysis is addressed, and time delays in incident response are reduced. Client server architecture is utilized to identify potential risks, misbehavior of packets, and origin of the attack [102]. It captures data from distributed locations, converts traffic into a database, analyzes network attacks, and produces visual reports by performing a statistical analysis of the entire process. The dynamic network forensic model simplifies the situation by providing authentic, integral, and valid digital evidence collected from distributed locations in the network [18]. Agents installed at distributed locations capture raw network traffic and apply antigenic presentation coding on it. This condition helps NFIs collect real-time evidence on network attacks.

The evaluation attribute in Table 5 represents the techniques employed by NFFs to analyze frameworks. Different NFFs employ various methods to evaluate the framework. These methods include test bed [71, 88, 91, 97], simulation [8, 71, 85, 98], tree analysis algorithm [9], S-TLC+ [62], finite state machine [92], prototyped [93], synthetic & real attack graph [95], case studies [96], and many other scenarios [17]. The limitation attribute reveals the shortcomings of NFFs that affect the entire network forensics process as the investigation is performed. These shortcomings include computational overhead [71, 94, 95], storage overhead [71, 92, 96, 101], router overhead [8, 9], capturing real time network traffic

[16], scalability [85, 87, 93, 96, 97], time consuming [88], bandwidth utilization [88], forensics server bottle neck [18, 87, 102], observation depended [98], lack of awareness [98], and specific attack investigation [100]. Table 5 also presents the output derived from evaluating NFFs based on their evaluation values in the performance attribute.

4. Review of Network Forensics Frameworks in the Context of Adaptability to MCC

This section highlights the parameters used for evaluating adaptability of existing NFFs in MCC networks. The parameters include scalability, overhead, accuracy, complexity, and privacy. Most of the NFFs discussed in the review deal with capturing the network packets from networks, followed by the analysis phase on forensics servers. In case of large distributed networks, forensic servers should scale to analyze huge amount of network traffic. In particular, existing NFFs must support network forensics by collecting network evidences from various distributed networks connecting millions of MCC resources. Therefore, NFFs must scale in order to collect, preserve, analyze, and report network evidence in MCC in a real-time. The overhead of existing NFFs is required to be minimal in terms of (a) computation and (b) storage. Computational resources are required to analyze huge amount of networked data to extract sources of evidence. On the other hand, this data is also stored in a system that has to be analyzed later by various forensics mechanisms. The accuracy attribute for NFFs is vital to measure filtration of irrelevant network traffic for analysis of evidence. Extraction of significant data from the network is important in MCC due to restricted access to the cloud data. High accuracy results in less time for analysis of data and producing quick incident response. The complexity of existing NFFs can be analyzed for MCC networks in terms of its implementation, collection, investigation, and analysis. It implies that it is difficult to apply existing NFFs in MCC networks due to inaccessibility of cloud network, virtualization, and distributed networks. The privacy is considered as one of the most important issues

TABLE 5: Structure of network forensics frameworks.

Frameworks	Approach	Methods	Evaluation	Limitations	Performance	
Traceback	NFEA [71]	LO, PM	Authenticated evidence marking scheme (AEMS)	Test bed & Simulation	Computational & Storage overhead	50% performance degrades when AEMS applied to each packet. However, performance gains 40% when it is applied to only select packets.
	LWIP [9]	PM	Lightweight IP traceback based on TTL	Tree analysis algorithm	Router overhead	Significant path reconstruction in DDoS attack
	Scalable-NF [16]	LO	Scalable network forensics	Real world traffic traces	Capture real time traffic	Reduce 97% of irrelevant data for analysis
	HB-SST [85]	Spread spectrum techniques	Hopping based spread spectrum	Simulation	Scalability	False positive decrease exponential with increase in signal length.
	ITP [8]	LO	IP traceback protocol (ITP)	Simulation	Router overhead	ITP shows better results in term of false positive rate & attack detection as comparing with existing frameworks
Converge network	PBNF [87]	LO	VoIP network forensics patterns	Suggest to use NFATs	Scalability, Forensics server bottle neck	Faster and structural investigation in VoIP traffic
	VoIP-NFDE [88]	LO	VoIP network forensics with digital evidence	Test bed	Time consuming, bandwidth utilization	Collects, analyzes, and performs forensics in VoIP DEFSOP operational stage
	VoIPEM [62]	LO	VoIP Evidence Model	S-TLC+	Not trace anonymous attacks	Identifies significant information relate to attacks
Intrusion detection system	AIDF [91]	Probabilistic model	Probabilistic discovery & inference	Test bed	Database for untreated data	Perfect discovery results in 16.67% and information combining from multiple IDS for forensics explanation is 87%
	DFITM [92]	Dynamic forensics intrusion tolerance	Formal methods	Finite state machine	Storage overhead	Enhancement of availability of forensics server with improvement of collected significant evidence
	IIFDH [93]	LO	Steganography	Prototyped	Scalability	Real-time detection with preservation of evidence
	NFIDA [94]	LO	Multi-dimensional analysis	Not applicable	Computational overhead	Records complete network data with providing data integrity that results in network forensics solution based on intrusion detection analysis.

TABLE 5: Continued.

Frameworks	Approach	Methods	Evaluation	Limitations	Performance	
Attack graph (AG)	SA [95]	Measure current & future attacks	Scalable analysis	Synthetic & real AG	Computational overhead	For large graph the integer value k increases when processing time increase. However it remains stable for small graphs
	MLL-AT [96]	Network attack modeling	Multi-level & layer attack tree	Case study	Scalability, Storage overhead	Model attack more accurately, address system risk
	AGFE [97]	Forensics examination	Anti-forensics injection in AG	Test bed	Scalability	Identifies alteration performed by intruders in log files.
	FCM [98]	Network security evaluation	finite cognitive map & genetic algorithm	Simulation	Observation depended, lack of awareness	Results best fit value of 1.64 that shows the probability of goal achieved.
	CSBH [17]	Probabilistic	Design model	Scenario based	Computational overhead	It finds that an approach is user centric, with complexity $O(MN^2)$.
AGVI [99]	Visualization & Interaction	RAVEN	Not applicable	Visualization in real time situation	Address impact of HCI techniques on attack graphs	
Distributive	ForNet [100]	distributive network forensics	Architecture	Not applicable	Limited attack detection due to lightweight filtering	Provide valuable, trustworthy information about network events
	DRNIFS [101]	LO, PM	Architecture	Not applicable	Storage overhead	Real time detection with quick incident response
	DCNFM [102]	LO	Client Server Architecture	Not applicable	Forensics server bottle neck, Storage overhead	Identifies origin of attack and potential risk
	DNF-IA [18]	LO	Dynamic network forensics model	Laboratory test	Lack of cryptography, forensics server bottle neck	Integrated, accurate results in real-time situation when attacks are occurred.

Approaches: LO: logging; PM: packet marking.

for MCC nowadays. User migration towards the cloud must ensure the data integrity and safety from unauthorized access. Privacy in NFF is significantly different as compared to the MCC. Therefore, applicability of NFFs in MCC must be evaluated in terms of privacy.

4.1. Evaluation Parameters for the Analysis of NFFs. Scalability is an important MCC parameter [103]. Scalability is enabled by the concept of virtualization in different entities, such as servers, data centers, resources, operating systems, and networks. Scalability in terms of networks maximizes throughput, improves performance, and ensures availability and reliability [104]. Increased network scalability enhances data mobility, which ultimately demands more monitoring and analysis features in the context of network forensics.

Therefore, NFFs are a vital part of MCC to investigate network links and devices that scale according to the demands. Moreover, current NFFs [16–18, 99] can integrate features with attack patterns and security devices to investigate network traffic for MCC. Scalability can be regarded from two perspectives: horizontal and vertical scalability for MCC [105].

Horizontal scalability is also called “scale out”; it deals with system throughput by enhancing its complexity. Horizontal scalability increases the number of counts in terms of hardware resources and performs tasks more quickly [106]. The scope of horizontal scalability for MCC is mainly a result of disseminated networks that combine different data centers within and out of the cloud. The capacity of network forensic servers has to be sufficiently large to capture, record, and analyze network traffic from several network links and

devices in the cloud environment. For instance, one forensic server may be used for one trillion packets instead of using two or more forensic servers for the same number of packets for investigation [92]. Sending network packets through the shortest path to their destination reduces the number of hop counts and thus helps NFIs easily capture and analyze network packets with less time delay and increased system throughput.

Vertical scalability is known as “scale up.” It handles the improvement of the existing functions and features of a system by adding more hardware [105]. For instance, adding extra processors or memory in a network forensic server executes and stores more network packets with less time delay. Similarly, network speed and bandwidth can be enhanced to provide an efficient response to MCC user queries. NFFs attain efficiency by investigating more network packets within the same time frame. However, upgrading network resources does not always produce sound results because of diminishing returns that increase with a decrease in the performance rate. For this reason, one has to identify other factors to incorporate with scalability to produce effective network forensic outputs. Nevertheless, the vertical scalability of NFFs in MCC is important because of its rapid incident response to identifying the criminal behavior of attackers at the time of the attack. Most current NFFs lack such scalability, which needs to be adopted to meet the objectives in MCC [8, 71, 87, 88, 93, 94, 96, 97].

Overhead is related to the sophistication of NFFs and reduces the performance of the system. Network forensic overhead includes computational and storage overheads. These two overheads are calculated for MCC based on existing NFFs. The computational overhead of NFFs includes network processing, bandwidth delays, packet marking, preserving, analyzing, collecting data, investigating, logging, and integrating overheads [8, 9, 71, 97]. These factors are incorporated into the computational overhead attribute that degrades the computation of the system. The storage overhead of NFFs deals with network traffic storage at different locations and devices in the disseminated networks of MCC [8, 71].

Computational overhead (Co) is high for NFFs in MCC when a framework utilizes many resources for its computation, employs a less reliable investigation mechanism, and addresses irrelevant data, resulting in a time-consuming analysis [8, 9, 71, 87, 92, 93, 97]. The value for Co is low for NFFs in MCC when a system utilizes minimal resources to analyze network traffic packets [62, 85, 88, 91, 95, 96, 98, 100–102]. The value is moderated by utilizing resources that both increase nor decrease system performance and throughput in the computation [16–18, 94, 99].

Storage overhead (So) is ranked high when it does not have a proper mechanism to store large dispersed network packets at different locations and devices for several networks in MCC [8, 71]. Network traffic is mostly monitored and captured at different network security devices, which suffer from small buffer space to store large and high-speed data network packets. However, the value for So is low and decreases drastically when a dedicated physical resource is assigned near the cloud to store entire logs of the network traffic [16–18, 87, 92–96]. Such store logs can be further

sent to forensic servers placed in the cloud for investigation depending on the architectural framework of NFFs to manage high network traffic in the disseminated networks of MCC.

Accuracy of NFFs in MCC is calculated by separating irrelevant data from large amounts of network traffic. The accuracy of NFFs is high when network traffic is filtered to investigate the rest of the traffic; otherwise, the accuracy is low. Network traffic that enters the network incorporates two types of data traffic, namely, normal and abnormal or infected data traffic [16]. NFIs are highly concerned about abnormal data traffic because it contains evidence on attacks and the behavior of attackers. Reducing or separating normal data traffic from abnormal data traffic increases accuracy and helps investigators perform their investigation on a specific network traffic with minimal time delay [16].

However, many NFFs [18, 71, 87, 94, 97, 102] capture the entire network data traffic and thus result in delays and decline of system performance. To overcome these problems, an investigation must be performed on relevant data depending on the situational requirements. Extracting relevant data is also a challenging task for NFIs, particularly in high-speed data rate networks of MCC.

Consequently, the accuracy attribute value will be high for NFFs if the maximum level of irrelevant data is separated, such as separating normal data traffic from abnormal data traffic, without utilizing additional resources and reducing system performance [16, 98]. The easiest means is to use cloud computational resources because these are computation-intensive and have more processors to execute the process rapidly. However, the value of NFFs is marked low when NFI is performed on the entire network traffic found on various networks links; this condition ultimately results in time delays and reduced system performance [18, 71, 87, 94, 97, 102]. The value is marked moderate when network traffic is reduced, allowing for the analysis of an attacker’s behavior within a reasonable range of time delay and system performance [8, 9, 17].

Complexity attribute of NFFs illustrates the problem faced by NFIs in conducting network forensic investigations in the MCC environment. The network forensic process consists of sequential steps performed in the current network infrastructure. Such steps include collection, examination, analysis, and reporting [48]. Current NFFs also incorporate architectural implementation complexity in MCC because of its proposed frameworks. These frameworks have to be adaptive for several networks in MCC to reduce the complexity faced by NFIs.

However, adapting current NFFs in MCC is a challenging task. The process must be flexible in terms of virtualization and distributed characteristics of MCC. In addition, the mobility characteristics of data in clouds have increased the complexity of tracking, collecting, and analyzing such data. Several types of network positioning, such as cloud access, data center, and intercloud networks, also affect the complexity of current NFFs. Thus, the complexity of current NFFs is high in collecting [18, 88, 96, 100], analyzing [9, 17, 18, 62, 87, 88, 91–102], and investigating network data [16] as well as in the implementation of NFFs in MCC [8, 18, 62, 71, 85, 87, 88, 91, 92, 94, 96, 97].

Privacy is one of the important factors that divert users toward MCC. Smartphone users execute and store data in clouds to protect their data from various attacks. User data reach cloud resources by passing through cloud access, data center, and intercloud networks. Network positioning in MCC becomes a battlefield for NFIs in extracting evidence against various attackers [56]. However, confidential user information could become compromised while accessing several networks of MCC. The possibility of user privacy being compromised makes users hesitant to use MCC.

A win-win solution must be established for current NFFs so as not to compromise user privacy while performing investigation on several network positions in MCC. The value of privacy is high for NFFs when user data are not accessed during an investigation; however, this condition remains a challenge, particularly in MCC [99]. The value of the privacy attribute is moderate when some user data are accessed during the investigation [8, 9, 17, 96–98]; the value is low when user data are accessed during the investigation in MCC [16, 18, 62, 71, 87, 88, 93–95, 100–102].

Adaptability parameter can show whether current NFFs are applicable for network positions in MCC or not because of certain constraints. MCC networking has diversified into current networks as a result of its connectivity to millions of servers, capability to transfer trillions of packets per second, stateless computing, dynamic application provisioning and positioning, and virtualized features [107]. MCC networks have to redefine their features to improve their management, scalability, and administrative aspects compared with current NFFs.

However, the adaptability of current NFFs must be measured from the perspective of MCC given its architectural framework, scalability, privacy, accuracy, complexity, overheads, and implementation aspects. These requirements can be met by identifying the differences between current and MCC networks to improve current NFFs with value-added features and make them adaptive for MCC networks. None of the current NFFs is completely adaptable to the MCC network because of the different operational requirements and on-demand services of MCC [8, 88, 94, 98, 102]. The value of the adaptability attribute is defined as “difficult,” “low,” and “moderate” in Table 6. Compatibility between NFFs and MCC networks is nearly impossible to achieve (difficult), but with a few changes in the frameworks and added features, the possibility could improve (moderate) [16]. Nevertheless, adaptability values are marked low when compatibility is between difficult and moderate; this condition reveals the need for major changes to adapt to MCC networks. In most cases, NFFs lack scalability [8, 71, 87, 88, 93, 94, 96, 97], accuracy [62, 85, 88, 91–93, 95, 96, 99–101], and privacy to enhance their adaptability to MCC networks [85, 91, 92].

4.2. Analysis of Existing Network Forensics Frameworks to MCC. This section presents a comprehensive analysis of existing NFFs in terms of their adaptability to MCC networks. Current NFFs are investigated based on the selected evaluation parameters discussed in Section 4.1.

4.2.1. Traceback Based Forensics Frameworks to MCC. The network forensic evidence acquisition (NFEA) framework [71] lacks scalability in terms of MCC because of packet capturing at edge routers in current networks. Identifying edge routers is difficult in MCC because of the seamless connectivity provided by CSPs to cloud users [108]. Determining and accessing the appropriate edge router are challenging tasks for network forensics in MCC. Moreover, computational overhead is high because a three-phase encoding scheme that collects, encrypts, and marks each packet at the edge router is utilized. This scheme requires computational resources to perform encoding for a large number of network packets, especially in MCC. Computational resources are managed in the cloud and accessed through the pay-as-you-go service model. Storage overhead is also high because no technique is incorporated to manage high network traffic. Storage resources in the cloud are utilized to store large numbers of marked network packets. However, payment is required to access the services and resources of computational clouds; thus, utilizing cloud resources is expensive. Accuracy is also low in terms of MCC because NFEA employs a flow-based selection marking scheme to categorize network traffic based on IP attributes [109]. However, network traffic is reduced, resulting in efficient management of large data in MCC. NFEA frameworks are highly complex in terms of implementation. Applying a three-step encoding scheme for each network packet at various network locations in the cloud is difficult. The privacy attribute is low because each packet is captured and marked at the router, which discloses important information on users instead of the attacker. Consequently, reliability decreases, and the number of users is reduced. In conclusion, adapting the NFEA framework for MCC networks is difficult given the abovementioned problems.

Lightweight IP traceback scheme (LWIP) [9] incorporates horizontal scalability with an increasing number of hop counts. Each router inserts a TTL value in the packet field to simplify the investigation process. Large networks have numerous routers that receive packets by setting the TTL value. This feature of LWIP is scalable enough for MCC networks but results in a high computational overhead as a result of analyzing the packets with the tree analysis algorithm utilized by LWIP. The analysis requires time frames with computational resources and thus increases the computational overhead, particularly with trillions of packets flowing through MCC networks. Storage overhead is at a moderate level because LWIP only stores packet information in terms of a packet package, which is utilized to tally with IP addresses. A router only marks the TTL field of the packet and does not store it. LWIP is also moderately accurate because it involves a filtration step to remove irrelevant packets and only mark packets that are a threat. LWIP complexity is high in the analysis phase of MCC because of the tree analysis algorithm. The algorithm requires sufficient packets to reconstruct the path with more time frames, which is difficult because of the seamless connectivity and real-time incidence response requirement in MCC [110]. LWIP reaches a moderate level of privacy because of the marking of packets at the TTL field rather than at the payload field. All these constraints make

TABLE 6: Analysis of network forensics frameworks in context of adaptability to MCC.

Frameworks	Scalability	Overhead		Accuracy	Complexity	Privacy	Adaptability	
		Computational	Storage					
Traceback	NFEA [71]	N/A	H	H	L	IM	L	N/A
	LWIP [9]	HT	H	M	M	AL	M	D
	Scalable NF [16]	VT	L	L	H	IV	L	M
	HB-SST [85]	HT	M	N/A	N/A	IM	N/A	D
	ITP [8]	N/A	H	H	M	IM	M	D
Converge networks	PBNF [87]	N/A	H	L	L	IM, AL	L	D
	VoIP-NFDE [88]	N/A	M	M	N/A	IM, CL, AL	L	D
	VoIPEM [62]	HT	M	M	N/A	IM, AL	L	N/A
Intrusion detection system	AIDF [91]	HT	M	M	N/A	IM, AL	N/A	D
	DFITM [92]	HT	H	L	N/A	IM, AL	N/A	D
	IIFDH [93]	N/A	H	L	N/A	AL	L	D
	NFIDA [94]	N/A	L	L	L	IM, AL	L	D
Attack graph	SA [95]	HT	M	L	N/A	AL	L	L
	MLL-AT [96]	N/A	M	L	N/A	IM, CL, AL	M	L
	AGFE [97]	N/A	H	M	L	IM, AL	M	D
	FCM [98]	HT	M	M	H	AL	M	M
	CSBH [17]	HT	L	L	M	AL	M	H
	AGVI [99]	HT	L	N/A	N/A	AL	H	M
Distribution	ForNet [100]	VT	M	M	N/A	CL, AL	L	M
	DRNIFS [101]	BT	M	M	N/A	AL	L	L
	DCNFM [102]	HT	M	M	L	CL, AL	L	L
	DNF-IA [18]	HT	L	L	L	IM, CL, AL	L	M

Scalability: HT: horizontal; VT: vertical; BT: both; N/A: not applicable.

Overhead: H: high; M: moderate; L: low; N/A: not applicable.

Accuracy: H: high; M: moderate; L: low; N/A: not applicable.

Complexity: IM: implementation; AL: analysis; CL: collection; IV: investigation.

Privacy: H: high; M: moderate; L: low; N/A: not applicable.

Adaptability: D: difficult; H: high; M: moderate; L: low; N/A: not applicable.

the LWIP framework difficult to be adapted for the MCC network.

Scalable network forensics strategy (scalable-NF) [16] adopts vertical scalability resulting from the improved computational power of the forensics server because only infected traffic is investigated. MCC network data can be investigated by analyzing only infected network packets rather than the traffic flows of the entire network. However, the computational overhead is low because of the training phase that generates a normal behavior profile from network traffic through probabilistic inference. Such generation requires computational resources that can be extracted with MCC resources in the cloud. A normal behavior profile contains a set of features with their values that can be stored in different storage resources provided by MCC. Thus, the storage overhead for scalable-NF is low in MCC. The accuracy of scalable-NF is high because infected packets are separated from normal packets, which is favorable for MCC networks. However, the complexity level is high because random moonwalk algorithm is employed in forensics investigation. This algorithm regards the directed host graph as an input and then investigates attack edges. Generating attack graphs

and identifying attack edges are difficult in MCC given its virtualized and distributed environment with frequent service and data migration [2]. The privacy parameter is low in MCC because network traffic is divided into normal and attack traffic (infected), both of which require a thorough analysis of the packets. Therefore, scalable-NF can adapt to MCC networks by improving the analysis phase and increasing the privacy of user data.

Hopping-based spread spectrum technique (HB-SST) [85] provides horizontal scalability in spreading a spectrum over a number of nodes. Large networks can utilize this technique through secure dispersion of spread codes in both time and frequency domains. The computational overhead is moderate owing to the sending of pseudonoise codes with a large number of packets in large networks, particularly in MCC. Computational overhead can be reduced by utilizing MCC resources according to the demand. However, implementing such a framework in MCC is complex because of the use of pseudocodes in high-speed data rate networks that connect data centers, servers, and network devices in different clouds. In MCC, data migrate within numerous devices that require a quick response when an attack occurs in

real time. However, storage, accuracy, and privacy parameters are inapplicable to MCC networks because there are no concerns over the storage and reduction of data in HB-SST. Therefore, adapting this technique to MCC network positions is extremely difficult.

IP traceback protocol (ITP) [8] lacks scalability because of its architectural implementation. ITP has a high computational overhead owing to the large message passing among system, router, data base, and attack analysis managers. Such message passing among various managers increases network latency utilization, which further increases the overall investigation time. The router manager detects an attack packet sent to the sinkhole router that stores and sends it to the system manager. This process creates high storage overhead at the sinkhole router and causes a bottleneck vulnerable to attacks. Each packet is treated at more than one place, resulting in high complexity, particularly for a large network infrastructure such as MCC. Complexity increases when trillions of packets enter the network. Each router then performs static functions, such as hashing, compressing, storing, and diverting packets toward the sinkhole router. ITP is moderately accurate because data are stored in a compressed hash table format by applying filtering to remove irrelevant data. ITP privacy is at a moderate level because only packet headers are targeted for investigation. An attacker can use packet headers to attack networks by altering different packet fields inside the packet. Therefore, adapting ITP for MCC networks with the current proposed framework is difficult.

4.2.2. Adaptability of Converge Network Based Forensics Frameworks to MCC. Pattern-based network forensics (PBNF) [87] incorporates limited scalability because of its framework implementation. The computational overhead is high because evidence is collected from different sensors installed before VoIP components and network forensic analysis tools are utilized to collect raw network traffic data. Collecting such data and sending them to a centralized forensics server consume network bandwidth and creates traffic load at the server. Analysis is performed in real time, and high bandwidth is required to generate an incident response at runtime. Collecting data also requires high storage capacity. PBNF stores voice data at numerous locations but still requires sufficient storage space to address large amounts of data in MCC. Cloud storage resources are suitable for storing voice data that can be further investigated by a forensics server. Thus, the storage overhead generated by capturing voice data in MCC is low for PBNF. Voice data are captured and analyzed for different attack patterns; this procedure decreases user communication privacy. The privacy factor is low for PBNF because the voice packets of a user are monitored and investigated at different locations in a converged network. The complexity level is high in terms of implementation and analysis because data are captured from multiple locations, the monitoring scheme is inefficient, and large volumes of data are analyzed. Thus, the adaptability of PBNF is ranked as difficult for networks in MCC given the aforementioned limitations.

VoIP network forensic analysis with digital evidence procedure (VoIP-NFDE) [88] lacks scalability because of the deployed analysis method. However, the computational overhead is moderate in terms of differentiating normal from abnormal packets in VoIP network traffic. Such differentiation of packets requires computational resources that can be obtained from MCC on a pay-per-demand basis. The computational overhead in traditional networks is high because of the lack of computational resources; however, the computational overhead in the MCC network infrastructure can be reduced by utilizing cloud resources. Comparing trillions of packets per second at multiple locations in MCC requires a large storage capacity for the storage of captured network data without losing their integrity. VoIP-NFDE has a moderate level of storage overhead because MCC storage resources are utilized at a financial cost. Furthermore, network traffic is not reduced by filtering; thus, the VoIP-NFDE framework in MCC lacks accuracy. The complexity of this method's implementation, collection, and analysis in MCC is high because of the static nature of the framework. The privacy level is low because differentiating the packets to expose malicious ones increases the risk of leaking out user information. Thus, VoIP-NFDE is difficult to adapt to large and fast disseminated networks of MCC.

The VoIP evidence model (VoIPEM) [62] contains horizontal scalability in terms of collecting information from different VoIP components in a converged network. The model can be utilized for large network infrastructures by providing information from various VoIP components at the cost of high complexity in implementation and analysis as well as low privacy of user data. Implementation complexity results from the synchronization of VoIPEM modules that collect and send information to generate an evidence module. Analysis complexity is high because of the different scenarios inferred from forward and backward chaining performed by S-TLC modeling. The computational overhead is moderate because evidence from attack scenarios is generated through S-TLC. However, storage space is required to store infected packets, from which a hypothesis is formulated to identify unknown attacks. VoIPEM lacks a proper storage mechanism for traditional networks; however, the model can use the storage resources of the MCC infrastructure. Moreover, VoIPEM does not focus on reducing any type of network data. Hence, its accuracy is inapplicable to both traditional and MCC networks. VoIPEM should thus improve its strategies on computational overhead, accuracy, and privacy to become adaptable for MCC networks.

4.2.3. Adaptability of Intrusion Detection Systems Based Forensics Frameworks to MCC. Analytical intrusion detection framework (AIDF) for distributive IDS [91] includes horizontal scalability in terms of distributed IDS to detect intrusions by generating attack alert messages. AIDF performs by inhibiting negative behavior in which an attack is detected as distributive but fails to report a message. AIDF scalability can be heightened in MCC by installing IDS at various locations in the network to detect malicious behavior in network packets. Enhanced scalability

increases complexity in MCC because of the installation of distributive IDS sensors, collection of attack patterns, and real-time performance analysis. However, AIDF has a moderate computational overhead because of its probabilistic inference that identifies hidden undetected attack patterns for better forensics explanation. Probabilistic inference increases delays when an ignorant sensor identifies matching signature rules from its neighboring sensors. The method incorporates storage overhead because of its storage requirement at the distributed IDS, large network traffic at specific sensor nodes, and storing of network traffic of all sensor nodes at the central location. The AIDF storage overhead with regard to all sensor nodes is at a medium level and high when a single sensor handles a large amount of network traffic. However, the storage overhead can be minimized by using MCC storage resources. Accuracy and privacy parameters are inapplicable to AIDF because it does not reduce network traffic for investigation and employs probabilistic inference for analysis. Therefore, adapting to MCC networks is difficult for AIDF given its existing infrastructure.

An intrusion tolerance system for modeling and analyzing dynamic forensics system (DFITM) was proposed in [92]. The system achieves horizontal scalability by employing two servers to receive packets. Scalability is increased by installing more forensic servers to analyze infected packets for legal evidence in MCC. However, DFITM has a high computational overhead because it employs a formalized method to investigate network packets. DFITM results in more time delays and requires computational resources, especially in a network that contains thousands of nodes, such as MCC. The storage overhead is low as a result of storing network traffic in a normal server and the use of infected network packets in a shadow server to store malicious packets that were investigated to identify the origin of the attack. Storage overhead can be minimized by using MCC storage resources to store all infected packets without overwriting them. However, DFITM has high complexity in terms of its implementation and analysis in the MCC context. The system employs multiple steps to perform analysis. Such steps include generating a security report through the shadow server; the report is sent to the evidence collector and further partitioned, encrypted, and replicated in various evidence databases. Accuracy and privacy are inapplicable to MCC because network traffic is not reduced and the analysis of network packets results in the lack of user privacy. Assessing thousands of nodes, large network channels, and trillions of packets per second as well as the virtualization environment of MCC limit the adaptability of DFITM to such an environment.

Intrusion investigation framework with data hiding schemes (IIFDH) [93] lacks scalability because of its architectural framework. A monitoring module is utilized to monitor trace log files altered by attackers. In MCC, the framework requires numerous monitoring modules because of the millions of trace log files present at disperse locations. IIFDH also lacks accuracy because it does not filter network traffic and consequently increases delays. The framework has a low computational overhead for traditional networks and a high computational overhead for MCC networks. In

MCC, additional working processes are required to embed log sources, trace log files, perform stenography, monitor alteration in the traces, and create backup files in log backups for numerous locations. The storage overhead is high for traditional networks and low for MCC networks. Storing original trace log files in log backups can be easily performed through cloud storage resources. IIFDH has high complexity because of the stenography performed on trace log files by changing the least significant bit. This task can be more challenging in MCC, which can have millions of log files at numerous locations in the cloud. Privacy is low for user data in MCC because monitoring log files can leak out confidential user data. Therefore, IIFDH is difficult to implement for MCC networks because of the non-scalable framework, lack of accuracy, minimal privacy, and high computational overhead of the employed framework.

Network forensics based on intrusion detection analysis (NFIDA) [94] lacks scalability because of its architecture, such as the use of network evidence-capturing engine and network forensics analysis engine. These engines work in sequence to produce digital evidence based on log messages and network data. The engines can be disadvantageous to large networks that require real-time investigation of high-speed data rate network traffic at various locations in the cloud. NFIDA has a high computational overhead in traditional networks as a result of its encryption method during both network data and log message capturing. In MCC, the computational overhead can be reduced by using computation-intensive resources for encryption. The storage overhead is high in a traditional network owing to the network traffic burden on the centralized network forensics analysis engine. The network forensics analysis engine receives log messages and network traffic. Nevertheless, the storage overhead in MCC can be minimized by using storage resources to store log messages and network data. NFIDA has low accuracy because it omits irrelevant data through the network forensics analysis engine. Moreover, NFIDA is complex due to its implementation and analysis phase in MCC. User data privacy is low in NFIDA because packets have to be investigated using the network forensic analysis engine. Making NFIDA adaptable to the MCC network infrastructure is difficult. NFIDA has to be modified in terms of its scalability, accuracy, and privacy before it can be applied to MCC networks.

4.2.4. Adaptability of Attack Graphs Based Forensics Frameworks to MCC. Scalable analysis approach (SA) [95] achieves high horizontal scalability in terms of addressing large amounts of raw security traffic. SA can address millions of nodes present in a network by determining their timespan distribution and dependency on one another. However, in a traditional framework, SA has a high computational overhead, such as timespan distribution and probabilistic temporal attack graph that requires computational resources to generate computation results for attack modeling. However, the computational overhead can be reduced by assigning cloud resources. SA has a medium-level storage overhead in traditional networks when the attack and dependency

graphs increase; a portion of the graphs is stored on the disk rather than in the main memory. The storage overhead can be also minimized by storing a portion of the graphs on cloud storage resources. The complexity of SA in MCC is high when millions of nodes are addressed because the virtualized and distributed setup creates difficulties in analyzing various attack paths in the attack graphs. SA compatibility is low in terms of its adaptability to MCC, and the accuracy factor is inapplicable because the entire network is focused on creating the attack graph. The framework of SA requires modification with regard to accuracy and complexity prior to its application to MCC networks.

Network attack modeling based on multilevel and layer attack tree (MLL-AT) [96] lacks scalability and has a medium level of computational overhead in terms of constructing attack sequences composed of various single attacks. MLL-AT requires computational resources to determine attack sequences and can use cloud resources in the case of MCC. The model has low storage overhead because it only stores attack sequences; storage overhead can be easily minimized by using the storage resources of MCC. Data are not reduced for evaluation and modeling; thus, accuracy is inapplicable to both traditional and MCC networks. MLL-AT has high complexity in terms of implementation, collection, and analysis to address attack sequences in attack trees. MLL-AT is more complex in MCC because it locates risk values for attack sequences, which can be difficult because millions of nodes are present in the MCC network. The method has a medium level of privacy because it has minimal access to user data given that it only identifies the attack sequence. Such privacy level is acceptable for MCC users who wish to keep their information safe from investigators. MLL-AT has a low level of adaptability to MCC with its current framework setup. However, it can be enhanced by modifying its strategies to increase its accuracy and reduce its complexity so that it can be applicable to MCC.

Attack graph for forensic examination (AGFE) [97] lacks the scalability factor because it inserts anti-forensics nodes in the attack graph; this condition results in sophistication owing to the incorporation of anti-forensics nodes into millions of nodes for large networks, such as MCC. AGFE has a high computational overhead because it inserts antiforensics nodes into the attack graph to trace out unexploited attack behavior. Antiforensics nodes are inserted into the attack graph that incorporates computational overhead as a result of the dependency among various nodes. In MCC, a cloud integrates millions of nodes connected with high data rate network links; millions of packets are transferred per second. To trace such nodes, antiforensics nodes should be sufficiently powerful to trace high-speed data rate traffic within the seamless and virtualized connectivity provided by MCC. An antiforensics database also involves overhead that can be solved by utilizing the storage resources of MCC. In addition, AGFE reduces the number of network nodes that are not utilized in the forensics examination. AGFE results in low accuracy because not enough network traffic is filtered. Complexity is high in terms of implementation and analysis stages because of the insertion of the antiforensics nodes and the determination of interdependency among network

nodes in MCC. Privacy in AGFE is at the maximum level because investigators are provided access to data. However, the method lacks scalability; otherwise, it is highly adaptive to MCC networks.

Fuzzy cognitive map (FCM) [98] achieves horizontal scalability by reducing a large number of attack paths to identify the worst one. FCM is useful for MCC, which has a wide range of network links that connect millions of resources. In MCC, the large number of network nodes and paths has to be reduced to identify the worst attack paths in the network and consequently help investigators in their investigation. FCM minimizes the number of attack paths through a genetic algorithm (GA) that identifies the worst attack paths in an attack graph generated for the network. FCM analyzes only relevant attack paths and thus increases accuracy in identifying the worst attack paths in MCC. FCM has a medium-level computational and storage overhead because it employs fuzzy cognitive map processing and stores attack graphs for the network. These attack graphs are further utilized to investigate the worst attack paths in the network. FCM can reduce both computational and storage overhead by using MCC computational and storage resources that are lacking in traditional networks. However, the complexity of FCM is high because of its analysis phase for MCC. The method generates cognitive maps by developing concepts and casual influence for large network nodes; this condition makes the entire process extremely complex. In addition, FCS has a medium level of privacy because user data are not observed and only attack nodes are targeted to develop attack graphs. Such privacy level is suitable for MCC networks to keep user data confidential from third-party investigators. Overall, FCM is adaptive to MCC networks; however, its complexity must be minimized further, particularly when worst attack paths are investigated in real-time scenarios.

Probabilistic approach to identify cost-benefit security hardening (CBSH) [17] incorporates horizontal scalability to perform risk assessment of nodes in large networks. The objective is to determine the cause of an attack in large networks with a large number of nodes. The scalability of CBSH is applicable to the disseminated network of MCC because it assesses risks to identify the cause of an attack. CBSH has low computational and storage overheads because of the application of probabilistic method to perform risk assessment for the cause of the attack without storing an entire network node for investigation. Computation-intensive resources are employed to perform probabilistic inferences, and storage resources are utilized to store targeted attack nodes. However, CBSH accuracy is ranked as medium in the MCC aspect because it fails to reduce all the irrelevant network nodes while performing probabilistic inferences. Privacy is also ranked as medium because it investigates target attack nodes without inferring user data, which increases confidentiality and privacy for users. CBSH employs analysis phase complexity by performing cost-benefit security hardening for large disseminated networks in MCC. Thus, CBSH is applicable to MCC networks; however, reducing unwanted network nodes is necessary for the model to perform real-time probabilistic inferences and produce a quick incident response to various queries.

Visualization and interaction framework for attack graphs (AGVI) [99] contains horizontal scalability in terms of visualizing attack nodes for large networks and helps investigators interact with a visualized interface. Such interface of visualized attack graphs enables investigators to search for a specific attack path in the attack graphs. Visualization and interaction are useful for large networks that contain millions of nodes, especially for MCC. AGVI also provides a facility for visualizing attack paths between two selected vertices on an interface and thus helps investigators during the investigation process. AGVI complexity is high in intercloud networks that should generate an attack path in the network in real-time situations. The model has low complexity in cloud data center networks. The level of privacy is high for users in the MCC infrastructure because AGVI provides visualized attack graphs for infected nodes of the network rather than accessing user data. However, accuracy and storage overhead factors are inapplicable to AGVI in MCC because the network nodes are not reduced and stored. Therefore, AGVI provides more benefits to service providers in their visualization of attack paths in MCC networks.

4.2.5. Adaptability of Distributive Forensics Frameworks to MCC. Distributed framework (ForNet) [100] follows vertical scalability in terms of a highly computational forensics server in MCC. In ForNet, synApps software module is installed at different network devices to collect network traffic and send it to the forensics server. The forensics server should be computation-intensive to investigate network traffic with a quick incident response in real-time situations. This condition can be easily achieved with related MCC resources. ForNet has a medium-level computational overhead because of synApps, which has to be fast enough to cope with fast-moving network traffic. The framework also has medium-level storage overhead because network traffic is diverted toward a centralized forensics server and has to be stored in its database for the investigation of various network vulnerabilities. However, ForNet lacks accuracy because it investigates all network traffic without any reduction through synApps installed at various locations of the network. The complexity of ForNet is viewed in MCC as a result of its collection and analysis phases. Collection is performed with synApps modules that should be capable of capturing entire network traffic while preserving its integrity and reliability for sending to the forensics server. ForNet cannot be applied to collect network traffic from disseminated parts of MCC and cannot be investigated centrally by the forensics server because it causes numerous time delays, low incident responses, and reduction of the entire system performance. Nevertheless, the privacy level is low because all network traffic is captured and investigated centrally by analyzing each of its packets to identify various vulnerabilities. Capturing all network traffic from dispersed network devices and investigating them centrally is a difficult task. The ForNet framework requires modification prior to application in MCC networks. ForNet can be applicable for MCC networks if it contains distribution

storage with decentralized analysis of captured network traffic to reduce time delays, provide a quick incidence response, and increase system performance.

Distributed agent-based real-time network intrusion forensic system (DRNIFS) [101] employs both horizontal and vertical scalability. Horizontal scalability is achieved by installing network agents at the sensitive areas of a large network. Vertical scalability is achieved by providing a highly computational forensics server at disseminated locations. DRNIFS helps collect network traffic from dispersed areas in MCC and provides a quick incident response for queries by performing investigation at various forensics servers. However, DRNIFS has medium-level computational and storage overheads in terms of collecting and storing data that are to be further investigated to identify malicious behavior and reconstruct various attacks. DRNIFS includes log and audit data, network traffic, and a historical network of misused patterns. The computational and storage resources of MCC can be utilized to minimize these overheads and make DRNIFS adaptive for MCC networks. Collecting and investigating various types of data result in high complexity because various data in distributed locations, particularly in real-time situations, are analyzed. The process is time consuming and can delay responses, which are significant in real-time investigations. Additionally, complete network traffic is captured and investigated by various forensics servers; thus, user privacy is reduced. DRNIFS also lacks accuracy in MCC networks. DRNIFS is partially applicable to MCC networks and can be improved by enhancing its accuracy and privacy parameters for network traffic that contains multiple users' information.

Distributed cooperative network forensics model (DCNFM) was proposed in [102]. This model implements horizontal scalability because of its forensics servers. These servers are distributed in the network, collect network traffic, and store data from various client agents in a database format. Several forensics servers are added to the network according to its requirement. To investigate the same type of network traffic (e.g., e-mail traffic), only one network forensics server is required. Although DCNFM can utilize computation-intensive cloud servers, it requires numerous servers because of the millions of data resources and high network traffic in MCC. DCNFM has a medium level of computational overhead because the process involves multiple executions, such as building databases, filtering and dumping traffic streams, converting traffic streams to a database format, mining forensics databases, network-surveying, and visualizing network attacks. The model has storage overhead because it captures the entire network traffic to reconstruct the attack behavior. This storage overhead can be easily addressed in MCC by storing the entire network traffic in multiple storage resources to prevent the data from being overwritten. However, capturing the entire network data reduces the accuracy level of DCNFM in MCC. Nonetheless, the level of accuracy is medium when the model focuses on specific types of network traffic, such as email data. This condition can be achieved at the filter and dump stage of the DCNFM process. DCNFM is complex because it collects and analyzes network traffic in MCC. Validation is required

to show that complete data are captured at various security devices in MCC networks. Correspondingly, the analysis performed at multiple locations requires synchronization of the investigated data at different forensics servers to provide a quick response to attack queries. The privacy level is low because entire data are captured and analyzed for malicious behavior; thus, confidential user data could leak out. As a result, DCNFM has low adaptability to MCC and has to improve its accuracy and privacy parameters. Network data collection must be performed in less time, and investigated data must be synchronized in the various forensics servers.

Dynamical network forensics framework based on immune agent (DNF-IA) [18] integrates horizontal scalability in terms of detector agents distributed in the network. Detector agents are spread across the network to collect network traffic and forward it to the forensics server. The number of detector agents varies depending on the network's requirement. This feature can be useful to MCC in locating detector agents in the sensitive area where network data must be recorded. The number of detector agents depends on the size of the network. The framework has medium-level storage overhead in traditional networks because network traffic is stored locally at various detector agents. However, DNF-IA can minimize the storage overhead by utilizing the storage resources of MCC. Similarly, computational overhead is also high in traditional networks because a single forensics server has to investigate millions of packets and send a quick incident response. The abundant computational resources of computational clouds can minimize the problem through fast analysis and investigation with quick incident responses. DNF-IA lacks accuracy because the entire network traffic is captured and recorded and data are not filtered; thus, DNF-IA is time consuming and problematic, particularly in MCC. DNF-IA has low privacy because it analyzes network traffic captured from various detector agents installed at disseminated locations of the network. The model investigates each network packet by identifying malicious codes that can violate user privacy through access to the packets. The complexity of DNF-IA in MCC is the result of the implementation of detector agents and collection of network traffic from various MCC networks. When, where, and how detector agents should be placed in MCC to capture complete and accurate network traffic have yet to be determined. Collecting complete network packets while preserving their integrity at disseminated locations in the MCC network is also an important task for CSPs; this task has to be verified and validated. Overall, DNF-IA could be made applicable to MCC network investigation; however, its functionality must be enhanced to increase accuracy and privacy for network traffic with improved real-time incident responses. Table 6 provides a comparison of current NFFs based on their adaptability to MCC.

NFFs must be modified in terms of their architecture by incorporating different modules and embedding different strategies prior to their adoption in MCC networks. NFF frameworks generally have computational and storage overheads because of the high-speed data rate and large network traffic in current networks [8, 9, 71, 87, 92, 93, 97]. Additionally, NFFs require computation-intensive resources

to investigate trillions of network packets in MCC because of the high bandwidth, real-time application support, advanced technologies, and various cloud services provided by MCC to their users. Capturing large amounts of network traffic requires a huge storage space to log network packets to be investigated for malicious behavior through different NFFs [16, 63, 98]. Utilizing cloud resources, such as computational and storage resources, can overcome these problems. The accuracy parameter also serves a vital function in the compatibility of current NFFs with MCC networks. Filtering large amounts of network traffic data allows NFFs to readily analyze the specific data and identify digital evidence against an attacker. The accuracy value is low in most NFFs. This condition shows that NFFs lack a filtering mechanism to reduce network traffic during an investigation and make NFFs adaptable for MCC networks [18, 71, 87, 94, 97, 102]. Incorporating a filtering mechanism into the existing architecture of NFFs to handle high-speed data rate and high network traffic in MCC is the most important factor in determining the adaptability of NFFs to MCC. The complexity of NFFs lies in their implementation, collection, and analysis. Implementing current NFFs for MCC networks is subject to constraints in architectural mismatch, scalability, accuracy, and privacy [8, 18, 62, 71, 85, 87, 88, 91, 92, 94, 96, 97]. The entire system is made sophisticated to produce an efficient output for investigating digital evidence in the various networks of MCC. The collection phase of NFFs does not guarantee complete capturing of network traffic and validation and verification of its integrity [18, 88, 96, 100, 102]. Capturing the entire network traffic at various network locations is important in determining the origin of the attack and the attack behavior. The analysis phase for various NFFs involves investigating the entire network traffic captured from the network and results in numerous time delays with less incident responses to investigation queries [9, 17, 18, 62, 87, 88, 91, 95, 99, 101]. This problem can be minimized by assigning distributed computation-intensive servers at various locations in the cloud. Furthermore, many NFFs lack privacy, which is important for MCC to gain users' trust [16, 18, 62, 71, 87, 88, 93–95, 100–102]. Privacy is low when network traffic logs are accessed; access to traffic logs involves access to other user information in the network, which is against the characteristics of MCC. An intelligent framework requires a multitenant environment of MCC to protect user data from being exploited.

In conclusion, NFFs' computational and storage overhead problems can be solved by utilizing the abundant resources of computational clouds. With these resources, NFFs can store network logs in storage resources and utilize computation-intensive resources to investigate network traffic. Similarly, the scalability of NFFs can be increased by capturing, examining, storing, preserving, and analyzing network traffic at disseminated locations in resource-rich computational clouds. Machine learning and data mining techniques should be incorporated to retrieve relevant data to trace the origin of the attacker and his/her malicious activity. Thus, the accuracy of NFFs in MCC can be increased. Complexity would also be reduced if the aforementioned suggestions and solutions are adopted for NFFs in the context of MCC. User privacy

can be increased by adopting artificial intelligence techniques to extract specific user data that have to be investigated accordingly. Such techniques help NFFs become adaptive to MCC networks and identify different vulnerabilities in the network. CSPs will also benefit from the use of current NFFs with slight modifications; FaaS can be provided to users, and additional revenue can be generated.

As discussed earlier, the adaption of existing NFF in MCC is challenging primarily due to the restricted access to the data. It implies that NFF adaption in MCC must depend upon predictive analysis and techniques of artificial intelligence. Artificial intelligence can help in the identification of features that reveal information worthy for analysis. It has been shown that artificial intelligence can be effectively used for information gathering from the distributed networks of the MCC [111]. Explicitly, Artificial Neural Networks (ANN) can be used to analyze network evidence from various databases and online resources connected through high bandwidth networks. Beside ANN, Support Vector Machine (SVM) can be used to find out the patterns in the MCC networks based upon their classification [112]. The SVM trains the data through various intelligent algorithms to identify patterns in the network traffic to help classify network traffic into categories. Furthermore, forensic in MCC can benefit from swarm intelligence as well [113]. Swarm intelligence can predict network attacks by modeling bioinspired algorithms which are designed to solve complex problems. Moreover, fuzzy logic can also be an option to identify network evidence in huge distributed networks of MCC [114]. The networked data in MCC has significantly increased due to steep rise in mobile users. To tackle the networked data, real-time network evidence mechanisms demand accurate approximations of vulnerabilities through fuzzy logic systems. Thus, it has been observed that existing NFF lacks artificial intelligence mechanisms, which can assist network forensics in MCC networks.

5. Issues and Challenges in Network Forensics for MCC

This section describes unresolved issues and research challenges in forensics investigation faced by CSPs in MCC. These issues must be addressed to employ current NFFs in MCC networks. Current NFFs would remain unsuitable for MCC infrastructures until robust methods and approaches are engineered, designed, and incorporated into them. Figure 3 presents the unresolved issues and challenges faced by CSPs in the MCC environment.

5.1. High Speed Data Network. Effective network forensics requires capturing, preserving, examining, and analyzing each event on a single device placed in the entire network [115]. Network forensics helps reconstruct, analyze, and track each incident of exploitation in the network. However, the aforementioned stages are restricted because of the network bandwidth and high-speed data rate in large network channels of MCC. In MCC, various data centers are linked with high-speed fiber optics network channels that send

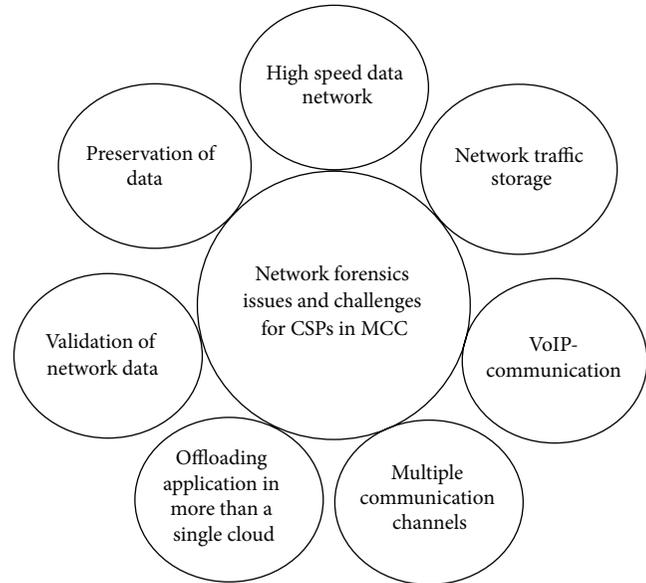


FIGURE 3: Network forensics: issues and challenges for CSPs in MCC.

millions of packets per second [69]. Various data centers link together to form a cloud, which provides services to users in the form of computation, storage, and others. The challenge for CSPs is to capture millions of packets per second from disseminated locations in the cloud in real time while preserving data integrity and reliability [116]. CSPs have to investigate network packets captured at various locations in the cloud to identify different vulnerabilities and trace the origin of an attack. CSPs experience difficulty in capturing, indexing, storing, and analyzing a large number of packets and producing an incident response to forensic queries in a time span of less than a second. In several cases, data have to be retrieved from other clouds; this procedure further delays the investigation process and reduces the quality of service.

Current network forensic analysis tools are unsuitable for high-speed data rate network traffic and cannot produce efficient results in large cloud networks, such as MCC [50–53]. The challenge is to develop an intelligent network forensic analysis tool that can help CSPs investigate network packets and ensure that FaaS outputs are provided in real-time situations for user queries.

5.2. Network Traffic Storage. Large network volumes with high-bandwidth network channels have made network forensics investigation complex and challenging, particularly for MCC networks. MCC is utilized by smartphone users because of data-related constraints [108], such as, integrity, security, preservation, consistency, and storage. Smartphone users send their data to the cloud for storage; they can easily access and retrieve these data any time [1]. Capturing and storing digital evidence with the large amount of network traffic and high-speed data rate in MCC are challenging for CSPs because all these activities are performed in less than a second [117]. Storing network traffic requires a proper storage

mechanism that does not affect data integrity and reliability. Moreover, storage resources in MCC should be selected in such a manner that they are readily accessible for retrieving network logs whenever necessary.

Cloud storage resources minimize the storage overhead for current NFFs in MCC; however, they increase the complexity of retrieving particular network records from numerous distributed storage databases [118]. Millions of network packets are considered and stored in the disseminated storage resources of a cloud; large storage capacity and an optimized searching algorithm are thus required [119]. Data mining techniques can be utilized to solve this problem by retrieving relevant network records from large storage databases that can be easily investigated for various vulnerabilities [120]. Therefore, cloud storage resources increase the adaptability of current NFFs to MCC and minimize their storage overhead. However, much effort is required to formalize the network traffic storage process in the cloud to produce standardized and real-time FaaS to MCC users.

5.3. Voice over Internet Protocol-Communication. Smartphone users utilize Skype, Viper, and other voice chat applications to communicate with other users. Such applications generate VoIP packets, with high priority on quality of services [121]. VoIP communication must be captured in a real-time manner during a forensic investigation. Forensics investigation that checks for latency, jitter, and packet loss has to be performed dynamically for VoIP traffic to identify the malicious behavior of voice packets [62].

However, CSPs face difficulties in collecting voice packets stored in different places and moved from one server to another in a cloud (e.g., registrar, redirect, location, and proxy servers) as well as in signaling gateways and billing systems. Many problems related to the privacy of users arise when accessing such servers to extract voice packets. These servers can be in other cloud territories, which restricts CSPs from investigating voice packets without prior approval because important user information could leak out and related privacy related are noted down in SLA. The entire process is subsequently delayed, and the probability that voice packets would lose their integrity or would be altered by an attacker increases [122]. Such constraints create problems for CSPs to provide full-fledged FaaS in real time to remote users in the MCC paradigm.

5.4. Multiple Communication Channels. Multiple communication channels are the paths available to send data from a mobile device to another destination, such as MCC. These channels increase the throughput of the network by sending and receiving data from and to the user at the same time [123]. CSPs face problems during data investigation when the data are sent from smartphone devices to the cloud using a wireless network with multiple communication channels [124]. CSPs do not have complete access to the network between a smartphone device and the cloud because smartphone users are generally mobile [125]. To provide FaaS to MCC users, CSPs must investigate various vulnerabilities present in cloud access networks, such as the network between

a smartphone user and the cloud. However, CSPs depend on ISPs and other network service providers to investigate the cloud access network with multiple communication channels open to various network susceptibilities.

Moreover, data pass through multiple communication channels via more than one network type because of the mobility of smartphone users. Such an issue produces additional complexity for investigators to trace smartphone user connectivity within various networks and perform live forensics for multiple communication channels. In multiple communication channels, a network packet is affected by interference with other wireless networks and results in incomplete capturing of network packets [126]. Incomplete capturing of network packets does not provide a complete picture of the evidence to identify attackers and their attack behaviors. Consequently, trustworthiness and dependability are required by CSPs and network service providers to investigate the situation in real time through the provision of transparent FaaS to smartphone users in MCC.

5.5. Validation of Network Data. The combination of data centers forms clouds, which are then assigned by CSPs to MCC users for the storage and execution of various applications. Each data center connects with other data centers through high-speed data rate network channels [69]. Each channel has to transfer millions of packets in a second and has to store them in a reliable storage medium without affecting their integrity. However, network forensics encounters difficulties in developing integrity and escalation among distributed network forensic storages in MCC [127]. For instance, user data are stored on two different data centers at two different clouds via connection through high-speed network links; verifying validity and integrity while accessing data scattered between the two different cloud data centers is challenging.

Data validation is important because forensics investigation is performed based on data stored in the data center to identify the origin of the attack and attacker [128]. Each CSP has to ask permission from other CSPs to investigate network resources for possible network susceptibilities. This condition creates problems for CSPs in freely capturing network evidence from intercloud and other networks in the cloud. Accessing evidence from an intercloud network is restricted by cross-border rules or might be delayed by CSPs by not responding in real time. Sometimes, real-time data are required for an investigation; however, such data could also be unavailable in MCC for network forensics [74]. This condition is important for sectors, such as business and health, where real-time situations with quick and accurate responses are necessary. Hence, the validation of network data in cloud computing is a challenge for CSPs in MCC.

5.6. Offloading Application to Multiple Clouds. Highly intensive computational applications in MCC are offloaded for execution to data centers in the cloud that are geographically dispersed through high-speed network links [2]. However, data centers sometimes encounter scarcity of a resource to execute an application; thus, a portion or an entire application

is migrated to the nearest available cloud for execution [1]. Increased application load on a data center causes the migration of applications to different data centers. This condition has to be observed in the investigation to identify various attacks. Hence, CSPs face the tough task of capturing legal evidence of an offloaded application from each network link that might not be accessible because of another cloud's boundaries. The only means to gain access to the intercloud network is to obtain legal permission from the CSP, with the risk of being denied because of user data privacy and confidentiality issues.

Real-time investigation technique is required to capture offloaded network packets in other clouds because of the volatile nature of network traffic. When the link between clouds is disconnected, network data are lost and tracing back the origin of the attack becomes difficult without enough evidence in the network.

5.7. Preservation of Data. Preservation of network data is as important as collecting, examining, and analyzing network traffic to obtain legal evidence against an attacker. Preserving data in the cloud is significant because it provides long-term continuity and usability of digital records for stored network data [129]. Preservation helps in future investigations and can function as a pattern match rule for security devices to manage various attacks in the future [57]. The challenges that CSPs face in data preservation in the cloud include data increase, tendency of data to be lost, changing rules and regulation, data migration, interoperability in the clouds, and lack of authenticity verification. CSPs must employ a formal method to preserve data in the cloud to provide FaaS to the user on demand [130].

A preservation process is required to ensure the extensive availability of network traffic and assists in digital auditing, accessing logs, confidentiality, indexing, data center footprints, security, and high availability. Therefore, advancements in technology guided by legal requirements are required to preserve network data for network forensics investigation.

6. Conclusion and Future Directions

This paper discussed the functions, approaches, and structures of current NFFs. We qualitatively analyzed current NFFs based on selected evaluation parameters in the context of their adaptability to MCC. The findings will benefit CSPs by allowing them to save time and money that might be spent on reinventing the use of novel NFFs for various MCC networks. This study provides motivation to users through newly added services, such as FaaS. Current NFFs involve computational and storage overhead problems because of the limited computing potentials of intermediate network devices. However, MCC utilizes resource-rich computational clouds that can solve these problems and help existing NFFs adapt to MCC networks. Similarly, the scalability of NFFs can be increased by capturing, examining, storing, preserving, and analyzing network traffic at disseminated locations in resource-rich computational clouds. Machine

learning and data mining techniques can be incorporated to retrieve relevant data to be investigated and trace the origin and malicious activity of the attacker. Doing so will increase the accuracy of NFFs for MCC. In addition, complexity will be reduced if the suggestions and solutions presented in this paper are adapted for NFFs in the context of MCC. User privacy can also be improved with the use of artificial intelligence techniques to extract specific user data that have to be investigated. Such methods help NFFs become adaptive for MCC networks and identify different vulnerabilities in the network. CSPs will also benefit from using current NFFs with slight modifications; FaaS would be provided to users, and additional revenue would be generated.

We conclude that new research roadmaps and programs are required to overcome the issues and challenges faced by CSPs. Standardized rules, secure reference models, protocols, trust architectures, legal contemplation, technological development, and a global regularity body should be established. These requirements can be achieved by harmonizing the efforts of industrial experts, academic researchers, and investigators under legal entity bodies. Thus, the significance of network forensics will increase gradually and will provide economical and viable solutions for investigators in identifying digital evidence against attackers in MCC networks.

As a future research direction, network forensics as an open research area (particularly for MCC networks) has to be further explored with newly adaptive frameworks that consider user privacy, data integrity, data confidentiality, data segregation, and many other factors. However, these frameworks should be incorporated with newly developed dynamic cloud-based network forensic tools to cope with the MCC infrastructure. Similarly, various security threats and attacks must be identified and studied to achieve optimum results in MCC network forensics.

List of Acronyms

3G-4G:	Third-fourth generation
AGFE:	Attack graph for forensic examination
AGVI:	Attack graph visualization and interaction
AIDF:	Analytical intrusion detection framework
CBSH:	Cost-benefit security hardening
CFS:	Collaborative forensics scheme
CSP:	Cloud service provider
DCN:	Data center network
DCNFM:	Distributed cooperative network forensics model
DDoS:	Distributed denial-of-service
DFITM:	Dynamic forensics intrusion tolerance method
DNF-IA:	Dynamical network forensics-immune agent
DoS:	Denial-of-service
DRNIFS:	Distributed real-time network intrusion forensics system
FaaS:	Forensics-as-a-services

FCM:	Fuzzy cognitive map
FTK:	Forensics toolkit
GA:	Genetic Algorithm
HB-SST:	Hopping based spread spectrum techniques
HCI:	Human computer interaction
I/O:	Input/output
IDS:	Intrusion detection system
IIFDH:	Intrusion investigation framework data hiding
ITP:	IP Traceback protocol
LTE:	Long-term evaluation
LWIP:	Lightweight IP traceback
MCC:	Mobile cloud computing
MLL-AT:	Multilevel and -layer attack tree
NFAT:	Network forensics analysis tool
NFEA:	Network forensics evidence acquisition
NFF:	Network forensics framework
NFI:	Network forensics investigator
NIST:	National Institute of Standards & Technology
PBNF:	Pattern based network forensics
RAN:	Radio access network
QoS:	Quality of service
RMW:	Random moonwalk
SA:	Scalable analysis
SLA:	Service level agreement
S-TLA+:	Secure Temporal logic of action
TDPM:	Topology assisted deterministic packet marking
TTL:	Time-to-live
VoIP:	Voice over Internet protocol
VoIPeM:	Voice over Internet protocol evidence model
VoIP-NFDE:	VoIP-network forensic with digital evidence
Wi-Fi:	Wireless fidelity
WLAN:	Wireless local area network.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is fully funded by Bright Spark Unit, University of Malaya, Malaysia, and partially funded by Malaysian Ministry of Higher Education under the University of Malaya High Impact Research Grant UM.C/625/1/HIR/MOE/FCSIT/03.

References

- [1] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1294–1313, 2013.
- [2] M. Shiraz and A. Gani, "A lightweight active service migration framework for computational offloading in mobile cloud computing," *The Journal of Supercomputing*, vol. 68, no. 2, pp. 978–995, 2013.
- [3] A. Tzanakaki, M. Anastasopoulos, G. S. Zervas, B. R. Rofoee, R. Nejabati, and D. Simeonidou, "Virtualization of heterogeneous wireless-optical network and IT infrastructures in support of cloud and mobile cloud services," *IEEE Communications Magazine*, vol. 51, no. 8, pp. 155–161, 2013.
- [4] A. Karim, S. A. A. Shah, and R. Salleh, "Mobile botnet attacks: a thematic taxonomy," in *New Perspectives in Information Systems and Technologies*, vol. 2, pp. 153–164, Springer, Berlin, Germany, 2014.
- [5] A. Karim, R. B. Salleh, M. Shiraz, S. A. A. Shah, I. Awan, and N. B. Anuar, "Botnet detection techniques: review, future trends and issues," *Journal of Zhejiang University SCIENCE C*. In press.
- [6] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.
- [7] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks," *Computers and Security*, vol. 24, no. 1, pp. 31–43, 2005.
- [8] E. Jeong and B. Lee, "An IP Traceback Protocol using a Compressed Hash Table, a Sinkhole router and data mining based on network forensics against network attacks," *Future Generation Computer Systems*, vol. 33, pp. 42–52, 2014.
- [9] Y. Fen, Z. Hui, C. Shuang-Shuang, and Y. Xin-Chun, "A lightweight IP traceback scheme depending on TTL," *Procedia Engineering*, vol. 29, pp. 1932–1937, 2012.
- [10] A. C. Kim, W. H. Park, and D. H. Lee, "A study on the live forensic techniques for anomaly detection in user terminals," *International Journal of Security and Its Applications*, vol. 7, no. 1, pp. 181–188, 2013.
- [11] R. Bejtlich, *The Practice of Network Security Monitoring: Understanding Incident Detection and Response*, No Starch Press, 2013.
- [12] S. Gupta, P. Kumar, and A. Abraham, "A profile based network intrusion detection and prevention system for securing cloud environment," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 364575, 12 pages, 2013.
- [13] S. Zawoad and R. Hasan, "Cloud forensics: a meta-study of challenges, approaches, and open problems," <http://arxiv.org/abs/1302.6312>.
- [14] K. Ruan, J. Carthy, T. Kechadi, and I. Baggili, "Cloud forensics definitions and critical criteria for cloud forensic capability: an overview of survey results," *Digital Investigation*, vol. 10, no. 1, pp. 34–43, 2013.
- [15] Y. Wang, Z. Liu, Z. Du, and Y. Huang, "Mobile cloud computing network attack and defense learning system based on fuzzy soft sets," *Procedia Computer Science*, vol. 17, pp. 214–221, 2013.
- [16] L. M. Chen, M. C. Chen, W. Liao, and Y. S. Sun, "A scalable network forensics mechanism for stealthy self-propagating attacks," *Computer Communications*, vol. 36, no. 13, pp. 1471–1484, 2013.
- [17] S. Wang, Z. Zhang, and Y. Kadobayashi, "Exploring attack graph for cost-benefit security hardening: a probabilistic approach," *Computers and Security*, vol. 32, pp. 158–169, 2013.
- [18] D. Wang, T. Li, S. Liu, J. Zhang, and C. Liu, "Dynamical network forensics based on immune agent," in *Proceedings of the 3rd International Conference on Natural Computation (ICNC '07)*, vol. 3, pp. 651–656, IEEE, Haikou, China, August 2007.

- [19] B.-G. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution," in *Proceedings of the 12th Conference on Hot Topics in Operating Systems (HotOS '09)*, pp. 8–11, USENIX Association, Berkeley, Calif, USA, 2009.
- [20] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.
- [21] A. Dou, V. Kalogeraki, D. Gunopulos, T. Mielikainen, and V. H. Tuulos, "Misco: a MapReduce framework for mobile systems," in *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '10)*, p. 32, ACM, June 2010.
- [22] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: elastic execution between mobile device and cloud," in *Proceedings of the 6th ACM EuroSys Conference on Computer Systems (EuroSys '11)*, pp. 301–314, ACM, April 2011.
- [23] H. Qi, M. Shiraz, A. Gani, M. Whaiduzzaman, and S. Khan, "Sierpinski triangle based data center architecture in cloud computing," *The Journal of Supercomputing*, pp. 1–21, 2014.
- [24] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.
- [25] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer Applications*, vol. 40, pp. 325–344, 2014.
- [26] N. Sandlin, "Pay as you go," *Planning*, vol. 55, no. 7, pp. 18–22, 1989.
- [27] Amazon Web Services, "Amazon S3," 2014, <http://aws.amazon.com/s3>.
- [28] D. Houston, "Drop Box," 2014, <http://www.dropbox.com/>.
- [29] Google, "Google Drive," 2014, <https://drive.google.com/>.
- [30] Google, "Google Docs," 2014, <https://docs.google.com/>.
- [31] M. Benioff, "Sales Force," 2014, <http://www.salesforce.com/>.
- [32] S. Funke, S. Laue, Z. Lotker, and R. Naujoks, "Power assignment problems in wireless communication: covering points by disks, reaching few receivers quickly, and energy-efficient travelling salesman tours," *Ad Hoc Networks*, vol. 9, no. 6, pp. 1028–1035, 2011.
- [33] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14–15, pp. 2826–2841, 2007.
- [34] W. Yue, K. Miyazaki, and X. Deng, "Optimal channel assignment in wireless communication networks with distance and frequency interferences," *Computer Communications*, vol. 27, no. 16, pp. 1661–1669, 2004.
- [35] G. Sharma, S. Bala, and A. K. Verma, "Security frameworks for wireless sensor networks-review," *Procedia Technology*, vol. 6, pp. 978–987, 2012.
- [36] B. Schneier, *Secrets and Lies: Digital Security in a Networked World*, John Wiley & Sons, 2011.
- [37] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A survey of mobile malware in the wild," in *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '11)*, pp. 3–14, October 2011.
- [38] E. Casey, *Handbook of Digital Forensics and Investigation*, Elsevier, 2009.
- [39] I. Ray and N. Poolsapassit, "Using attack trees to identify malicious attacks from authorized insiders," in *Computer Security—ESORICS 2005*, pp. 231–246, Springer, Berlin, Germany, 2005.
- [40] G. Palmer, "A road map for digital forensic research," in *Proceedings of the 1st Digital Forensic Research Workshop*, pp. 27–30, Utica, NY, USA.
- [41] M. Reith, C. Carr, and G. Gunsch, "An examination of digital forensic models," *International Journal of Digital Evidence*, vol. 1, no. 3, pp. 1–12, 2002.
- [42] B. Carrier and E. H. Spafford, "Getting physical with the digital investigation process," *International Journal of Digital Evidence*, vol. 2, no. 2, pp. 1–20, 2003.
- [43] B. Carrier and E. H. Spafford, "An event-based digital forensic investigation framework," in *Digital Forensic Research Workshop*, 2004.
- [44] N. L. Beebe and J. G. Clark, "A hierarchical, objectives-based framework for the digital investigations process," *Digital Investigation*, vol. 2, no. 2, pp. 147–167, 2005.
- [45] P. Stavroulakis and M. Stamp, *Handbook of Information and Communication Security*, Springer, Heidelberg, Germany, 2010.
- [46] M. Pollitt, "Computer forensics: an approach to evidence in cyberspace," in *Proceedings of the National Information Systems Security Conference*, pp. 487–491, 1995.
- [47] A. Valjarevic and H. S. Venter, "Harmonised digital forensic investigation process model," in *Proceedings of the Conference on Information Security for South Africa (ISSA '12)*, pp. 1–10, IEEE, Johannesburg, South Africa, August 2012.
- [48] K. Kent, S. Chevalier, T. Grance, and H. Dang, *Guide to Integrating Forensic Techniques into Incident Response*, National Institute of Standards and Technology (NIST), 2006.
- [49] J. Cosic and M. Baca, "Do we have full control over integrity in digital evidence life cycle?" in *Proceedings of the 32nd International Conference on Information Technology Interfaces (ITI '10)*, pp. 429–434, IEEE, June 2010.
- [50] Guidance Software, "Encase," 2014, <http://www.guidancesoftware.com/>.
- [51] Access Data, "Forensics Toolkit," 2014, <http://www.accessdata.com/products/digital-forensics/ftk>.
- [52] B. Carrier, "The Sleuth Kit," 2014, <http://www.sleuthkit.org/index.php>.
- [53] e-fense Carpe Datum, "Helix," 2014, <http://www.e-fense.com/index.php>.
- [54] C. Boyd and P. Forster, "Time and date issues in forensic computing—a case study," *Digital Investigation*, vol. 1, no. 1, pp. 18–23, 2004.
- [55] B. Carrier, *File System Forensic Analysis*, Addison-Wesley, Boston, Mass, USA, 2005.
- [56] W. Dou, Q. Chen, and J. Chen, "A confidence-based filtering method for DDoS attack defense in cloud environment," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1838–1850, 2013.
- [57] S. Mizoguchi, K. Takemori, Y. Miyake, Y. Hori, and K. Sakurai, "Traceback framework against botmaster by sharing network communication pattern information," in *Proceedings of the 5th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS '11)*, pp. 639–644, July 2011.
- [58] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie, "Cloud forensics: an overview," in *Proceedings of the IFIP International Conference on Digital Forensics*, vol. 7, 2011.
- [59] R. W. Ritchey and P. Ammann, "Using model checking to analyze network vulnerabilities," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 156–165, IEEE, May 2000.

- [60] N. Kumar and S. Sharma, "Study of intrusion detection system for DDoS attacks in cloud computing," in *Proceedings of the 10th International Conference on Wireless and Optical Communications Networks (WOCN '13)*, pp. 1–5, Bhopal, India, 2013.
- [61] P. Casas, J. Mazel, and P. Owezarski, "Steps towards autonomous network security: unsupervised detection of network attacks," in *Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS '11)*, pp. 1–5, Paris, France, February 2011.
- [62] M. Ibrahim, M. T. Abdullah, and A. Dehghantanha, "VoIP evidence model: a new forensic method for investigating VoIP malicious attacks," in *Proceedings of the International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec '12)*, pp. 201–206, June 2012.
- [63] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *Computer Security—ESORICS 2009*, pp. 355–370, Springer, New York, NY, USA, 2009.
- [64] E. Chan, S. Venkataraman, F. David, A. Chaugule, and R. Campbell, "Forenscope: a framework for live forensics," in *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC '10)*, pp. 307–316, December 2010.
- [65] S. Sitaraman and S. Venkatesan, "Computer and network forensics," in *Digital Crime and Forensic Science in Cyberspace*, P. Kanellis and E. Kiountouzis, Eds., chapter 3, pp. 55–74, Idea Group Publishing, 2006.
- [66] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using WiFi," in *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '10)*, pp. 209–222, ACM, June 2010.
- [67] K. Asatani, "Carrier-grade networks toward the future—NGN and its issues," in *Green Communication and Networking*, vol. 113 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 1–15, Springer, New York, NY, USA, 2013.
- [68] B.-S. P. Lin, W.-H. Tsai, C. C. Wu, P. H. Hsu, J. Y. Huang, and T.-H. Liu, "The design of cloud-based 4G/LTE for mobile augmented reality with smart mobile devices," in *Proceedings of the IEEE 7th International Symposium on Service-Oriented System Engineering (SOSE '13)*, pp. 561–566, Redwood City, Calif, USA, March 2013.
- [69] S. Yoo, "Intra- and inter-data center networking architectures for future cloud computing," in *Proceedings of the Asia Communications and Photonics Conference (ACP '12)*, p. 1, IEEE, Guangzhou, China, 2012.
- [70] M. Mechtri, I. Houidi, W. Louati, and D. Zeghlache, "SDN for inter cloud networking," in *Proceedings of the IEEE SDN for Future Networks and Services (SDN4FNS '13)*, pp. 1–7, IEEE, Trento, Italy, November 2013.
- [71] H. S. Kim and H. K. Kim, "Network forensic evidence acquisition (NFEA) with packet marking," in *Proceedings of the 9th IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops (ISPAW '11)*, pp. 388–393, Busan, Republic of Korea, May 2011.
- [72] H. Suo, Z. Liu, J. Wan, and K. Zhou, "Security and privacy in mobile cloud computing," in *Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC '13)*, pp. 655–659.
- [73] H. Guo, B. Jin, and T. Shang, "Forensic investigations in cloud environments," in *Proceedings of the International Conference on Computer Science and Information Processing (CSIP '12)*, pp. 248–251, IEEE, August 2012.
- [74] D. Birk, "Technical challenges of forensic investigations in cloud computing environments," in *Proceedings of the Workshop on Cryptography and Security in Clouds*, pp. 1–6, 2011.
- [75] G. Grispos, T. Storer, and W. B. Glisson, "Calm before the storm: the challenges of cloud computing in digital forensics," *Emerging Digital Forensics Applications for Crime Detection, Prevention, and Security*, vol. 4, no. 2, pp. 28–48, 2011.
- [76] M. Shiraz, E. Ahmed, A. Gani, and Q. Han, "Investigation on runtime partitioning of elastic mobile applications for mobile cloud computing," *The Journal of Supercomputing*, vol. 67, no. 1, pp. 84–103, 2014.
- [77] D. Kliazovich, P. Bouvry, and S. U. Khan, "DENS: data center energy-efficient network-aware scheduling," *Cluster Computing*, vol. 16, no. 1, pp. 65–75, 2013.
- [78] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie, "Cloud forensics," in *Advances in Digital Forensics VII*, pp. 35–46, Springer, 2011.
- [79] D. Reilly, C. Wren, and T. Berry, "Cloud computing: pros and cons for computer forensic investigations," *International Journal Multimedia and Image Processing*, vol. 1, pp. 26–34, 2011.
- [80] J. Dykstra and A. T. Sherman, "Understanding issues in cloud forensics: two hypothetical case studies," in *Proceedings of the Conference on Digital Forensics Security and Law (ADFSL '11)*, pp. 191–206, 2011.
- [81] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*, O'Reilly, 2009.
- [82] S. Zawood and R. Hasan, "I have the proof: providing proofs of past data possession in cloud forensics," in *Proceedings of the ASE International Conference on Cyber Security (CyberSecurity '12)*, pp. 75–82, Alexandria, Va, USA, December 2012.
- [83] S. D. Wolthusen, "Overcast: forensic discovery in cloud environments," in *Proceedings of the 5th International Conference on IT Security Incident Management and IT Forensics (IMF '09)*, pp. 3–9, IEEE, September 2009.
- [84] M. Taylor, J. Haggerty, D. Gresty, and R. Hegarty, "Digital evidence in cloud computing systems," *Computer Law & Security Review*, vol. 26, no. 3, pp. 304–308, 2010.
- [85] W. Yu, X. Fu, E. Blasch et al., "On effectiveness of hopping-based spread spectrum techniques for network forensic traceback," in *Proceedings of the 14th IEEE ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD '13)*, pp. 101–106, 2013.
- [86] T. Akyuz and I. Sogukpinar, "Packet marking with distance based probabilities for IP traceback," in *Proceedings of the 1st International Conference on Networks and Communications (NetCoM '09)*, pp. 433–438, IEEE, Chennai, India, December 2009.
- [87] J. C. Pelaez and E. B. Fernandez, "VoIP network forensic patterns," in *Proceedings of the 4th International Multi-Conference on Computing in the Global Information Technology (ICCGI '09)*, pp. 175–180, IEEE, August 2009.
- [88] I.-L. Lin, Y.-S. Yen, B.-L. Wu, and H.-Y. Wang, "VoIP network forensic analysis with digital evidence procedure," in *Proceedings of the 6th International Conference on Networked Computing and Advanced Information Management (NCM '10)*, pp. 236–241, IEEE, August 2010.
- [89] P. Park, *Voice over IP Security*, Pearson Education, 2008.
- [90] H.-M. Hsu, Y. S. Sun, and M. C. Chen, "A collaborative forensics framework for VoIP services in multi-network environments," in *Intelligence and Security Informatics*, pp. 260–271, Springer, 2008.

- [91] B. K. Sy, "Integrating intrusion alert information to aid forensic explanation: an analytical intrusion detection framework for distributive IDS," *Information Fusion*, vol. 10, no. 4, pp. 325–341, 2009.
- [92] L. Chen, Z. Li, C. Gao, and Y. Liu, "Modeling and analyzing dynamic forensics system based on intrusion tolerance," in *Proceedings of the IEEE 9th International Conference on Computer and Information Technology (CIT '09)*, pp. 230–235, October 2009.
- [93] Y.-T. Fan and S.-J. Wang, "Intrusion investigations with data-hiding for computer log-file forensics," in *Proceedings of the 5th International Conference on Future Information Technology (FutureTech '10)*, pp. 1–6, IEEE, May 2010.
- [94] L. Jiang, G. Tian, and S. Zhu, "Design and implementation of network forensic system based on intrusion detection analysis," in *Proceedings of the International Conference on Control Engineering and Communication Technology (ICCECT '12)*, pp. 689–692, Liaoning, China, December 2012.
- [95] M. Albanese, S. Jajodia, A. Pugliese, and V. Subrahmanian, "Scalable analysis of attack scenarios," in *Computer Security—ESORICS 2011*, vol. 6879 of *Lecture Notes in Computer Science*, pp. 416–433, Springer, New York, NY, USA, 2011.
- [96] Y. Fen, Y. Xinchun, and H. Hao, "An network attack modeling method based on MLL-AT," *Physics Procedia C*, vol. 24, pp. 1765–1772, 2012.
- [97] C. Liu, A. Singhal, and D. Wijesekera, "Using attack graphs in forensic examinations," in *Proceedings of the 7th International Conference on Availability, Reliability and Security (ARES '12)*, pp. 596–603, IEEE, Prague, Czech Republic, August 2012.
- [98] A. Diamah, M. Mohammadian, and B. M. Balachandran, "Network security evaluation method via attack graphs and fuzzy cognitive maps," in *Intelligent Decision Technologies*, pp. 433–440, Springer, 2012.
- [99] Z. Harbort, G. Louthan, and J. Hale, "Techniques for attack graph visualization and interaction," in *Proceedings of the 7th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW '11)*, p. 74, ACM, October 2011.
- [100] K. Shanmugasundaram, N. Memon, A. Savant, and H. Bronnimann, "ForNet: a distributed forensics network," in *Computer Network Security*, pp. 1–16, Springer, 2003.
- [101] W. Ren and H. Jin, "Distributed agent-based real time network intrusion forensics system architecture design," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA '05)*, pp. 177–182, IEEE, March 2005.
- [102] W. Ren, "On a reference model of distributed cooperative network, forensics system," in *Proceedings of the 6th International Conference on Information Integration and Web-Based Applications Services (iiWAS '04)*, 2004.
- [103] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.
- [104] B. Stephens, A. L. Cox, S. Rixner, and T. S. E. Ng, "A scalability study of enterprise network architectures," in *Proceedings of the 7th ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS '11)*, pp. 111–121, Brooklyn, NY, USA, October 2011.
- [105] M. Michael, J. E. Moreira, D. Shiloach, and R. W. Wisniewski, "Scale-up x scale-out: a case study using nutch/Lucene," in *Proceedings of the 21st International Parallel and Distributed Processing Symposium (IPDPS '07)*, pp. 1–8, IEEE, Long Beach, Calif, USA, March 2007.
- [106] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, "Dynamically scaling applications in the cloud," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 45–52, 2011.
- [107] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 369–392, 2014.
- [108] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.
- [109] B. Claise, "Cisco systems NetFlow services export version 9," 2004.
- [110] A. Gani, G. M. Nayeem, M. Shiraz, M. Sookhak, M. Whaiduzzaman, and S. Khan, "A review on interworking and mobility techniques for seamless connectivity in mobile cloud computing," *Journal of Network and Computer Applications*, vol. 43, pp. 84–102, 2014.
- [111] E. S. Pilli, R. C. Joshi, and R. Niyogi, "Network forensic frameworks: survey and research challenges," *Digital Investigation*, vol. 7, no. 1-2, pp. 14–27, 2010.
- [112] Y. Zhang, N. Meratnia, and P. J. M. Havinga, "Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1062–1074, 2013.
- [113] S. Pushp, T. H. Cho, J. Han, D. Lee, J. Song, and S. Choi, "An efficient way to track peers in mobile P2P network," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom '12)*, pp. 431–434, ACM, August 2012.
- [114] N. Liao, S. Tian, and T. Wang, "Network forensics based on fuzzy logic and expert system," *Computer Communications*, vol. 32, no. 17, pp. 1881–1892, 2009.
- [115] B. Cusack and M. Alqahtani, *Acquisition of Evidence from Network Intrusion Detection Systems*, 2013.
- [116] L. Slusky, P. Partow-Navid, and M. Doshi, "Cloud computing and computer forensics for business applications," *Journal of Technology Research*, vol. 3, p. 1, 2012.
- [117] R. G. Clegg, M. S. Withall, A. W. Moore et al., "Challenges in the capture and dissemination of measurements from high-speed networks," *IET Communications*, vol. 3, no. 6, pp. 957–966, 2009.
- [118] G. Chockler, D. Dobre, and A. Shraer, "Brief announcement: consistency and complexity tradeoffs for highly-available multi-cloud store," in *Proceedings of the the International Symposium on Distributed Computing (DISC '13)*, 2013.
- [119] L. Zhou, K. He, X. Sheng, and B. Wang, "A survey of data management system for cloud computing: models and searching methods," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 6, no. 2, pp. 244–248, 2013.
- [120] J. Xu, Y. Yu, Z. Chen et al., "MobSafe: cloud computing based forensic analysis for massive mobile applications using data mining," *Tsinghua Science and Technology*, vol. 18, no. 4, 2013.
- [121] I. Sadeh, S. Kitanov, and D. Davcev, "Application of multiple description coding for adaptive QoS mechanism for mobile cloud computing," *International Journal of Cloud Computing and Services Science*, vol. 2, no. 6, 2014.
- [122] Y. Lin, Y. Yen, and P. Wu, "Primary research on VoIP security threat vulnerability and attack prevention," in *ISMAD*, 2009.
- [123] D. S. Chan, T. Berger, and L. Tong, "Carrier sense multiple access communications on multipacket reception channels: theory and applications to IEEE 802.11 wireless networks," *IEEE Transactions on Communications*, vol. 61, no. 1, pp. 266–278, 2013.

- [124] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 34–44, 2013.
- [125] X. Wu, K. N. Brown, and C. J. Sreenan, "Analysis of smartphone user mobility traces for opportunistic data collection in wireless sensor networks," *Pervasive and Mobile Computing*, vol. 9, pp. 881–891, 2013.
- [126] N. Chilamkurti, S. Zeadally, and H. Chaouchi, *Next-Generation Wireless Technologies: 4G and Beyond*, Springer, London, UK, 2013.
- [127] A. Lee and T. Brewer, "Guidelines for smart grid cyber security: vol. 1, smart grid cyber security strategy, architecture, and high-level requirements," NISTIR, 2010.
- [128] J. Dykstra and A. T. Sherman, "Acquiring forensic evidence from infrastructure-as-a-service cloud computing: exploring and evaluating tools, trust, and techniques," *Digital Investigation*, vol. 9, pp. S90–S98, 2012.
- [129] M. Afanasyev, T. Kohno, J. Ma et al., "Privacy-preserving network forensics," *Communications of the ACM*, vol. 54, no. 5, pp. 78–87, 2011.
- [130] A. Waqar, A. Raza, H. Abbas, and M. K. Khan, "A framework for preservation of cloud users' data privacy using dynamic reconstruction of metadata," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 235–248, 2013.

Research Article

A Model Based Security Testing Method for Protocol Implementation

Yu Long Fu¹ and Xiao Long Xin²

¹LIUPPA, University of PAU and Academy of Bordeaux, 40000 Mont de Marsan, France

²Department of Mathematics, Northwest University, Xi'an 710069, China

Correspondence should be addressed to Xiao Long Xin; xlxinx@126.com

Received 11 March 2014; Accepted 23 May 2014; Published 6 July 2014

Academic Editor: Fei Yu

Copyright © 2014 Y. L. Fu and X. L. Xin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The security of protocol implementation is important and hard to be verified. Since the penetration testing is usually based on the experience of the security tester and the specific protocol specifications, a formal and automatic verification method is always required. In this paper, we propose an extended model of IOLTS to describe the legal roles and intruders of security protocol implementations, and then combine them together to generate the suitable test cases to verify the security of protocol implementation.

1. Introduction

Since the former employee of CIA, Edward J. Snowden, reveals the global surveillance of National Security Agency (NSA) [1], the suspicions over the existing security mechanisms and products spread over the world. As a consequence, the cryptographic scientists may need to reverify the popular cryptography methods and protocol designs (RSA, DES, etc.). The documents revealed from Snowden also imply the back-doors may exist in some popular network devices/services, such as the productions from CISSCO, IBM, Microsoft, and Google, which recalls the importance of the security verification over the network productions (including hardware and software if the interactions between the possible intruders and the protocol implementations will not result in any leakage of privacy, the protocol implements are considered as secure). This is because, even when the security protocols are designed perfectly, the back-doors may be also proposed during the implementing of the productions and may make the privacies leak to some potential attackers. In this paper, we are concerned with this important security problem and propose the model and algorithms to verify the security of protocol implementations automatically.

Security protocols are communication protocols, which guarantee the securities (authentication or confidentiality) by

the defined rules and cryptography methods [2]. A protocol implementation is a solid solution of the corresponding protocol specification, which can be one or a part of software/hardware, and is supposed to work as the designs of the protocol. The implementations of security protocol are usually separated into different network devices and are in charge of the security functions to guarantee the communication security. As the importance to those implementations, those declared security functions should be well verified. However, the classic verification method of protocol implementation, the *protocol testing* [3] methods, is not suitable for security verifications. Although several methods have been proposed in the literature [4–6], the problem of security verification over protocol implementations is still an open problem.

Related Works. Recently, the problems of model based security verifications on protocol implementations have attracted more and more attention of academy. Based on the developing security reasoning theorems (PCL [7], Pi-Calculus [8], and HLP SL [9]) and the corresponding automatic model checking approaches (the methods described in [10–12]), the authors of [4] came up with an approach to verify the security of protocol implementation. They use model checking methods to automatically generate the test cases, which contain the possible security flaws of the protocol,

and then execute those test cases on the implementations to verify their security. Although the presented approach fills the gaps between “security protocol analysis” and “protocol implementation verification” to some degree, several problems are still noticed. First of all, the presented model checker is based on ASlan, which is a specific protocol description language, and makes the translation to other model checkers become difficult. Secondly, the generated test cases only contain the counterexamples related to the detected security flaws; other possible messages sequences which can also cause the insecurity (e.g., the message sequences contain some wrong messages formats) are not included. In [5], the authors proposed a mutation based test generation method for security protocol implementation verification. They propose several mutation rules over the conformed messages sequences of the protocol specification to construct the possible counterexamples. Although the concepts of mutations and fuzzy sets [13] are quite useful to simulate the artificial intelligence of computer network, it is difficult to decide the completeness of the proposed mutator.

Paper Contribution. In this paper, by investigating the features of security protocols and the limitations of the existing protocol testing methods, we propose a security extension of Input/Output Transition System to model and test the protocol implementation. The proposed model inherits the clarity of finite automata and can describe the security properties and most of the protocol behaviors with the definition of transition. We also consider the possible network intruder as one part of the system and propose a model to simulate the attack actions. Then, by the proposed models, the complex security protocol actions can be presented in the way of sequence transitions of the reachable graphic model. The possible test cases can be generated through the model with the proposed algorithm.

The following sections are organized as follows. In Section 2, we briefly introduce the used protocol testing method IOLTS and some testing theories over it. In Section 3, from the requirements of security protocol, we analyze the limitations of the existed IOLTS model and present an extended model of IOLTS under some required assumptions. Then, in Section 4, those modeled protocol components are combined as one system and an intruder model is proposed to simulate the malicious attack actions. The test approach and test case generation algorithms are also presented in this section. And finally, in Section 5, we conclude this paper and introduce some future works.

2. Protocol Testing Theory

Protocol testing represents a set of automata based modeling and testing methods to verify the correctness of the protocol implementations, which has been proposed and widely accepted in the industries since Gonenc [14] found the distinguishing sequences in 1970 [3]. Through the serious preset inputs, the unobservable system status and some internal actions can be deduced by checking the corresponding outputs; then the protocol implementations can be verified by the test execution. This verification method has been used to

solve many problems, such as conformance [15], robustness [16], web service security [17], and cloud computing security [18].

There are several models existing in protocol testing methods. But in this paper, we only consider the model of IOLTS. An IOLTS takes the system behavior as labels and serves as a semantic model for various formal specification languages [19]. Normally a finite IOLTS system can be described as a 4-tuple array.

Definition 1 (labeled transition system). A labeled transition system is a 4-tuple array $\langle S, L, T, s_0 \rangle$ where

- (i) S is a finite set of states;
- (ii) L is a finite set of labels, which contains two subsets: input labels L_I and output labels L_U ;
- (iii) T is the transition relation, where $T \subseteq S \times (L \cup \{\tau\}) \times S$;
- (iv) s_0 is the initial state.

If $\exists a \in L_I$, then we write $?a$ to represent that a is an input label; if $\exists b \in L_U$, then we write $!b$ to identify that b is an output label; we have $L_I \cap L_U = \emptyset$, $L_I \cup L_U = L$. If the input and output labels are not identified, the model is labeled transition system (LTS). τ denotes the internal and unobservable (to the tester) actions of the system. A trace means a finite sequence of the observable actions; they are going to be used as test sequences to verify the system. If $\sigma \in L^*$ ($L^* = L \cup \{\tau\}$), then $|\sigma|$ denotes the length of trace of σ . If $q \in S$, then $Out(q)$ denotes all the output labels from q , $In(q)$ denotes all the input labels to q , and $Out(S, \sigma)$ denotes the output of S after σ . We denote the class of all IOLTS over L_I and L_U by $IOLTS(L_I, L_U)$ or $LTS(L_I \cup L_U)$.

Definition 2. Let $P = \langle S, L, T, s_0 \rangle$ be a labeled transition system, s and $s' \in S$, and let $\mu_i \in L \cup \{\tau\}$, $a_i \in L$, and $\sigma \in L^*$; then we have the following notations:

$$\begin{aligned}
 s &\xrightarrow{\mu} s' =_{def} \exists t \in T, t = (s, \mu, s'), \\
 &\quad t.pre = s, \quad t.nex = s', \quad t.lab = \mu \\
 s &\xrightarrow{\mu_1 \dots \mu_n} s' =_{def} \exists s_0, \dots, s_n : \\
 &\quad s = s_0 \xrightarrow{\mu_1} s_1 \xrightarrow{\mu_2} \dots \xrightarrow{\mu_n} s_n = s' \\
 s &\xrightarrow{\mu_1 \dots \mu_n} =_{def} \exists s' : s \xrightarrow{\mu_1 \dots \mu_n} s' \\
 s &\xrightarrow{\epsilon} s' =_{def} s = s' \text{ or } s \xrightarrow{\tau \dots \tau} s' \\
 s &\xrightarrow{a} s' =_{def} \exists s_1, s_2 : s \xrightarrow{\epsilon} s_1 \xrightarrow{a} s_2 \xrightarrow{\epsilon} s' \\
 s &\xrightarrow{a_1 \dots a_n} s' =_{def} \exists s_0 \dots s_n : \\
 &\quad s = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s_n = s' \\
 s &\xrightarrow{\sigma} =_{def} \exists s' : s \xrightarrow{\sigma} s'
 \end{aligned}$$

$$\begin{aligned} trace(p) &=_{def} \{ \sigma \in L^* \mid p \xRightarrow{\sigma} \} \\ init(p) &=_{def} \{ a \in L \mid p \xRightarrow{a} \}. \end{aligned} \quad (1)$$

A test case te is a specification for the behaviours of a tester, which exists in an experiment to be carried out with an implementation under test (IUT). Such behaviour, like other behaviours, can be specified by an IOLTS [20]. Then the test case t can be described as an IOLTS over some specific action set L_I and L_U and with a test verdict ($pass$ or $fail$) in the end ($te : i \rightarrow \{fail, pass\}, i \in IOLTS(L_I, L_U)$). In other words, a test case is a set of transition sequence end with test verdict. Generally, the considered actions of the test cases must relate to the specific protocol specification.

3. Transition Models on Security Protocols

Although the IOLTS based testing methods have been proved to be useful in verifying the protocol implementations, it cannot be used to verify the correctness of security protocol implementations, because of some important features of security protocol. Comparing with the normal system protocols, security protocols have the following important features which stop the modeling by IOLTS.

- (i) Security properties: security protocols usually need to consider security properties such as nonce and session id to help the security functions to identify the security. Those security properties are usually the contents of the exchanging or received messages and are not considerable in IOLTS system.
- (ii) Security checking functions: security protocols use several predefined security rules (e.g., comparing the received nonce with the holding one) to verify the security, which are called security checking functions here. Those fundamental security functions are hard to model by IOLTS. Someone may say that the internal action τ of IOLTS can present those functions, but considering that the security properties cannot be distinguished, an input of IOLTS will trigger multiply checking functions and it is meaningless to the testing.
- (iii) Multiple roles: security protocols naturally contain multiple roles (at least one initiator and one responder), while the IOLTS is designed for one system component.
- (iv) Intruder: most of the security protocols are designed to avoid the attacks from intruder, which is considered as participation of protocol. Notice that this intruder is not presented in the security protocol specification, and it is impossible to propose a model with IOLTS.

In order to describe the required features, we propose extensions of IOLTS with several considerations. To identify the security properties, we define the security properties as “partial label” of the inputs actions and define “parsing

state” and “combination state” to construct the transition. This transition then is going to trigger the security checking functions. Secondly, we impact the way of IO automata [21], which defines the internal actions τ with different suffix to identify different internal actions to this extended model. The security checking functions which are triggered by the proposed “partial label transition” can be identified. The methods of solving the problems of “multiple roles” and “intruder” are going to be presented in the next section. Then, we give a model of security extended labeled transition system (SE-LTS) model as follows.

Definition 3. (SE-LTS) A security extended labeled transition system is an extension of IOLTS for the components of security protocol; it is a 4-tuple array $\langle s_0, S_{se}, L_{se}, T_{se} \rangle$, where

- (i) s_0 is the initial state;
- (ii) S_{se} is a set of states and $S_{se} = S_n \cup S_p$, where S_p is a set of “parsing states,” which parses the received input action into partial labels and S_n represents the set of other normal states;
- (iii) L_{se} is a finite set of labels and $\{l \in L_{se} \mid l = l_0 \cdot l_1 \cdots l_n\}$, where l_n is one partial label of l and \cdot represents the function of concatenation. We define L_p as a finite set of partial labels of all labels in L_{se} ; then $\forall l_n \in L_p$. The unobservable internal actions are marked as τ_n , which indicates different internal actions by the suffix. The set of internal actions is marked as Θ ;
- (iv) T_{se} is the transition relation, $T_{se} \subseteq \{S_n \times L_{se} \times S_p\} \cup \{S_p \times L_p \times S_n\} \cup \{S_n \times \Theta \times S_c\} \cup \{S_c \times L_{se} \times S_n\}$.

A partial label is one part of system action which is sufficient to trigger an internal action of the system. A system action may contain one or multiple partial labels.

Definition 4. Let $P = \langle s_0, S_{se}, L_{se}, T_{se} \rangle$ be a SE-LTS with $s, s' \in S_{se}$, and let $\mu_i \in L_{se} \cup \Theta$, $a_i \in L_{se}$, and $a_i = \{a_{i0} \cdot a_{i1} \cdots a_{in}\}$; we have the following notations:

$$\begin{aligned} s &\xrightarrow{\mu} s' =_{def} \exists t \in T, \\ t &= (s, \mu, s'), \quad t.pre = s, \quad t.nex = s', \quad t.lab = \mu \\ s &\xrightarrow{\mu_1 \cdots \mu_n} s' =_{def} \exists s_0, \dots, s_n : \\ & \quad s = s_0 \xrightarrow{\mu_1} s_1 \xrightarrow{\mu_2} \cdots \xrightarrow{\mu_n} s_n = s' \\ s &\xrightarrow{\mu_1 \cdots \mu_n} =_{def} \exists s' : s \xrightarrow{\mu_1 \cdots \mu_n} s' \\ s &\xrightarrow{a} s' =_{def} \exists s_1, s_3, \dots, s_{2n+1} \in S_p \cup S_c, \\ & \quad s_2, s_4, \dots, s_{2n+2} \in S_n : \\ & \quad s \xrightarrow{a} s_1 \xrightarrow{a_0} s_2 \xrightarrow{\tau_0} \cdots s_{2n+1} \xrightarrow{a_n} s_{2n+2} \xrightarrow{\tau_n} s' \\ Parse(a) &=_{def} \{a_0, a_1, \dots, a_n\}. \end{aligned} \quad (2)$$

Example 5. The Needham-Schroeder Public Key (NSPK) protocol [22] is an asymmetric cryptography based authentication protocol, which defines the handshakes between two participants: the initiator i and the responder r . The brief protocol narration can be presented with the three message exchanges as follows:

$$\begin{aligned}
 \text{Msg 1. } & i \longrightarrow r \text{ (Ask) : } \{n_i, i\}_{pk_r} \\
 \text{Msg 2. } & r \longrightarrow i \text{ (Rpl) : } \{n_i, n_r\}_{pk_i} \\
 \text{Msg 3. } & i \longrightarrow r \text{ (Cfm) : } \{n_r\}_{pk_r}.
 \end{aligned} \tag{3}$$

The NSPK protocol assumes the nonces n_i and n_r are completely random and only the one who holds the private key can decrypt the cipher encrypted by the public key. In this case, when the unique n_i is sent to r through *Msg1*, although this message may be captured by some intruders, the n_i can be only obtained by r . Similar to the nonce n_r through message *Msg2*, it can be only obtained by i . Then these unique nonces are used to verify the correctness and authentication of the protocol participations.

It is hard to model and analyze the security issues of security protocols by using the general models of automata, because the important security properties are usually hidden inside the exchanging messages. However, with our SE-LTS model, the authentication processes can be modeled as internal actions and the important security properties are identified and modeled as partial labels, which are going to be used to trigger those internal actions of the protocol participations. Figure 1 presents the receiver of NSPK protocol described by SE-LTS. In this example, $S_n = \{s_0, s_2, s_4, s_6, s_8, s_{10}, s_{12}, s_{14}, s_{15}\}$, $S_p = \{s_1, s_3, s_5, s_7, s_9, s_{11}, s_{13}\}$, and $S_c = \{s_7\}$; $L_{se} = \{?Ack, !Rpl, ?Cfm\}$, where $?Ack = \{?sender.?receiver.?cipher\}$, $?Cfm = \{?sender.?receiver.?cipher\}$; $\Theta = \{\tau_0, \tau_1, \tau_2, \tau_3, \tau_4\}$, where τ_0 means r records the sender id; τ_1 means r verifies the receiver id in *?Ack* message equal to its id; τ_2 means a successful description; τ_3 means the received sender id in *?Cfm* is equal to the recorded in *?Ack*; τ_4 means the received decrypted message is equal to n_r . The mark of “ \neg ” means the conditions inverse.

4. Networked Model and Intruder

4.1. Networked Model for Multiple Components. The security protocol implements usually are installed in distributed network devices. To present the connected feature of the security protocol, we use SE-LTS model to describe each form of participation first and then combine them together as a networked transition system. As the method proposed in [23], a communication medium is needed to glue those components. The normal state $s \in S_n$ of $LTS_{se}(L)$ is defined with two levels:

- (i) higher_level state $s_i.u$ connects to the environment or other states of the same component;
- (ii) lower_level state $s_i.l$ connects to the states of other components.

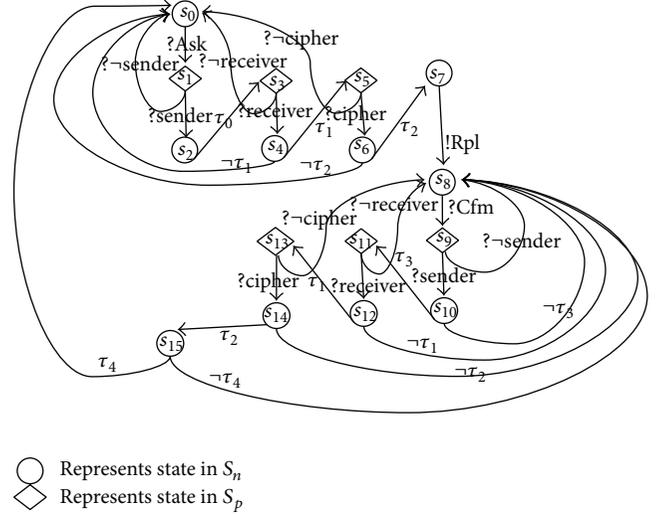


FIGURE 1: SE-LTS for NSPK receiver.

Then a common medium is considered by such transition, which begins from the lower_level state of one component and ends with the lower_level of initial state of another component. S_i, L_i denote the states and labels in $LTS_{se}(L_i)$, and S_j and L_j denote the state and labels in $LTS_{se}(L_j)$; then if $\exists !l \in L_j, \exists s_i \in S_i, !l \in Out(s_i)$, and $\exists s_j \in S_j, ?l \in L_j, ?l \in In(s_j)$, the transition of the common medium between

$IOLTS_i$ and $IOLTS_j$ is presented as $s_i.l \xrightarrow{!l} s_0.l$. We make S_{medium} to denote all the states and T_{medium} to denote all the transitions in the medium.

Definition 6 (networked SE-LTS). The implementation of a security protocol contains a set of SE-LTS systems ($LTS_{SE}(L_n)$; L_n represents the set of labels of the n th components, $n = 0, 1 \dots$), where those systems are connected sequentially by some transitions of the medium. It is also a four-tuple array: $\langle s_{0sp}, S_{sp}, L_{sp}, T_{sp} \rangle$, where

- (i) s_{0sp} is the set of initial states of $LTS_{SE}(L_n)$, $n = 0, 1 \dots$;
- (ii) S_{sp} is a set of states and $S_{sp} = S_{LTS_{SE}(L_0)} \cup S_{LTS_{SE}(L_1)} \cup \dots \cup S_{LTS_{SE}(L_n)} \cup S_{medium}$;
- (iii) L_{sp} is a finite set of labels, $L_{sp} = L_{LTS_{SE}(L_0)} \cup L_{LTS_{SE}(L_1)} \cup \dots \cup L_{LTS_{SE}(L_n)}$;
- (iv) T_{sp} is the transition relation, $T_{sp} = T_{LTS_{SE}(L_0)} \cup T_{LTS_{SE}(L_1)} \cup \dots \cup T_{LTS_{SE}(L_n)} \cup T_{medium}$.

Example 7. The networked model of NSPK protocol can be presented as in Figure 2. The transitions between two communications are the medium transition. The lower_interface and higher_interface represent the same state.

4.2. Intruder of Security Protocol Communication. With the help of networked SE-LTS model (noted as $NLTS_{se}(L)$), the specification of security protocol can be modeled as transition system. The sequential actions designed in security protocols are modeled as transition sequences of the presented

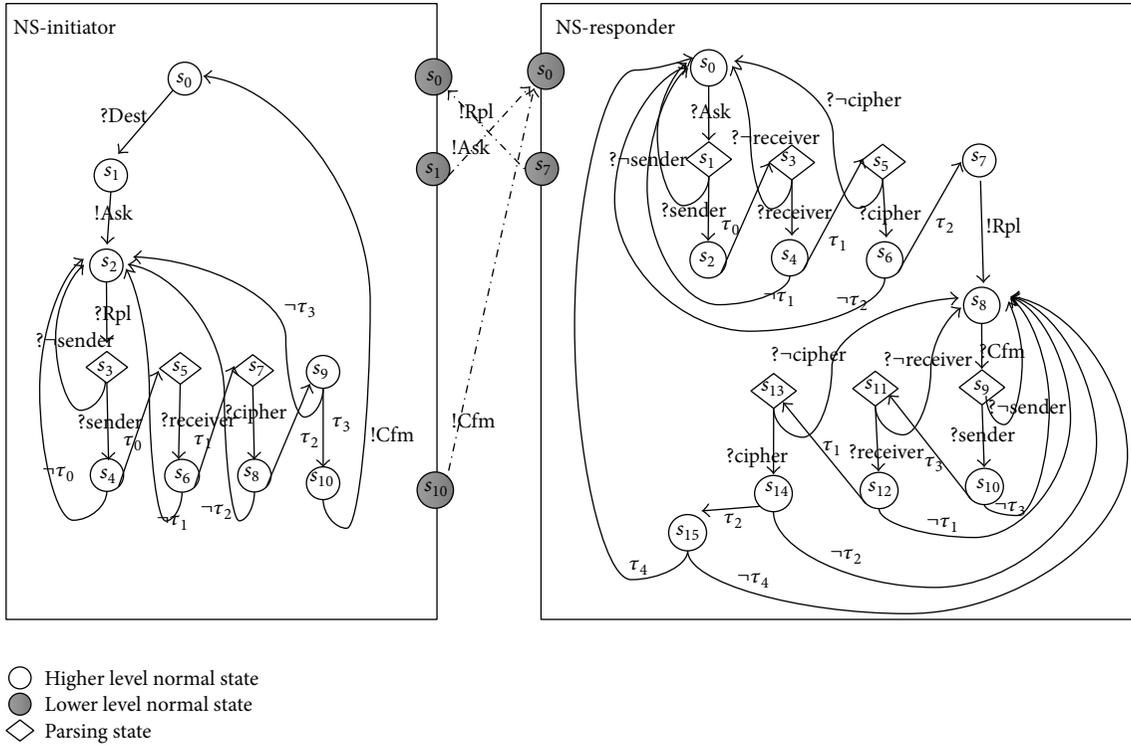


FIGURE 2: Glued security extension graph of NSPK.

reachable graph. Now, we need to consider the following question: “what kind of test cases (transition sequences) can be used to verify the security of the protocol implementations?” To address this, we give a soundness definition of the security of protocol implementations.

Definition 8 (soundness security of protocol implementation). If the interactions between the possible intruders and the protocol implementations will not result any leakage of privacy, the protocol implements are considered as secure.

The privacies are the values of some security properties, such as nonce and private key. Then the problem changes to be “how to calculate the possible transition traces which contain the interactions between intruder and protocol implementations” and “how to distinguish the leakage of privacy.” To address these questions, we need to simulate the intruder model in the proposed networked SE-LTS model.

4.2.1. Testable Intruder Model. An intruder is a powerful agent, which may participate in the protocol executions but with a purpose to attack the system. An intruder usually pretends as a legal role and has the abilities to eavesdrop, insert, or intercept the communicating messages between the legal agents. An intruder usually executes a man-in-the-middle attack. We consider the network intruder with the following assumptions.

- (1) Dolev and Yao intruder assumption: the Dolev and Yao intruder assumption was proposed since 1983 with two main properties [24]:

- (a) the cryptography is assumed to be perfect: a message can only be decrypted by someone who has the proper key (there is no way to crack the scheme);
- (b) the messages are considered to be abstract terms: either the intruder learns all messages inside the encryption (because he has the key) or he learns the encrypted message.

- (2) Tester is powerful: a tester of the system has more permissions; a tester can operate the exchanging message as the proposed intruder assumption.

And a network intruder has the following features:

- (i) no refusal: an intruder can pretend as any legal roles of the security protocol and participate in the protocol execution. It accepts all types of output messages from any legal/illegal participations and gives the corresponding response messages (including attack messages);
- (ii) knowledge learning: an intruder can parse the received messages, analyze them, and update its knowledge. This knowledge may contribute to the generation of an executable attack message;
- (iii) message matching: the output messages of the intruder are not generated randomly; they are going to be accepted by the destined principle. The outputs are generated through the intruder knowledge and will match the format of the input messages of the destination.

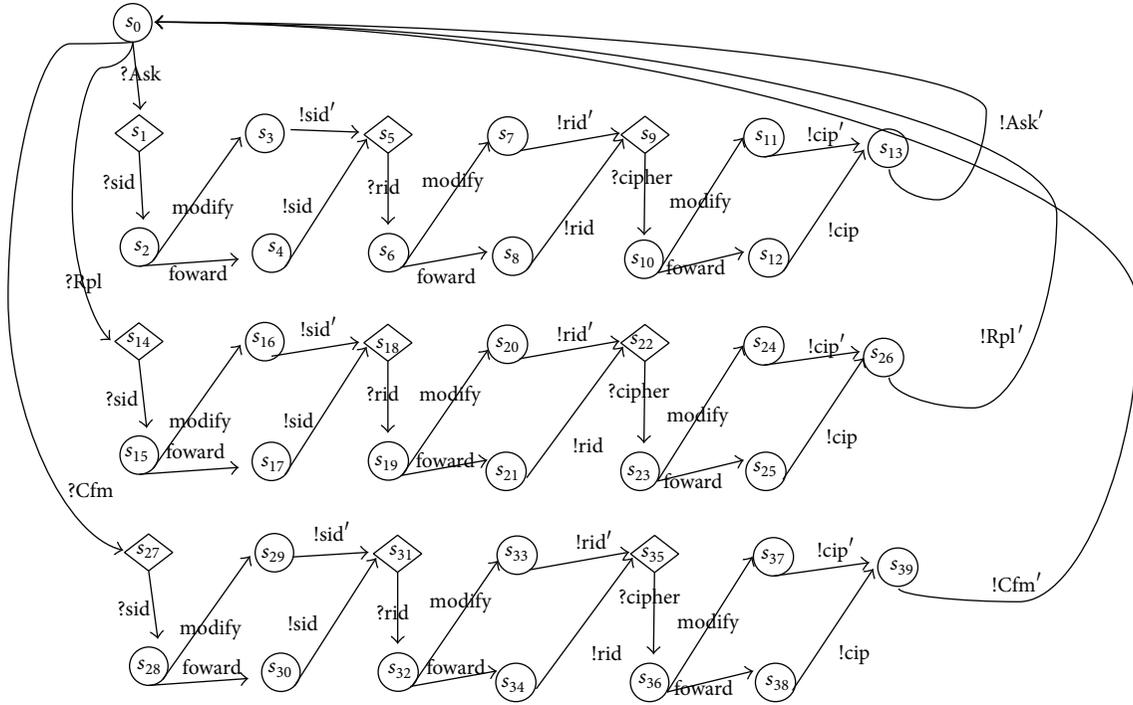


FIGURE 3: Intruder model of NSPK.

The intruder model needs to present those listed features and must be compatible with the models of the specifications. According to our assumption, the testable intruder is under control of the tester, so all of its functions (including the security functions) are reachable and observable to the tester. Here we simply use two categories of actions: *forward* and *modify*, which are triggered by the partial labels of the received messages. An intruder model of SE-LTS is defined as follows.

Definition 9 (testable intruder model). A testable intruder model of a specific protocol $\langle s_{0,sp}, S_{sp}, L_{sp}, T_{sp} \rangle$ is a SE-LTS system, which is a 4-tuple array $\langle s_0, S_{int}, L_{int}, T_{int} \rangle$, where $L_{int} \cap L_{sp} = L_{U,sp}$ and $\forall a_i \in L_{int}, \forall a'_i \in Parse(a_i), \exists s, s' \in S_{int}$,

$$s \xrightarrow{a'} \{forward, modify\} s'. \quad (4)$$

An example of testable intruder model of NSPK protocol is presented in Figure 3.

4.2.2. Combination of the Intruder Model and the Networked SE-LTS. In order to calculate the possible transitions related to the intruder actions, we consider the intruder models that exist inside the communication medium. In this case, all the related actions to the security of protocols specifications are modeled as transitions of one combined transition system and can be presented in one reachable graph. For the example of NSPK protocol, the final result of reachable graph is presented in Figure 4. The used intruder model is presented in Figure 3. We denote this combination model as $CNLTS_{se}(L_{sp})$. Generally, in a security protocol, once the current security session finishes, the security implementations

will always prepare for the next session. So the transition traces of each session of the protocol implementations usually begin and end with initial states. Meanwhile, if the received message is wrong, the implementation will go back to the waiting state and wait for the correct message. Those two kinds of situations should be both considered by the test cases, and we say a *security test case is a self-complete transition trace, which begins from the initial state and ends with the state which is waiting for the inputs from other components*. We say a transition sequence is self-complete if its input actions, which are satisfied by the output of other components, are also included in this transition trace.

4.2.3. Test Cases Generation and Verification. As the model $CNLTS_{se}(L_{sp})$ can be presented with a reachable graph, the problem of calculating the required traces can be solved by searching traces in graph and it is NP-complete. Here we propose a “deep first search algorithm” to calculate the possible traces automatically. Similar to the method we used in [25], the proposed algorithm calculates the traces from the possible end state to the initial state. The pseudocodes are presented in Algorithm 1.

Proof of Algorithm 1. The reason of the correctness of Algorithm 1 can be shown inductively as follows.

- (i) A basic transition machine M only contains one transition t . Then by following the algorithm of “Traceback” $trace_i = \{t\}$; because $t.pre$ is initial state, the Checkglue is called. Because M only contains one transition, $t.stimuli$ is empty. And “check_glue”

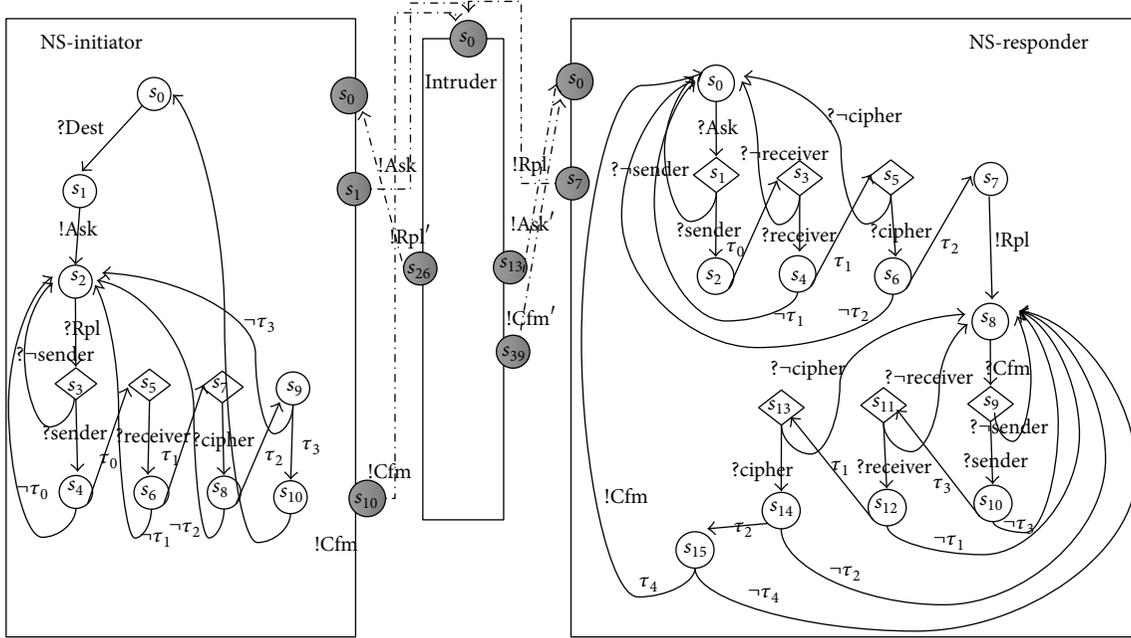


FIGURE 4: Intruder contained model, NSPK.

ends; then “Traceback” ends. The transition trace is completely found and recorded.

- (ii) In general, we can assume that the first $n - 1$ transitions of M are traced after $n - 1$ iterations of “Traceback.” Then $trace_i = \{t_0, t_1 \dots t_{n-1}\}$. $\forall t \in trace_i, t.stimuli$ is empty.
- (iii) The transition $t = t_n$ is taken as inputs of “Traceback” algorithm. By following the algorithm, t is recorded in $trace_i$, then $trace_i = \{t_0, t_1 \dots t_{n-1}, t_n\}$; $t.pre_state$ is initial state; then “check_glue” is recalled. According to the last step, the first $n - 1$ elements of $trace_i$ do not have stimuli transition. We only need to check whether t_n has stimuli transition. Because t_n is the last transition, if $t_n.stimuli$ is not empty, then another Traceback will be called, and t_n will not be the last transition. So t_n must not have stimuli transition; then $trace_hassti$ is empty. The “check_glue” ends, and “Traceback” ends. $trace_i = \{t_0, t_1 \dots t_{n-1}, t_n\}$. All the transitions of M traveled. \square

By using this proposed algorithm through $CNLTSe(L_{sp})$, the transition traces related to the security of protocol implementations (denoted as Tr_{gen}) can be calculated. Then we need to decide the *test verdict* for each transition trace to obtain the security test cases. According to the soundness security of Definition 8, a secure implementation can detect the modifications from the intruder. The test verdict of each transition trace is defined as follows.

Definition 10 (test verdict). If $\sigma \in Tr_{gen}$ is generated from protocol specification S and M is the protocol implementation, then one has the following:

- (i) if σ contains any *modify* action of intruder, the test case $te : \{\sigma \rightarrow fail \mid Out(\sigma, S) = Out(\sigma, M)\}$;
- (ii) if σ does not contain any *modify* actions, the test case $te : \{\sigma \rightarrow pass \mid Out(\sigma, S) = Out(\sigma, M)\}$;
- (iii) otherwise, the test verdict is uncertain.

Then the security test cases to verify the protocol implementations are generated. For example, by using our algorithm, 304 test cases are generated to verify the security of NSPK protocol implementations. In Table 1, we present 5 security test cases as an example. As we know, the NSPK protocol has been proved insecure with the attacks of MITM; one possible attack strategy has been proposed by Gavin Lowe in 1995 and this attack strategy corresponds to Test 4 of our example.

5. Conclusion and Future Works

In this paper, we proposed an extension model of the classic IOLTS for the purpose of security verification of protocol implementation. This extended model defines the nonnegligible security properties with partial labels to identify the security actions. Then by proposing the intruder and the combination model, the security protocol can be modeled. The proposed model can help users to generate the probable transition sequences which contain the interactions between intruder and the protocol implementations. Those generated sequences are the abstract message scenarios to verify the implementations of security protocol. We also presented a corresponding test generation algorithm to automatically generate the test cases. Some test cases of the NSPK protocol are presented also and the MITM attack of NSPK is verified to be concluded in our test cases. In the future work, we

```

Input:
Transitions set  $T_{cnlts}$  of  $CNLTS_{se}(L_s p)$ 
Output:
Transition sequence:  $trace$ 
Begin:
ArrayList < transition >  $trace_{rec}$ ;
ArrayList < transition >  $T_{search}$ ;
Public static void main(String[] args){
    For each  $t \in T_{cnlts}$ {
        If ( $t.nex \in s_{0cnlts}$ ){
             $trace = \text{Traceback}(t, T_{cnlts})$ ;
             $\text{Output}(trace)$ ;
        }
    }
    For each  $t_1 \in T_{cnlts}$ {
        For each  $t_2 \in T_{cnlts}$ {
            If ( $t_1.nex = t_2.nex$ ){
                 $trace = \text{Traceback}(t_1, T_{cnlts})$ ;
                 $\text{Output}(trace)$ ;
            }
        }
    }
}
Public transition[] Traceback(transition  $t$ , transition[]  $M$ ){
    If ( $t.pre \notin s_{0M}$ ){
         $trace_{ec}.add(t)$ ;
        For each  $t_{pre} \in \text{Search\_tran}(t.pre, M)$ {
             $\text{Traceback}(t_{pre}, M)$ ;
        }
    }
    Checkglue( $t.pre, trace_{rec}, M$ );
    return  $trace_{ec}$ ;
}
Public transition[] Checkglue(state  $s$ , ArrayList < transition >  $trace_i$ , transition[]  $M$ ){
    For each  $t \in trace_i$ {
        If  $t.lab \in L_{IM}$  and  $t.stimuli \neq$  and  $t.stimuli \notin trace_i$ {
            For each  $t_s \in \text{Search\_tran}(s, M)$ {
                If ( $t_s.lab = t.stimuli.lab$ ){
                     $\text{Traceback}(t_s, M)$ ;
                     $t.stimuli =$ ;
                }
            }
        }
    }
}
Public ArrayList < transition > Search_tran(state  $s$ , transition[]  $M$ ){
    For each  $t \in T_M$ {
        If ( $t.nex = s$ ){
             $T_{search}.add(t)$ ;
        }
    }
    return  $T_{search}$ ;
}
End;

```

ALGORITHM 1: Test generation algorithm.

TABLE 1: Some security test cases for NSPK protocol.

Test 1	?Dest, !Ask, ?Ask, ?sid, modify, !sid, ?rid, forward, !rid, ?cipher, forward, !cip, !Ask, ?Ask, ?sender, $\tau_{0,r}$, ?receiver, $\tau_{1,r}$, ?cipher, $\tau_{2,r}$, !Rpl, ?Rpl, ?sid, forward, !sid, ?rid, forward, !rid, ?cipher, forward, !cip, !Rpl, ?Rpl, \neg sender, fail;
Test 2	?Dest, !Ask, ?Ask, ?sid, forward, !sid, ?rid, forward, !rid, ?cipher, forward, !cip, !Ask, ?Ask, ?sender, $\tau_{0,r}$, ?receiver, $\tau_{1,r}$, ?cipher, $\tau_{2,r}$, !Rpl, ?Rpl, ?sid, forward, !sid, ?rid, forward, !rid, ?cipher, forward, !cip, !Rpl, ?Rpl, \neg sender, pass;
Test 3	?Dest, !Ask, ?Ask, ?sid, forward, !sid, ?rid, forward, !rid, ?cipher, forward, !cip, !Ask, ?Ask, ?sender, $\tau_{0,r}$, ?receiver, $\tau_{1,r}$, ?cipher, $\tau_{2,r}$, !Rpl, ?Rpl, ?sid, forward, !sid, ?rid, forward, !rid, ?cipher, forward, !cip, !Rpl, ?Rpl, ?sender, $\tau_{0,i}$, ?receiver, $\tau_{1,i}$, ?cipher, $\tau_{2,i}$, $\tau_{3,i}$, !Cfm, ?sid, forward, !sid, ?rid, forward, !rid, ?cipher, forward, !cip, !Cfm, ?Cfm, ?sender, $\tau_{3,r}$, ?receive, $\tau_{1,r}$, ?cipher, $\tau_{2,r}$, $\tau_{4,r}$, pass;
Test 4	?Dest, !Ask, ?Ask, ?sid, modify, !sid, ?rid, forward, !rid, ?cipher, forward, !cip, !Ask, ?Ask, ?sender, $\tau_{0,r}$, ?receiver, $\tau_{1,r}$, ?cipher, $\tau_{2,r}$, !Rpl, ?Rpl, ?sid, modify, !sid, ?rid, modify, !rid, ?cipher, modify, !cip, !Rpl, ?Rpl, ?sender, $\tau_{0,i}$, ?receiver, $\tau_{1,i}$, ?cipher, $\tau_{2,i}$, $\tau_{3,i}$, !Cfm, ?sid, modify, !sid, ?rid, modify, !rid, ?cipher, modify, !cip, !Cfm, ?Cfm, ?sender, $\tau_{3,r}$, ?receive, $\tau_{1,r}$, ?cipher, $\tau_{2,r}$, $\tau_{4,r}$, fail;
Test 5	?Dest, !Ask, ?Ask, ?sid, forward, !sid, ?rid, forward, !rid, ?cipher, forward, !cip, !Ask, ?Ask, ?sender, $\tau_{0,r}$, ?receiver, $\tau_{1,r}$, ?cipher, $\tau_{2,r}$, !Rpl, ?Rpl, ?sid, forward, !sid, ?rid, forward, !rid, ?cipher, forward, !cip, !Rpl, ?Rpl, ?sender, $\tau_{0,i}$, ?receiver, $\tau_{1,i}$, ?cipher, $\tau_{2,i}$, $\tau_{3,i}$, !Cfm, ?sid, modify, !sid, ?rid, forward, !rid, ?cipher, forward, !cip, !Cfm, ?Cfm, ?sender, $\tau_{3,r}$, ?receive, $\tau_{1,r}$, ?cipher, $\tau_{2,r}$, $\tau_{4,r}$, fail;

plan to add the fuzzy sets during the test generation process, which can better simulate the actions of intruder of a network system. Some real network protocols, such as RADIUS and the protocol used in [26], are going to be considered. And the more complex experiments [27] are going to be defined.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] Wikipedia, "Global surveillance disclosures (2013-present)," 2014, [http://en.wikipedia.org/wiki/Global_surveillance_disclosures_\(2013-present\)](http://en.wikipedia.org/wiki/Global_surveillance_disclosures_(2013-present)).
- [2] Y. L. Fu and O. Koné, "Validation of security protocol implementations from security objectives," *Computers & Security*, vol. 36, pp. 27–39, 2013.
- [3] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines—a survey," *Proceedings of the IEEE*, vol. 84, no. 8, pp. 1090–1123, 1996.
- [4] A. Armando, G. Pellegrino, R. Carbone, A. Merlo, and D. Balzarotti, "From model-checking to automated testing of security protocols: bridging the gap," in *Tests and Proofs*, vol. 7305 of *Lecture Notes in Computer Science*, pp. 3–18, 2012.
- [5] F. Dadeau, P. C. Héam, and R. Kheddad, "Mutation-based test generation from security protocols in HLPSTL," in *Proceedings of the 4th IEEE International Conference on Software Testing, Verification, and Validation (ICST '11)*, pp. 240–248, Berlin, Germany, March 2011.
- [6] Y. Hsu, G. Q. Shu, and D. Lee, "Model-based approach to security flaw detection of network protocol implementations," in *Proceedings of the 16th IEEE International Conference on Network Protocols (ICNP '08)*, pp. 114–123, Orlando, Fla, USA, October 2008.
- [7] A. Datta, A. Derek, J. C. Mitchell, and A. Roy, "Protocol composition logic (PCL)," *Electronic Notes in Theoretical Computer Science*, vol. 172, pp. 311–358, 2007.
- [8] S. Andova, C. Cremers, K. Gjosteen, S. Mauw, S. F. Mjolsnes, and S. Radomirovic, "A framework for compositional verification of security protocols," *Information and Computation*, vol. 206, no. 2–4, pp. 425–459, 2008.
- [9] AVISPA-Project, "The high level protocol specification language," 2003, <http://www.avispa-project.org/delivs/2.1/d2-1.pdf>.
- [10] S. Salamah, A. Gates, and V. Kreinovich, "Validated templates for specification of complex LTL formulas," *Journal of Systems and Software*, vol. 85, no. 8, pp. 1915–1929, 2012.
- [11] K. W. Kim and J. D. Lee, "On the security of two remote user authentication schemes for telecare medical information systems," *Journal of Medical Systems*, vol. 35, article 17, 2012.
- [12] E. M. Clarke, O. Grumberg Jr., and D. A. Peled, *Model Checking*, The MIT Press, 2000.
- [13] W. Wang and X. L. Xin, "On fuzzy filters of pseudo BL-algebras," *Fuzzy Sets and Systems*, vol. 162, no. 1, pp. 27–38, 2011.
- [14] G. Gonenc, "A method for the design of fault detection experiments," *IEEE Transactions on Computers C*, vol. 19, no. 6, pp. 551–558, 1970.
- [15] W. H. Chen, "An optimization technique for protocol conformance testing based on the wp method," *International Journal of Applied Science and Engineering*, vol. 1, no. 1, pp. 45–54, 2003.
- [16] Y. L. Fu and O. Koné, "Security and robustness by protocol testing," *IEEE System Journal*, no. 99, pp. 1–8, 2012.
- [17] T. Avanesov, Y. Chevalier, M. A. Mekki, and M. Rusinowitch, "Web services verification and prudent implementation," in *Data Privacy Management and Autonomous Spontaneous Security*, vol. 7122 of *Lecture Notes in Computer Science*, pp. 173–189, Springer, Berlin, Germany, 2012.
- [18] A. Gouglidis, I. Mavridis, and V. C. Hu, "Security policy verification for multi-domains in cloud systems," *International Journal of Information Security*, vol. 13, no. 2, pp. 97–111, 2014.
- [19] Y. L. Fu and O. Koné, "Using transition systems to model and verify the implementation of security protocol," in *Proceedings of the 6th International Conference on Security of Information and Networks*, pp. 210–217, Aksaray, Turkey, November 2013.
- [20] J. Tretmans, "Conformance testing with labelled transition systems: implementation relations and test generation," *Computer Networks and ISDN Systems*, vol. 29, no. 1, pp. 49–79, 1996.
- [21] N. A. Lynch and M. R. Tuttle, "An introduction to input/output automata," *CWI Quarterly*, vol. 2, pp. 219–246, 1989.

- [22] G. Lowe, "Breaking and fixing the needham-schroeder public-key protocol using fdr," in *Tools and Algorithms for the Construction and Analysis of Systems*, vol. 1055 of *Lecture Notes in Computer Science*, pp. 147–166, Springer, Berlin, Germany, 1996.
- [23] Y. L. Fu and O. Koné, "Network security against threatening requests," in *Data Privacy Management and Autonomous Spontaneous Security*, vol. 7122 of *Lecture Notes in Computer Science*, pp. 280–294, Springer, Berlin, Germany, 2012.
- [24] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Information Theory Society*, vol. 29, no. 2, pp. 198–208, 1983.
- [25] Y. L. Fu and O. Koné, "A robustness testing method for network security," in *Global Security, Safety and Sustainability & e-Democracy*, vol. 99 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 38–45, Springer, Berlin, Germany, 2012.
- [26] W. Z. Zhang, Y. Zhang, and T. H. Kim, "Detecting bad information in mobile wireless networks based on the wireless application protocol," *Computing*, pp. 1–20, 2013.
- [27] W. Z. Zhang, X. H. Wang, B. Lu, and T. H. Kim, "Secure encapsulation and publication of biological services in the cloud computing environment," *BioMed Research International*, vol. 2013, Article ID 170580, 8 pages, 2013.

Research Article

A Game-Theoretic Response Strategy for Coordinator Attack in Wireless Sensor Networks

Jianhua Liu,¹ Guangxue Yue,¹ Shigen Shen,¹ Huiliang Shang,² and Hongjie Li¹

¹ College of Mathematics, Physics and Information Engineering, Jiaying University, Jiaying 314001, China

² Department of Electronic Engineering, Fudan University, Shanghai 200433, China

Correspondence should be addressed to Jianhua Liu; ljh_541@163.com

Received 11 March 2014; Accepted 11 June 2014; Published 1 July 2014

Academic Editor: Fei Yu

Copyright © 2014 Jianhua Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The coordinator is a specific node that controls the whole network and has a significant impact on the performance in cooperative multihop ZigBee wireless sensor networks (ZWSNs). However, the malicious node attacks coordinator nodes in an effort to waste the resources and disrupt the operation of the network. Attacking leads to a failure of one round of communication between the source nodes and destination nodes. Coordinator selection is a technique that can considerably defend against attack and reduce the data delivery delay, and increase network performance of cooperative communications. In this paper, we propose an adaptive coordinator selection algorithm using game and fuzzy logic aiming at both minimizing the average number of hops and maximizing network lifetime. The proposed game model consists of two interrelated formulations: a stochastic game for dynamic defense and a best response policy using evolutionary game formulation for coordinator selection. The stable equilibrium best policy to response defense is obtained from this game model. It is shown that the proposed scheme can improve reliability and save energy during the network lifetime with respect to security.

1. Introduction

Both the security and the quality of service (QoS) of ZWSNs are important factors which will affect various services of sensor data delivery, for example, environmental monitoring [1], building monitoring to assess earthquake damage [2], and intelligent home monitoring [3]. On one hand, QoS is one of the key factors for various services that include diverse important parameters, that is, delay, throughput, and dropping probability of packets. Different types of services always need different requirements of QoS. On the other hand, sensor transmission always faces many malicious attacks [4]. In order to get secure network, various security protocols are presented to address security issues [5], including wired equivalent privacy (WEP), 802.1X port access control with extensible authentication protocol (EAP), and IP security protocol (IPsec) [6].

An IEEE 802.15.4 WSN is composed of one coordinator and a set of nodes [7]. The network topology defined in the standard is called cluster tree, where nodes associate with

coordinators to establish parent child relationships and form a tree rooted at coordinator. Existing ZWSNs can cooperate with each other to compose sensor service and provide effective and efficient data delivery.

It is evident that sensor services have a certain requirement on both security and QoS to get good performance. Since the requirements might be different in a different circumstance or time, it may be impossible to satisfy requirements from both security and QoS simultaneously due to limited network resources. Considering only the costs of transmission without taking into account the possibility of coordinator attacked is not sufficient for providing secure composition in these environments. To avoid single point of failures owing to attack and increase network lifetime, it is desirable that the service composition based on coordinator selection method executes in a distributed manner.

Wireless sensor data can be proactively received by a coordinator owned by the same network operator. However, when a coordinator node is no longer in the reliable state, the rate of data delivery and QoS will be poor. Therefore, to

improve QoS, multiple sensor nodes can share the reliable coordinator in which case the cost of sensor data transmission for each sensor node will be reduced. In other words, multiple sensor nodes can form a coalition to share the coordinator. When the coordinator nodes in the same coalition are reliable owing to unattack, each of the sensors can fully access the coordinator. However, if one coordinator node from the same coalition is unreliable owing to heavy attack, coordinators failures can occur; therefore, a coordinator selection mechanism would be required for sensor data delivery in the same coalition. In this context, two key questions are (i) how to form coalitions among sensor nodes to share the reliable coordinator to minimize the cost of energy and (ii) how to defend attacker to meet the required QoS requirements. To answer these questions, a joint dynamic defense and coordinator selection scheme are proposed using game theoretical concepts.

Given that the sensors are rational to minimize their own cost, a game-theoretic model is developed to find a solution of the defense attack and coordinator selection problem. This game model consists of two interrelated formulations, that is, a stochastic game for dynamic attack response and evolutionary game for coordinator selecting. The stochastic game formulation utilizes the coalitional structure obtained from the evolutionary game, while the evolutionary game formulation utilizes the cost and QoS performance measures from the stochastic game.

In this paper, our main contributions can be summarized as follows.

- (i) We propose a proactive scheme for defending networking coordinators. It enables the defender to proactively select reliable coordinator to minimize the expected network energy loss. To our best knowledge, this is the first work that considers defending and attacking from the perspective of games.
- (ii) We formulate the problem of the defending networking coordinators as a 2-player zero-sum game. The payoff of our problem is measured by the maximum sensor service network utility. We propose an evolutionary game-theoretic framework for the defense response policy in which nodes in the network are regarded as players and the local combination of estimation information from different neighbors is regarded as different strategies of coordinator selection.
- (iii) We propose a new state estimation algorithm for selecting coordinators using fuzzy logic. We prove that a global Nash equilibrium (NE) exists. We then design a mixed-strategy solution for the defender and attacker that combines the evolutionary game NE strategies and stochastic game NE strategies in order to achieve the maximum payoffs for both players.

The rest of this paper is organized as follows. Section 2 describes related works. Section 3 describes the system model. Section 4 presents the stochastic game for dynamic defense and a best response policy using evolutionary game formulation for coordinator selection. Section 5 performs

numerical experiments; the influence of a cost parameter is illustrated. Section 6 concludes the paper.

2. Related Works

ZWSNs security and the quality of service (QoS) combining deployment and management related topics have become an active research area. One of the major constraints of ZWSNs deployment and management is the limited energy. It is crucial for maximizing the lifetime of ZWSNs that data packets are routed to the destination in an energy-efficient manner [8]. ZWSNs are widely studied route metrics for the number of hops [9–11]; hierarchical protocols [12, 13] group nodes into clusters and energy expenditure [14–17]. The relationship between number of hops and network energy for a single packet is investigated in [18]. Most of these approaches do not consider the reliability on the coordinator nodes that suffer attack from malicious nodes. In this paper, we study the possibility of using game theoretical approach to defend malicious nodes. To combat the attack on relay, several lightweight authentication protocols, which are based on computationally efficient hash chain, can be applied in cooperative wireless communication networks. Timed efficient stream loss tolerant authentication (TESLA) is a broadcast authentication protocol based on loose time synchronization [19]. Law et al. [20] showed how the jamming can be used to perform attacks on the network link layer protocols. Xu et al. [21] surveyed issues related to performing a jamming attack against sensor networks by examining both the attack and defense; they presented the following jamming models: constant jammer, deceptive jammer, random jammer, and reactive jammer.

Yao et al. [22] proposed a parameterized and localized trust management scheme for WSN security, particularly for secure routing, where each node only maintains highly abstracted parameters to evaluate its neighbors. Aivaloglou and Gritzalis [23] proposed a hybrid trust and reputation management protocol for WSNs by combining certificate-based and behavior-based trust evaluations. Gabrielli et al. [24] analyzed the security vulnerabilities of PEAS, ASCENT, and CCP and represented securing topology maintenance protocols. Bao et al. [25] considered multidimensional trust attributes derived from communication and social networks to evaluate the overall trust of a sensor node. Zonouz et al. [26] employed a game-theoretic response strategy against adversaries modeled as opponents in a two-player Stackelberg stochastic game and proposed fuzzy logic theory to calculate the network-level security metric values. However, [24–26] do not provide response strategy against attackers for coordinators.

Game theory provides a rich set of tools that can be used to model the attack behavior of malicious nodes. Game theory models were applied to solve various issues in wireless networks. In [27], a stochastic game was formulated for network selection problem in cognitive heterogeneous networks. In [28], an evolutionary game model was used to analyze the information diffusion process and the filtering over the adaptive networks. In this paper, we use stochastic and

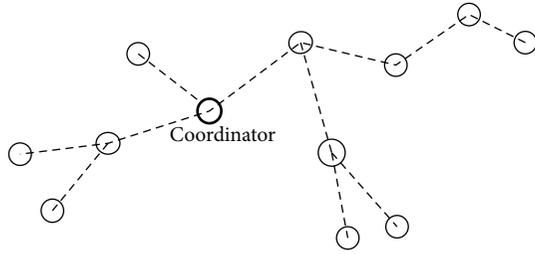


FIGURE 1: Selecting coordinator coalitional game model.

evolutionary game theory to study the cooperative defense behavior of sensors for coordinator selection under security constraints.

3. System Model

3.1. ZWSN Functionality and QoS. Each node typically provides a basic functionality for operating on the monitored data; sensors provide their functionalities through receiving and forwarding operations. That is, a data packet comes to a sensor, and the sensor receives and forwards to coordinator, while the network of sensor nodes collectively provides a composite service to the end nodes. Unlike the web environment where service provider availability and ample communication bandwidth are typically assured, sensor networks are highly dynamic as nodes often fail or become disconnected and wireless communication capacity is limited. The operation of every ZWSN is associated with two QoS attributes: the average number of hops to the coordinator (hop for minimization) and energy cost (energy for minimization). Energy cost is defined as the fee that a ZWSN consumer has to pay to receiving and forwarding operations. Hop is defined as the total length of path covered between the source of the sensor and the coordinator. Figure 1 illustrates the forwarding and selecting coordinator coalitional game model. Cluster coalition can communicate with other coalitions by coordinators. Not only can a coordinator request or be invited to join a coalition, but also it can leave a coalition. Every coalition has a coordinator and a set of sensors as its members. The coordinator is responsible for receiving and forwarding data and managing the cooperation among these members.

In a coalition, coordinator is the controller of the network and it is responsible for initiating the network set-up; it starts by selecting a suitable communication channel. This selection is performed by the energy detection scan which assesses the level of interference on each channel by measuring the peak energy on each available channel. If a node is available, it selects a node to join coalition; this node starts an exchange of signaling packets with the chosen coordinator to complete formation coalition. A cooperation coalition is managed by the coordinator, which is a sensor composition $S = [s[1], \dots, s[n]]$ of the coalition members $\{1, \dots, n\}$; once the coordinator receives a sensor's request, it forwards $s[1]$. If $s[1]$ is available, the coordinator will send data to $s[1]$. Otherwise, it will forward $s[2]$. This process continues until a sensor is

available and data is delivered to destination node. Suppose sense i 's availability is a_i . Clearly, the resource availability of the coalitions M is given by

$$a_r = 1 - \prod_{i=1}^n (1 - a_i). \quad (1)$$

As a coalition can be regarded as a composite sensor, it also possesses other performance metrics as discussed earlier. In particular, its price p_M relies on its availability. Besides, following the widely accepted assumption that the cost of a composite sensor totally depends on the cost of the average number of hops to the coordinator and energy cost, the cost of a coalition M is given by

$$c_M(s) = \sum_{i=1}^n \prod_{j=1}^{i-1} (1 - a_{s[j]}) c_{s[i]}, \quad (2)$$

where $c_{s[i]}$ is the cost of coalition $s[j]$. Let h_i be the average number of hops from the node k to the coordinator. The energy cost for a single packet can be calculated as follows: $c_{s[i]} = (E_r + E_f) \times h_i \times N_i$, where E_r and E_f are energy of the receiving and forwarding, respectively. N_i is the number of nodes in coalition i . The value of a coalition M is a sensor composition S given by $v_M(s) = p_M \times a_M - c_M(s)$.

3.2. Stochastic Game for Coordinator Attack. In this section, we present the game-theoretic formulation of the self-organized network selection problem. We model the coordinator attacked problem as a noncooperative game where the malicious sensors are the attackers, the coordinator sensors are the defenders, and the coordinator run state is considered as the internal state (*NormalState* (NS) or *HackedState* (HS)). The game is represented as

$$G = (N, Z, \{A_k\}_{k \in N}, \{u_k\}_{k \in N}), \quad (3)$$

where N is the set of players, $N = \{1, 2\} = \{\text{attacker, defender}\}$.

Z is the space of states, $Z_i = \{\text{NS, HS}\}$, and $\{A_i\}_{i \in N}$ is the set of actions (attack and defense) that player i can take. $A_i = \{a_i, r_i, d_i, \theta_i\}$; a_i is the attack action bringing the coordinator from state *NormalState* to *HackedState*. r_i is the resignation of the attack in state *NormalState*. d_i represents that the attack action a_i will be detected by the defender. θ_i represents that the attack a_i action will be undetected.

$\{u_i\}_{i \in N}$ is the utility function of player i . The defenders aim at maximizing the network lifetime with carefully-designed coordinator selecting schedules, while the malicious attackers want to decrease the network lifetime by strategic jamming. Therefore, they have opposite objectives and their dynamic interactions can be well modeled as a noncooperative (zero-sum) game. The coordinators as defenders are cooperative and rational players with the objective of maximizing their network throughput and decreasing the average number of hops and energy cost to itself. Thus, we define the utility function as defender's expected payoff for a choice of action as

$$u_i(a_i, a_{-i}) = E[v_i(s) | (a_i, a_{-i})] = \sum_{a \in A} p(a) v_k(a). \quad (4)$$

The choice of defender i that maximize defender i 's expected payoff over its action space A_k is called the player's best response action. The decision making of defender i in a game then becomes

$$(G): \max_{a_i \in A_k} u_i(a_i, a_{-i}), \quad \forall i \in N. \quad (5)$$

3.3. Best Response Policy Using Evolutionary Game Formulation. Given a network topology denoted as a directed graph $G = (N, E)$, N is the set of nodes and E is the set of arcs. Each node $i \in N$ has the initial resource availability a_i . Let r_i be the self-resistance of node i to attacks. Let x_i and y_i be the defending and attacking resource allocated to defenders and attackers, respectively. We adopt the contest model proposed in [29] where the resource availability loss ratio of node i is given by

$$\tau_{ij}(x_i) = \frac{(y_i)^m}{\alpha_i(r_i + x_i)^m + (y_i)^m}, \quad (6)$$

where $m \in (0, 1]$ reflects the nonlinearity or linearity of the loss ratio on node i , and α_i is a parameter reflecting the relative difficulties for the defender to protect in a particular node compared with the attacker. When $\alpha_i \in (0, 1)$, the defender has to allocate more resources than the attacker in order to mitigate the effect of the attack, while $\alpha_i > 1$ means that the defender can easily detect and mitigate the effect of the attack. The payoff of a coalition M is a sensor composition S rewritten as

$$v_M^a(s) = p_M \times a_M - c_M(s) - \tau_M(s). \quad (7)$$

Let A_0 be the original sensor composition S resource availability without suffering from any attack. For the defender, its goal is to maximize the payoff $v_M^a(s)$ over x to protect the maximum network QoS by selecting coordinator as much as possible. On the other hand, the attacker aims to minimize $v_M^a(s)$ by attacking key coordinator nodes or, equivalently, maximize $A_0 - v_M^a(s)$ over y . Equation (7) suggests that $v_M^a(s)$ can be increased if $c_M^a(s)$ can be reduced. It is to be noticed that $c_M(s)$ depends on h_i, E_r , and E_f ; a reduction of h_i, E_r , and E_f also reduces $c_M(s)$. This can be obtained by evolutionarily selecting the coordinator position and deciding its reliability state. For coordinator node i with degree d_i (not including node itself) and coalition set $\{i_1, \dots, i_{d_i}\}$, the general parameter updating rule can be written as

$$\begin{aligned} h_{i,t+1} &= B_{i,t+1}(\Phi(h_{i_1,t}), \Phi(h_{i_2,t}), \dots, \Phi(h_{i_{d_i},t})) \\ &= \sum_{l \in w} \sum_{j \in N} B_{i,t+1}(j, l) \Phi(h_{j,l,t}), \end{aligned} \quad (8)$$

where $\Phi(\cdot)$ can be any adaptive role configuration function. $B_{i,t+1}$ represents some specific linear combination rules, w represents the number of coalitions, and N denotes the number of coalitions members.

Evolutionary game theory (EGT) is originated from the study of ecological biology [30], which differs from the classical game theory by emphasizing more on the dynamics

and stability of the whole population's strategies, instead of only the property of the equilibrium. Such an equilibrium strategy is defined as the evolutionarily stable strategy (ESS).

Let us consider an evolutionary game with k strategies $\theta = \{1, 2, \dots, k\}$. The utility matrix U is a matrix $k \times k$, whose entries γ_{ij} denote the payoff for strategy i versus strategy j . The population fraction of strategy i is given by p_i , where $0 < p_i < 1, i \in \{1, \dots, k\}$. The fitness of strategy is given by $f_i = \sum_{j=1}^k p_j \gamma_{ij}$. For the average fitness of the whole population, we have $\eta = \sum_{i=1}^k p_i f_i$. The Wright-Fisher model has been widely adopted to let a group of players converge to the ESS [27], where the strategy updating equation for each player can be written as

$$p_i(t+1) = \frac{p_i(t) f_i(t)}{\eta(t)}. \quad (9)$$

From (8), it can be seen that the strategy updating process in the evolutionary game is similar to the position parameter updating process in adaptive selecting of the coordinator problem. It is intuitive that we can use evolutionary game to formulate the distributed adaptive selecting of the coordinator position problem. Given the definition of the players, strategy space, and payoffs, the maximum network QoS of evolutionary game Ω can be defined as

- (i) players: defender, attacker,
- (ii) strategy: θ_{ij} ,
- (iii) payoffs: $v_M^a(s), A_0 - v_M^a(s)$.

Each coordinator node represents a defense player; θ_{ij} denotes the probability that the strategy of node i will replace that of node j from its neighbor. We first discuss how players' strategies are updated in EGT, which is then applied to the position parameter updating in distributed adaptive coordinator selection. In EGT, the fitness of a player is locally determined from interactions with all adjacent players, which is defined as

$$f = (1 - \lambda) \cdot v_m(s) + \lambda \cdot v_M^a(s), \quad (10)$$

where λ parameter represents the selection of new coordinator node intensity, that is, the relative contribution of the game to fitness. The case $\lambda \rightarrow 0$ represents the limit of weak selection of new coordinator node owing to jam weak attack, while $\lambda = 1$ denotes strong selection, where fitness equals payoff. There are two different strategy updating rules for the evolution dynamics called AC, TC.

- (i) AC (alternative coordinator) update rule: a coordinator player is chosen to abandon his/her current coordinator role. Then, the chosen player selects one of its neighbors as coordinator with the probability of being proportional to their fitness; its neighbor copies its strategy and configuration, as shown in Figure 2(a).
- (ii) TC (temporary coordinator) update rule: a neighbor player adopts the strategy and configuration of one coordinator as a temporary coordinator node and

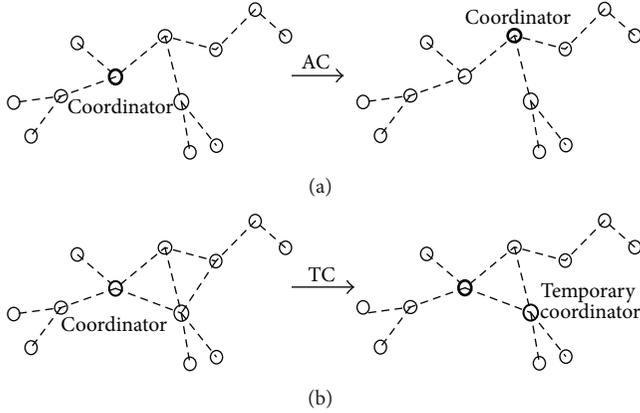


FIGURE 2: (a) Alternative coordinator. (b) Temporary coordinator.

remains with its current strategy; old coordinator configures as a temporary ordinary (TO) node, with the probability of being proportional to fitness. Moreover, the neighbor player keeps the strategy and configuration until the jamming attack weakens and old coordinator node recovers its role, as shown in Figure 2(b).

These two kinds of strategy updating rules can be matched to two different kinds of position parameter updating algorithms in distributed adaptive coordinator selection. The degree of coordinator node i is d_i . We use N to denote the set of all nodes in a coalition.

For the AC update rule, the probability that the coordinator player selects and configures one of its neighbors j as coordinator is

$$P_j = \frac{f_j}{\sum_{q \in N} f_q} \frac{1}{\Gamma_j}, \quad (11)$$

where the $f_j / \sum_{q \in N} f_q$ is the probability that the neighboring node j is chosen to act as coordinator, and $1/\Gamma_j$ is the probability that node j is chosen for copying coordinator's strategy and updating configuration. The equivalent parameter updating rule for ZWSN can be written as

$$h_{i,t+1} = \left(\frac{f_j}{\sum_{q \in N} f_q} \frac{1}{\Gamma_j} \right) \Phi_{AC}(h_{j,t}) + \left(1 - \frac{f_j}{\sum_{q \in N} f_q} \frac{1}{\Gamma_j} \right) \sum_{i \in \omega \cup N \setminus \{j\}} \Phi(h_{i,t}), \quad (12)$$

where the first term is that the neighboring node is chosen for configuration as an alternative coordinator, and the second term is that all nodes are configured as a new average hop from source node to new coordinator.

For the TC update rule, the equivalent parameter updating rule for ZWSN can be written as

$$h_{z,t+1} = \left(\frac{f_j}{\sum_{q \in N} f_q} \frac{1}{\Gamma_j} \right) \Phi_{TC}(h_{j,t}) + \left(\frac{f_j}{\sum_{q \in N} f_q} \frac{1}{\Gamma_j} \right) \Phi_{TO}(h_{i,t}) + \left(1 - \frac{f_j}{\sum_{q \in N} f_q} \frac{1}{\Gamma_j} \right) \sum_{z \in \omega \cup N \setminus \{j,i\}} \Phi(h_{z,t}), \quad (13)$$

where the first term is that the neighboring node is chosen for configuration as a temporary coordinator, the second term is that it itself is configured as a temporary ordinary node, and the third term is that all nodes are configured as a new average hop from source node to new temporary coordinator. The payoff of a coalition M is a sensor composition S rewritten by

$$v_M^a(s) = p_M \times a_M - c_M(s) - \tau_M(s) - c_\Phi(s), \quad (14)$$

where $c_\Phi(s)$ is the cost of configuration. Its immediate reward for a cluster sensor with n coordinators is defined as a weighted sum of the performance of a cluster sensor:

$$r_\Phi = \sum_{i=1}^n (w_i \cdot c_\Phi^{AC}(i) + (1 - w_i) \cdot (c_\Phi^{TC}(i) + c_\Phi^{TO}(i))), \quad (15)$$

where $w_i < 1$. The "goodness" of a configuration action in a given evolutionary state is measured by a value function $Q(Z, a)$; we employ temporal-difference (TD) method for configuration function update:

$$Q(Z_t, a_t) \leftarrow Q(Z_t, a_t) + \omega \cdot [r_{\Phi,t+1} + \xi \cdot Q(Z_{t+1}, a_{t+1}) - Q(Z_t, a_t)], \quad (16)$$

where ω is a learning rate parameter that facilitates convergence in the presence of stochastic transitions.

4. FQL Based Reinforcement Learning for Coordinator Selection

4.1. Fuzzy Logic. Fuzzy logic is a mathematical approach to emulate human way of thinking and learning. Fuzzy systems have been used as function approximating to facilitate generalization in state space for generating continuous actions. We propose fuzzy Q learning (FQL) [10] to a fuzzy evolutionary game decision (FEGD) setting. The proposed FEGD takes into account the channel occupied information with respect to the coordinator (C_o) and the amount of remaining battery energy of that coordinator (E_b). Their degree of relevance is expressed as a function $\chi = f(C_o, E_b)$, where χ denotes "selection level" or "quality" of the coordinator. In FEGD system, the input linguistic parameters are the amount of channel occupied with coordinator (C_o) and the amount of remaining battery energy of that coordinator (E_b). The

term sets for each input linguistic parameter are defined, respectively, as

$$T(C_0) = \{\text{Low (LO)}, \text{High (HG)}\}, \quad (17)$$

$$T(E_0) = \{\text{Low (LO)}, \text{Moderate (ME)}, \text{High (HG)}\}.$$

The output linguistic parameter that is the possibility of coordinator selection is defined as

$$O(\chi) = \{\text{Low (LO)}, \text{Moderate (ME)}, \text{High (HG)}\}. \quad (18)$$

The fuzzy rules matrix is also summarized in Table 1. Following FQL, we define fuzzy inference system for fuzzy evolutionary game as consisting of 4 rules of the following form.

- (1) IF C_0 is HG AND E_0 is HG THEN χ is HG.
- (2) IF C_0 is HG AND E_0 is ME THEN χ is ME.
- (3) IF C_0 is LO AND E_0 is ME THEN χ is ME.
- (4) IF C_0 is LO AND E_0 is HG THEN χ is ME.

There are a number of shapes that can be used for the membership function of each input such as trapezoidal and Gaussian shapes. We have chosen the Gaussian shape, since it is common in engineering applications and easy to use. A Gaussian fuzzy set membership degree in name is defined as follows:

$$\mu_A(x) = \exp\left(-\frac{(x-\varepsilon)^2}{2\sigma^2}\right), \quad (19)$$

where ε and σ are the fuzzy number mean and standard deviation and are assigned initially, $\mu_A : U \rightarrow [0, 1]$, for instance, to quantify the rule shown in Table 1 for the input $f(0.4, 0.5)$, using (19) to calculate the $\mu_A(0.4) = 0.14$, $\mu_A(0.5) = 0.01$, $\varepsilon = 0.2$, and $\sigma = 0.1$. The main remaining part is how to quantify the logical "and" operation that combines the meaning of two linguistic terms into a single premise. Consider

$$f_A = (0.4, 0.7) = \mu_A(0.4 \wedge 0.5) = \min\{0.14, 0.01\} = 0.01. \quad (20)$$

Finally, the numerical result of this fuzzy operation, defuzzification, is the last step in the operating procedure of the fuzzy inference mechanism; we use the most common method called a center of gravity. This method converts the fuzzy set into the value for which the area under the graph of the membership function, $\chi = f(C_0, E_b)$, is given by computing the center of gravity (CoG) of the area at the center:

$$\mu_A^* = \frac{\sum_{i=1}^n \mu_A(x_i) \times x_i}{\sum_{i=1}^n \mu(x_i)}. \quad (21)$$

4.2. Stochastic Learning Procedure. Here, we discuss obtaining the NE via stochastic evolutionary learning. As the attacker strategy is time-varying and the defense action is selected by each player simultaneously. We propose a decentralized algorithm based on stochastic evolutionary learning

TABLE 1: The fuzzy rule matrix.

E_0/C_0	LO	HG
LO	LO	LO
ME	ME	ME
HG	ME	HG

(SEL), by which the coordinator learn toward the equilibrium strategy from their individual action-reward history.

To facilitate the development of the SEL-based algorithm, let the mixed strategy $P_i(t) = [p_{i,1}(t), \dots, p_{i,M}(t)]$ the coordinator selection probability vector for player i , where $p_{i,a_i}(t)$ is the probability that player i selects strategy $a_i \in A_i$ at time t . The proposed self-organized defense algorithm by selecting coordinator is described in Algorithm 1.

Algorithm 1. Self-organized defense by selecting coordinator (SoDSC):

- (1) Initially, set $t = 0$ and the coordinator selection probability vector as $p_{i,a_i}(t) = 1/\Gamma_j$.
- (2) At every time t , each player selects an action $a_i(t)$ as the outcome of a probabilistic strategy based on $P_i(t)$.
- (3) The coalitions receive the instantaneous reward $v_i(t)$ specified.
- (4) Each coordinator in coalitions updates its selection probability vectors according to the following rules:

CID = getcurrent_node (ID)

CRS = Get_CoordinatorResourceState (CID) using FQL coordinator selection.

REPEAT

IF CRS = HG

$h_{i,t+1} \leftarrow h_{i,t}$ according to (12)

$p_{i,a_i}(t+1) \leftarrow p_{i,a_i}(t) + \varpi \cdot h_{i,t+1}(1_{\{a_i=d_i\}} - p_{i,a_i}(t))$

$Q(Z_{t+1}, a_{t+1})$

$\leftarrow Q(Z_t, a_t)$

$+ \omega \cdot \left(\sum_{i=1}^n (w_i \cdot c_{\Phi}^{AC}(i)) + \xi \cdot Q(Z_{t+1}, a_{t+1}) \right) \quad (22)$

$\cdot 1_{\{a_i=\Phi_{AC}\}} - Q(Z_t, a_t) \Big)$.

ELSE, IF CRS = ME

$h_{z,t+1} \leftarrow h_{z,t}$ according to (13)

$$p_{i,a_i}(t+1) \leftarrow p_{i,a_i}(t) + \omega \cdot h_{z,t+1} (1_{\{a_i=d_i\}} - p_{i,a_i}(t))$$

$$Q(Z_{t+1}, a_{t+1})$$

$$\leftarrow Q(Z_t, a_t) + \omega \cdot \left(\sum_{i=1}^n ((1 - w_i) \cdot (c_{\Phi}^{TC}(i) + c_{\Phi}^{TO}(i))) + \xi \cdot Q(Z_{t+1}, a_{t+1}) \cdot 1_{\{a_i=\Phi_{TCATO}\}} - Q(Z_t, a_t) \right). \quad (23)$$

(5) UNTIL value function converges,

where $0 < \omega < 1$ is the learning rate. $1_{\{i\}}$ is the indicator function. $h_{i,t+1}$ or $h_{z,t+1}$ is the normalized reward.

The instantaneous reward serves as a reinforcement signal so that a high reward brings a high probability in the next strategy update (Step 4). Also note that coordinator selection based on a probabilistic experiment (Step 2) might result in reconfiguration between different evolutionary rules in the beginning of the learning procedure. However, a stable long-term best response strategy for defending will be yielded after the learning period and the time required for convergence is a small fraction of the total operation time.

Algorithm 2. Get_CoordinatorResourceState (CID):

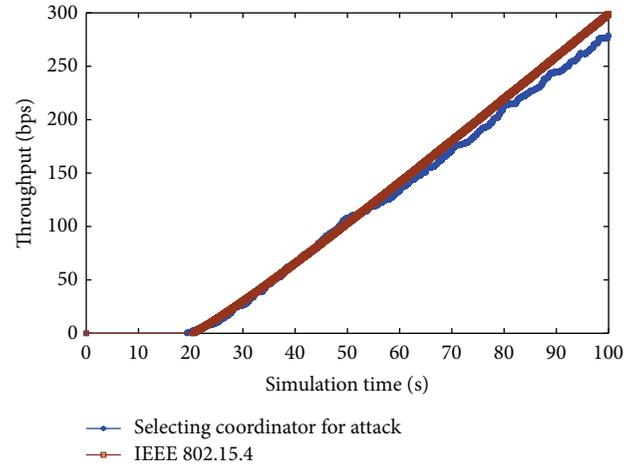
- (1) Initialization: C_o, E_b .
- (2) Use (19) to calculate $\mu_A(C_o)$'s and $\mu_A(E_b)$'s for $T(C_o)$ and $T(E_o)$, respectively, given by (17) and (18).
- (3) Combine $T(C_o)$ and $T(E_o)$ using Table 1 to form $O(\chi)$.
- (4) Calculate the $\mu_A(C_o \wedge E_b)$'s $O(\chi)$ resulting from step 3 as $\mu_A(O(\chi)) = \min\{\mu_A(C_o), \mu_A(E_b)\}$.
- (5) Calculate the output of defuzzification $\chi = f(C_o, E_b) = \mu_A^*$ according to (21).
- (6) Return $O(\chi)$.

Algorithm 2 describes the proposed fuzzy logic-based coordinator resource state decision algorithm from the point of view of a single coordinator node C_i . Moreover, we also assume that the relay C_i is able to read its battery level $E_{b,i}$ and estimate $C_{o,i}$ using the ACK message from the coordinator.

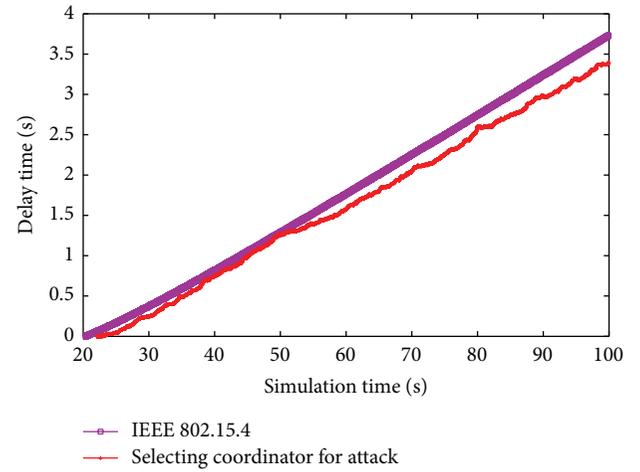
Proposition 3. *The SoDSC Algorithm converges to NE when the learning rate ω , ω is sufficiently small.*

Proof. Let limit P of the interpolated process satisfies the ODE [31] and consists of N probability vectors.

Let $P^d = (P_1^d, \dots, P_N^d)$ be the mixed strategy of all players, which are denoted by P_{ij}^d . Let $\Theta(P_i^d) = E[u_i^e]$ and $Y(P_j^d) = E[\Phi_j^c]$ be the expected response reward function of player i and the expected configuration function, respectively, over the mixed strategy P^d . Ψ also has the same number of



(a)



(b)

FIGURE 3: (a) The throughput related to the new topology for coordinator attacks. (b) The delay related to the new topology for coordinator attack.

mixed strategy which will be denoted as Ψ_{ij} . The component equations of (12)–(17) are

$$\begin{aligned} \frac{dp_{i,a_i}(t)}{dt} &= p_{i,a_i}(t) \sum_{a'_i} p_{i,a'_i}(t) [\Theta_i(\pi_i, P_{-i}) - \Theta_i(\pi_{i'}, P_{-i})] \\ \frac{dQ_{z,a_j}(t)}{dt} &= Q_{z,a_j}(t) \sum_{a'_j} q_{z,a'_j}(t) [Y_j(\pi_j, P_{-j}) - Y_i(\pi_j, P_{-j})] \\ \frac{d\Psi_{ij}(P^d)}{dt} &= \sum_i \sum_j \frac{\partial \Psi_{ij}(P^d)}{\partial p_{i,j}} \frac{dp_{i,a_i}(t)}{dt} \frac{dQ_{z,a_j}(t)}{dt} \\ &= p_{i,a_i}(t) \cdot p_{i,a'_i}(t) \cdot \Theta(\pi, P) \cdot Q_{z,a_j}(t) \\ &\quad \cdot q_{z,a'_j}(t) \cdot Y(\pi, P) \geq 0, \end{aligned} \quad (24)$$

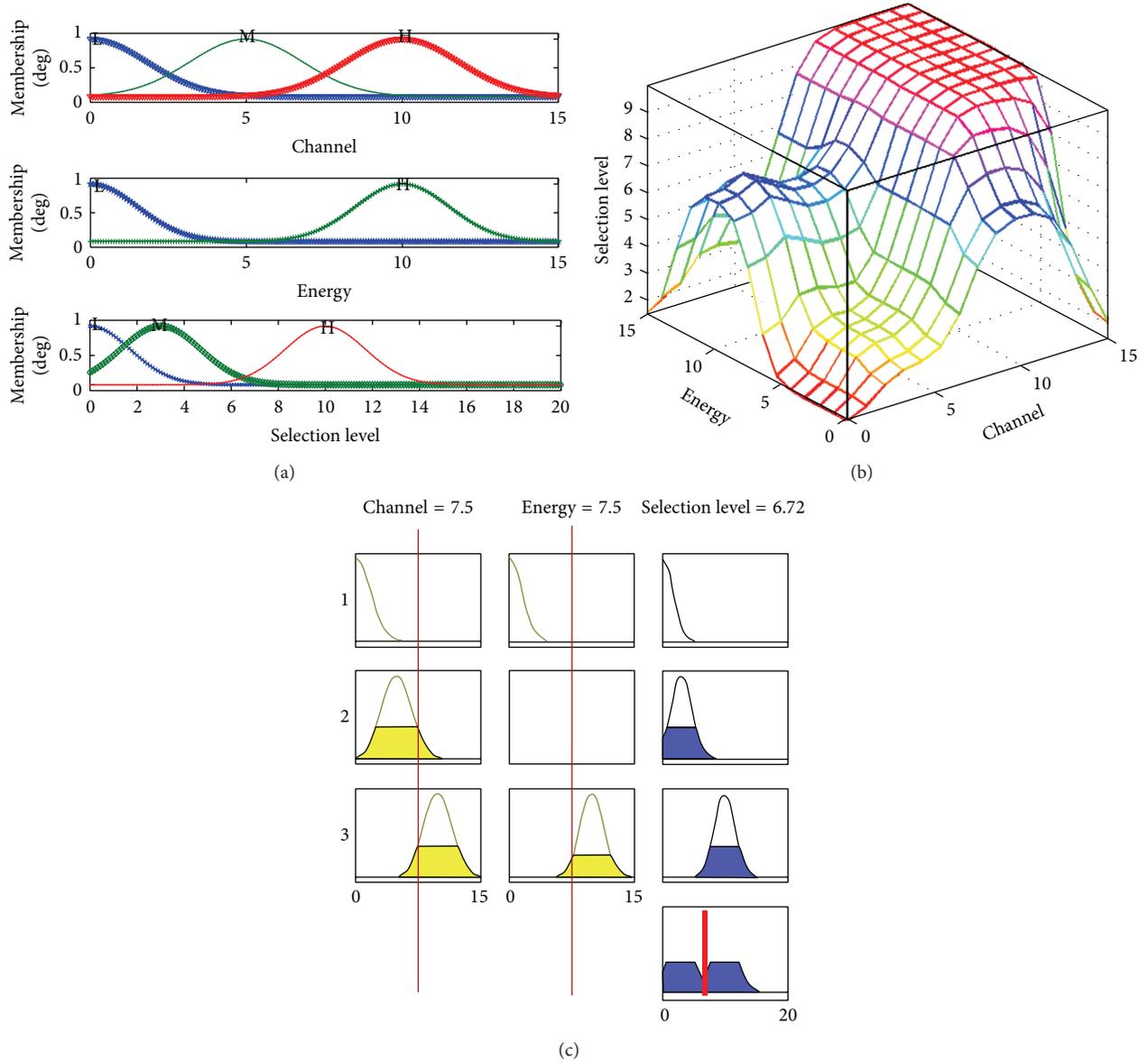


FIGURE 4: (a) Membership function for selection of coordinator. (b) The fuzzy inference for selection of coordinator. (c) The fuzzy inference process for selection of coordinator.

where

$$\begin{aligned} \Theta(\pi, P) &= \Theta_i(\pi_i, P_{-i}) - \Theta_i(\pi'_i, P_{-i}), \\ \Upsilon(\pi, P) &= \Upsilon_j(\pi_j, P_{-j}) - \Upsilon_j(\pi'_j, P_{-j}). \end{aligned} \quad (25)$$

$\Theta(\pi, P)$ and $\Upsilon(\pi, P)$ always have the same sign and are greater than zero. While the convergence to an NE is guaranteed as $\omega \rightarrow 0$, $\omega \rightarrow 0$. A smaller value of ω , ω leads to a slower convergence rate. A proper value of ω , ω can be numerically determined to strike the desired tradeoff between the accuracy and the rate of convergence for practical operations of the algorithm. \square

5. Simulation

An extensive simulation evaluation of dynamic defense and response strategy is reported in this section. We carry out our experiments using the Network Simulator 2 version 2.34 tool, which is a simulator implementing physical and MAC layers of the IEEE 802.15.4 standard. We first show that dynamic defense and response strategy increases average throughput by selecting unattacked coordinator to form a new network topology, hence a new coordinator, starting from the initial IEEE 802.15.4/ZigBee cluster trees. We also show the network lifetime increase over the basic IEEE 802.15.4/ZigBee configuration when fuzzy logic and evolutionary game for the defense response policy are applied to a network with coordinators failures owing to heavy attacks.

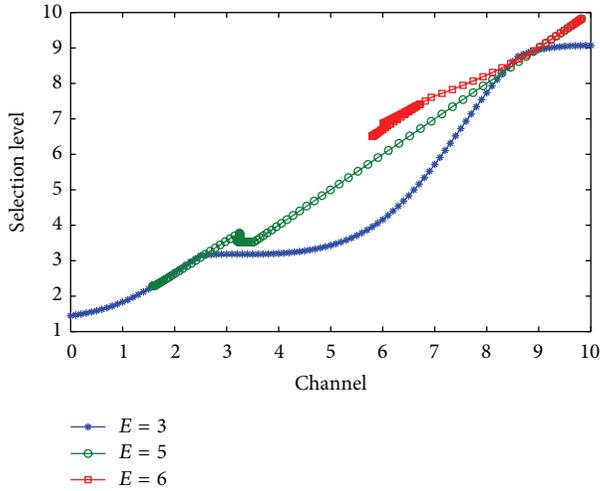


FIGURE 5: Effect of $E = 3$, $E = 5$, and $E = 6$ on the selection level for coordinator in face of jam attack.

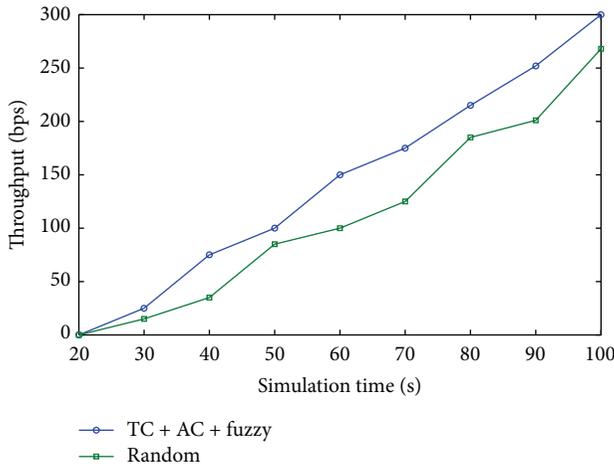


FIGURE 6: End-to-end throughput with $K = 20$ nodes.

TABLE 2: Simulation scenarios.

Parameters	Value
Protocols	AODV, Mac/802.15.4
Number of nodes	20
Simulation area	50 × 50
Traffic type	cbr, Poisson
Packet size	70 Bytes
Packets rate	250 k
Distance	5 m, 9 m, 10 m, 11 m, and 12 m
Simulation time	100 s

This section shows a comparison, in terms of jamming attack for cluster trees; network topology is formed with fuzzy logic and evolutionary game. We consider a network scenario consisting of N nodes, randomly deployed in a square area. The nodes' transmission range is reported in Table 2. At the beginning of each experiment the initial coordinator is randomly selected.

As for the physical layer Network Simulator-2 implements all primitives described in the IEEE 802.15.4 Standard and uses the two-ray ground propagation model. Each packet received at the physical layer should be above the receive threshold value, that is assumed to be equal to 3.24×10^{-10} W to be correctly received. By varying the position of coordinator node and jamming attack on the field, we repeat the IEEE 802.15.4 association procedure for N times. At the end of the procedure we record the throughput related to the new topology for coordinator attack. Figures 3(a) and 3(b) show throughput and delay of defense for coordinator attack, respectively, and the results for IEEE 802.15.4 and selection of coordinator using Algorithm 1. When a network has a high jam attack, coordinator improves the topology configuration for defending jam attack by game selection; game selection has the same throughput and delay as IEEE 802.15.4 that is not selection of coordinator in face of jam attack. This is because the game selection Algorithm 1 uses TC and AC rules to select coordinator and keep higher throughput to approximate to IEEE 802.15.4 that does not face jam attack. Moreover, shorter routing paths can be established between any node and the coordinator, which saves node's energy and reduces data delivery delay.

Figures 4(a), 4(b), and 4(c) show the fuzzy inference for selection coordinator in face of jam attack. Figure 5 shows energy effect of $E = 3$, $E = 5$, and $E = 6$ on the selection level for coordinator in face of jam attack. Figure 6 shows that, by selecting coordinator, the throughput of networks is increased, for different selection coordinator methods for defending jam attack. Performance of the proposed algorithm shows that, by game selection and fuzzy inference, the throughput of networks is increased up to 300 bps (for a Zigbee network with 20 nodes) with TC + AC + Fuzzy rules, while the throughput of networks is increased up to 275 bps with random selection coordinator.

6. Conclusion

We have presented a coordinator selection scheme for ZWSNs to defend action from malicious nodes and minimize the cost of wireless transmission energy. By exploiting coordinator selection among multiple sensor nodes, the path security and reliability can be improved and the cost of data transmission can be reduced by selecting rules. We have formulated a game-theoretic model for joint dynamic defense and response strategy, taking into account the fact that each sensor node is rational to maximize its own payoff. The proposed game model is composed of two formulations, that is, a stochastic game for dynamic attack response and evolutionary game for coordinator selecting. The solutions of these games can achieve Nash equilibrium for the attack response strategy game.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China under Grant no. 61272034; Zhejiang Provincial Natural Science Foundation of China under Grant nos. LY12F02019 and LY13F030012; Research Start-up Foundation of Jiaying University under Grant no. 70512020; and Scientific Research Foundation of Zhejiang Provincial Education Department of China under Grant no. Y201431192.

References

- [1] H. Yang, Y. Qin, G. Feng, and H. Ci, "Online monitoring of geological CO₂ storage and leakage based on wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 2, pp. 556–562, 2013.
- [2] T. Torfs, T. Sterken, S. Brebels et al., "Low power wireless sensor network for building monitoring," *IEEE Sensors Journal*, vol. 13, no. 3, pp. 909–915, 2013.
- [3] N. K. Suryadevara and S. C. Mukhopadhyay, "Wireless sensor network based home monitoring system for wellness determination of elderly," *IEEE Sensors Journal*, vol. 12, no. 6, pp. 1965–1972, 2012.
- [4] D. E. Tiliute, "Security of mobile ad hoc wireless networks: a brief survey," *Advances in Electrical and Computer Engineering*, vol. 7, no. 2, pp. 37–40, 2007.
- [5] N. L. S. Da Fonseca, "Second quarter 2009 IEEE communications surveys and tutorials," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 2, pp. 1–2, 2009.
- [6] C. Tang and D. O. Wu, "An efficient mobile authentication scheme for wireless networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 4, pp. 1408–1416, 2008.
- [7] Zigbee specification, 2006, <http://www.zigbee.org/>.
- [8] C. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.
- [9] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, pp. 234–244, London, UK, 1994.
- [10] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. F. Korth, Eds., pp. 153–181, Kluwer Academic Publishers, 1996.
- [11] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, New Orleans, La, USA, February 1999.
- [12] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks," in *Proceedings of the Hawaii International Conference System Sciences*, vol. 8, pp. 1–10, 2000.
- [13] S. Lindsey and C. S. Raghavendra, "PEGASIS: power-efficient gathering in sensor information systems," in *Proceedings of the IEEE Aerospace Conference*, vol. 3, pp. 1125–1130, Big Sky, Mont, USA, March 2002.
- [14] J. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, 2004.
- [15] M. Zimmerling, W. Dargie, and J. M. Reason, "Energy-efficient routing in linear wireless sensor networks," in *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS '07)*, pp. 1–3, Pisa, Italy, October 2007.
- [16] F. Shao, X. Shen, and L. Cai, "Energy efficient reliable routing in wireless sensor networks," in *Proceedings of the 1st International Conference on Communications and Networking in China (ChinaCom '06)*, pp. 1–5, Beijing, China, October 2006.
- [17] H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad hoc networks," in *Proceedings of the IEEE INFOCOM*, pp. 22–31, Tel-Aviv, Israel, 2000.
- [18] F. Cuomo, A. Abbagnale, and E. Cipollone, "Cross-layer network formation for energy-efficient IEEE 802.15.4/ZigBee Wireless Sensor Networks," *Ad Hoc Networks*, vol. 11, no. 2, pp. 672–686, 2013.
- [19] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 56–73, May 2000.
- [20] Y. W. Law, L. van Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '05)*, pp. 76–88, November 2005.
- [21] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *IEEE Network*, vol. 20, no. 3, pp. 41–47, 2006.
- [22] Z. Yao, D. Kim, and Y. Doh, "PLUS: parameterized and localized trust management scheme for sensor networks security," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS '06)*, pp. 437–446, October 2006.
- [23] E. Aivaloglou and S. Gritzalis, "Hybrid trust and reputation management for sensor networks," *Wireless Networks*, vol. 16, no. 5, pp. 1493–1510, 2010.
- [24] A. Gabrielli, L. V. Mancini, S. Setia, and S. Jajodia, "Securing topology maintenance protocols for sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 3, pp. 450–465, 2011.
- [25] F. Bao, I. Chen, M. Chang, and J. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 169–183, 2012.
- [26] S. A. Zonouz, H. Khurana, W. Sanders, and T. Yardley, "RRE: a game-theoretic intrusion response and recovery engine," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 395–406, 2013.
- [27] L.-C. Tseng, F.-T. Chien, D. Zhang, R. Y. Chang, W.-H. Chung, and C. Y. Huang, "Network selection in cognitive heterogeneous networks using stochastic learning," *IEEE Communications Letters*, vol. 17, no. 12, pp. 2304–2307, 2013.
- [28] C. X. Jiang, Y. Chen, and K. J. R. Liu, "Distributed adaptive networks: a graphical evolutionary game-theoretic view," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5675–5688, 2013.
- [29] S. Skaperdas, "Contest success functions," *Economic Theory*, vol. 7, no. 2, pp. 283–290, 1996.
- [30] J. M. Smith, *Evolution and The Theory of Games*, Cambridge University Press, Cambridge, UK, 1982.
- [31] P. S. Sastry, V. V. Phansalkar, and M. A. L. Thathachar, "Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 5, pp. 769–777, 1994.

Research Article

Calculating the Number of Cluster Heads Based on the Rate-Distortion Function in Wireless Sensor Networks

Mingxin Yang,^{1,2} Jingsha He,³ and Yuqiang Zhang¹

¹ College of Computer Science and Technology, Beijing University of Technology, Beijing 100124, China

² College of Economics Management, Hebei University of Science and Technology, Shijiazhuang 050018, China

³ School of Software Engineering, Beijing University of Technology, Beijing 100124, China

Correspondence should be addressed to Mingxin Yang; ymxspj@163.com

Received 14 March 2014; Revised 15 June 2014; Accepted 17 June 2014; Published 1 July 2014

Academic Editor: Fei Yu

Copyright © 2014 Mingxin Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to limited resources in wireless sensor nodes, energy efficiency is considered as one of the primary constraints in the design of the topology of wireless sensor networks (WSNs). Since data that are collected by wireless sensor nodes exhibit the characteristics of temporal association, data fusion has also become a very important means of reducing network traffic as well as eliminating data redundancy as far as data transmission is concerned. Another reason for data fusion is that, in many applications, only some of the data that are collected can meet the requirements of the sink node. In this paper, we propose a method to calculate the number of cluster heads or data aggregators during data fusion based on the rate-distortion function. In our discussion, we will first establish an energy consumption model and then describe a method for calculating the number of cluster heads from the point of view of reducing energy consumption. We will also show through theoretical analysis and experimentation that the network topology design based on the rate-distortion function is indeed more energy-efficient.

1. Introduction

Wireless sensor networks (WSNs) have become more and more widely used in a variety of applications. In most large-scale WSNs, individual wireless sensor nodes will first transmit sensed data to cluster heads which will then forward the data to the sink node. Due to limited resources in the sensor nodes, power consumption has become a primary consideration during data transmission. In general, it would consume more energy for data transmission than for data processing. Therefore, reducing the amount of data transmission is a very important means of reducing the total amount of energy consumption in WSNs. Among possible approaches for reducing data transmission, hierarchical network topology based data fusion has been considered as an effective means of reducing communication traffic. With data fusion, a certain level of distortion may have to be tolerated by actual applications, which makes it unnecessary to transmit all the collected data to the sink and the amount of data that should be transmitted to the clusters and to the sink depends on the level of distortion that can be tolerated. Hierarchical

data fusion topology, which can be developed based on the rate-distortion function, can help to reduce communication traffic while making sure that the amount of data collected and transmitted is sufficient to meet the requirement of real applications.

Castanedo summarized the state of the data fusion research and described many relevant studies [1]. Hall and Llinas introduced the emerging technology of multisensor data fusion [2]. Huang et al. proposed a novel weight-based clustering decision fusion algorithm (W-CDF) to detect target signal in wireless sensor networks [3]. Abdulsalam and Ali introduced a new data aggregation algorithm for uniform, nonuniform, and evolving networks while maintaining data accuracy [4]. Ahvar et al. proposed an energy-aware routing protocol (ERP) for query-based applications in WSNs, which offers the tradeoff between traditional energy balancing and energy saving objectives and supports soft real-time packet delivery [5]. Yang et al. proposed a method for achieving an optimal number of aggregation points with a power consumption model and analyzed the effect of different numbers of aggregation points on the performance [6]. But

the work did not consider the issue of distortion. Without measurement on data compression, there may be too much or too little information to the sink node. A large amount of data will lead to redundant information while causing an unnecessarily high level of energy consumption. Although a small amount of data can reduce energy consumption, the sink node may not be able to restore the original message, making the data less useful for the sink node. Akyildiz et al. introduced the concept of wireless sensor networks that have been made viable by the convergence of microelectromechanical system technologies, wireless communications, and digital electronics [7]. Deng and Huang established a communication model and analyzed energy consumption under two different circumstances, that is, collecting data once per round and collecting data several times per round [8] in which an optimal data collection scheme is designed by determining the optimal times of data collection to optimize data acquisition for hierarchical networks. The work also analyzed the differences among data acquisition schemes by assuming that all the sensor nodes have the same initial energy condition in WSNs. However, this paper did not study the exact corresponding number of aggregators. Heinzelman et al. studied the application of the networks in harsh network environment with severe resource constraints and proposed application-specific protocol architecture in contrast to the traditional layered approaches [9]. Yang et al. proposed a more reasonable energy consumption model, that is, the optimal energy consumption model (OECM) [10]. Yu et al. proposed a method to design an optimal path to acquire data in sparse wireless sensor networks based on a multiplicatively weighted Voronoi diagram [11].

In this paper, we propose a method for the calculation of the number of cluster heads based on the rate-distortion function after establishing an energy consumption model according to the data fusion framework in WSNs. Our energy consumption model includes three parts: data transmission from wireless sensor nodes to the cluster heads, data compression or aggregation in the cluster heads, and data transmission from the cluster heads to the sink. We will evaluate our proposed method on energy consumption based on the above established energy consumption model and the rate-distortion function to demonstrate the energy efficiency of the method.

The remainder of this paper is organized as follows. In Section 2, we introduce some preliminary knowledge required for our discussion which includes the theoretical derivation on the computability of the number of cluster heads, the concept of distortion, and the rate-distortion function. In Section 3, we introduce an energy consumption model and propose a method for calculating the number of cluster heads based on the rate-distortion function. In Section 4, we show that the design of the network topology based on the rate-distortion function is more energy-efficient than that without considering the distortion. In Section 5, we describe some experiment that we have performed and present and analyze the simulation results. Finally, in Section 6, we conclude this paper in which we also discuss some future research directions.

2. The Preliminaries

2.1. Computability of the Number of Cluster Heads. In WSNs, hierarchical topology for data fusion is generally preferred in which each round of the collection process will result in a fixed number K of sensor nodes as the cluster heads. At the beginning of each round of the process, every sensor node generates a random number between 0 and 1 and compares the random number with a probability value $P_i(t)$. If the random number is smaller than $P_i(t)$, the sensor node will periodically broadcast an ADV message to its neighboring nodes to inform that it will be the cluster head. The formula for the probability value $P_i(t)$ [12] is defined as follows:

$$P_i(t) = \begin{cases} \frac{K}{N - K * (r \bmod (N/K))} & C_i(t) = 1 \\ 0 & C_i(t) = 0, \end{cases} \quad (1)$$

where $P_i(t)$ is the probability that node i would act as the cluster head at time t . Let N denote the number of sensor nodes in a WSN, K denote the number of cluster heads at each round, and r denote the current working round. $C_i(t)$ would indicate whether node i has the right to become a cluster head at time t . When $C_i(t) = 1$, node i is entitled to become a cluster head at time t and when $C_i(t) = 0$ node i is not entitled to become a cluster head at time t .

It is clear that each node will be able to function as a cluster head once within N/K rounds. Every node has the opportunity to serve as the cluster head; those nodes that have already served as the cluster heads in the first round can no longer serve as the cluster heads in the next $N/K - 1$ rounds. Those nodes that can serve as the cluster head fall off; the probability that a remaining node can become a cluster head would go up. After $N/K - 2$ rounds, the probability that the remaining nodes who never serve as cluster head can become a cluster head would be 1.

Lemma 1. For a WSN with N nodes, if there are K clusters upon completing each round, then $P = K/N$ and every node can become a cluster head once during N/K rounds.

Proof. In round $(r + 1)$, if the probability that a remaining node can become a cluster head at time t is $P_i(t)$, the expectation of the cluster head denoted as N_{ch} is as follows:

$$E(N_{ch}) = \sum_i^N P_i(t) * 1 = K. \quad (2)$$

Since the number of nodes that have not served as the cluster heads in the previous r rounds is $N - K * r$, after N/K rounds, every node should become a cluster head exactly once. Regarding the meaning of $C_i(t)$, symbol $\sum_{i=1}^N C_i(t)$ represents the total number of nodes that can serve as the cluster heads at time t and we can then get the following formula:

$$E \left[\sum_{i=1}^N C_i(t) \right] = N - K * \left(r \bmod \frac{N}{K} \right). \quad (3)$$

According to Formulas (2) and (3), we can get the mathematical expectation for the cluster head number, which is $E(N_{ch})$:

$$\begin{aligned}
 E(N_{ch}) &= E\left(\sum_i^N P_i(t) * C_i(t)\right) \\
 &= \sum_i^N P_i(t) * E\left(\sum_i^N C_i(t)\right) \\
 &= \left(N - K * \left(r \bmod \frac{N}{K}\right)\right) \\
 &\quad \times \frac{K}{N - K * \left(r \bmod \frac{N}{K}\right)} \\
 &= K,
 \end{aligned} \tag{4}$$

where N_{ch} is the number of cluster heads, $E(N_{ch})$ is the expectation of cluster head number, $P_i(t)$ is the probability that node i will act as a cluster head at time t , and $C_i(t)$ indicates whether node i has the right to function as a cluster head at time t . Again, N is the number of sensor nodes in a WSN, K is the number of cluster heads after each round, and r is the current working round. \square

2.2. The Rate-Distortion Function. Generally, it is not necessary to transmit every piece of the collected data to the sink. As a result, a certain level of information distortion may occur, which must be under the tolerance level of the sink. For a given source entropy $H(X)$ and allowed distortion, the amount of information from the source should be as small as possible, which is derived as the theoretical value from the information rate-distortion function.

2.2.1. The Function. Let us define the discrete information source as follows [13]:

$$\begin{bmatrix} X \\ P \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ p(x_1) & p(x_2) & \cdots & p(x_n) \end{bmatrix}. \tag{5}$$

The output sequence after transmission through a channel is $y = [y_1 \ y_2 \ \cdots \ y_m]$. The distortion function is a nonnegative function which is a quantitative description of the receiver y_j from source x_i . Then, let us arrange all the $d(x_i, y_j)$, where $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$, and the resulting matrix $[d]$ can be expressed as follows:

$$[d] = \begin{bmatrix} d(x_1, y_1) & d(x_1, y_2) & \cdots & d(x_1, y_m) \\ d(x_2, y_1) & d(x_2, y_2) & \cdots & d(x_2, y_m) \\ \vdots & \vdots & \vdots & \vdots \\ d(x_n, y_1) & d(x_n, y_2) & \cdots & d(x_n, y_m) \end{bmatrix}. \tag{6}$$

This matrix $[d]$ is called the distortion matrix.

In the matrix, the nonnegative function $d(x_i, y_j)$ can be selected to meet specific needs, such as the squares cost function, the absolute cost function, and the uniform cost function.

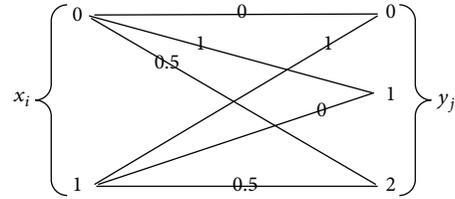


FIGURE 1: The distortion measurement flow.

2.2.2. Distortion Measurement Flow. The distortion function matrix is also called the distortion matrix $[D]$, where the upper limit of the distortion can be calculated based on the distortion matrix.

Let us suppose that $X \in \{0, 1\}$, $Y \in \{0, 1, 2\}$, $d(0, 0) = d(1, 1) = 0$, $d(0, 1) = d(1, 0) = 1$, $d(0, 2) = d(1, 2) = 0.5$. The distortion measurement flow can then be described in Figure 1.

The corresponding distortion matrix is then $D = \begin{bmatrix} 0 & 1 & 0.5 \\ 1 & 0 & 0.5 \end{bmatrix}$.

2.2.3. Information Rate-Distortion Function. Suppose the rate of information transmission through a channel with capacity C is R . If $R > C$, information at the source should be compressed or aggregated so that the compressed transmission rate R' is lower than the channel capacity C . Let us assume that the predetermined average distortion is D and the average distortion of the compressed source is \bar{D} ; for a given source, we should make the amount of the information transmitted as small as possible. All the channels that can satisfy the criterion $\bar{D} \leq D$ are called the permitted channels B_D .

We can therefore find a channel $p(Y/X)$ among the permitted B_D channels so that the channel transmission rate $I(X, Y)$ is minimized for a given source to push information through this channel. All the channels that can meet the above condition are called the rate-distortion function, namely,

$$R(D) = \min_{p(Y/X) \in B_D} I(X, Y). \tag{7}$$

For a discrete memoryless source, the rate-distortion function can then be expressed as follows [13]:

$$R(D) = \min_{p(y_j/x_i) \in B_D} \sum_{i=1}^n \sum_{j=1}^m p(x_i) p\left(\frac{y_j}{x_i}\right) \cdot \log \frac{p(y_j/x_i)}{p(y_j)}, \tag{8}$$

where $p(x_i)$ ($i = 1, 2, \dots, n$) is the probability distribution at the source, $p(y_j)$ ($j = 1, 2, \dots, m$) is the probability distribution at the receiver, and $p(y_j/x_i)$ ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$) is the transition probability distribution.

2.2.4. The Rate-Distortion Function for Different Types of Sources. An information source is a source for generating information or information sequence. Actually, there may be many information sources. The output of these sources is the information of a single symbol. Therefore, the number of such

symbols is limited and countable. We hence use x , which is a one-dimensional discrete random variable, to describe the output of the information source, which is called a discrete source. When the output of the source is a continuous function, which means that the value of the source is both continuous and random, the information source is called a continuous information source. A discrete source would include a probability source, such as a binary source and an n -element source. A continuous source would include a Gaussian source and so on. There is an upper bound on the degree of distortion and the rate-distortion function of several sources among which D is the distortion, σ^2 is the mean square error, α is the value of the distortion function, and $R(D)$ is the rate-distortion function as described below.

(1) For a binary source:

$$\begin{aligned}
 R(D) &= H(p) - H\left(\frac{D}{\alpha}\right) \\
 &= -p \ln p - (1-p) \ln(1-p) \\
 &\quad + \frac{D}{\alpha} \ln \frac{D}{\alpha} + \left(1 - \frac{D}{\alpha}\right) \ln \left(1 - \frac{D}{\alpha}\right),
 \end{aligned} \tag{9}$$

where $\left[\begin{smallmatrix} X \\ p(x_i) \end{smallmatrix} \right] = \left[\begin{smallmatrix} x_1 & x_2 \\ p & 1-p \end{smallmatrix} \right] p \leq 1/2$, and

$$[D] = \begin{bmatrix} 0 & \alpha \\ \alpha & 0 \end{bmatrix} \alpha > 0. \tag{10}$$

(2) For two-element equal probability source:

$$D_{\max} = \frac{1}{2} \alpha \tag{11}$$

$$R(D) = \ln 2 - H\left(\frac{D}{\alpha}\right), \tag{12}$$

where $p = 1/2$ and $[D] = \begin{bmatrix} 0 & \alpha \\ \alpha & 0 \end{bmatrix} \alpha > 0$.

(3) For an n -element equal probability source:

$$D_{\max} = \left(1 - \frac{1}{n}\right) \alpha \tag{13}$$

$$R(D) = \ln n + \frac{D}{\alpha} \ln \frac{D/\alpha}{n-1} + \left(1 - \frac{D}{\alpha}\right) \ln \left(1 - \frac{D}{\alpha}\right), \tag{14}$$

where $p(x_i) = 1/n, (i = 1 \sim n)$ and $d(x_i, y_j) = \begin{cases} 0 & i=j \\ \alpha & i \neq j \end{cases}$.

(4) For a one-dimensional Gaussian source:

$$R(D) = \begin{cases} \frac{1}{2} \log_2 \frac{\sigma^2}{D} & D < \sigma^2 \\ 0 & D \geq \sigma^2 \end{cases} \tag{15}$$

3. Calculating the Number of Cluster Heads of Data Fusion

3.1. Number of Cluster Heads Based on the Rate-distortion Function. Assuming that all wireless sensor nodes are distributed in a circular area with radius “ a ”, and the sink node is located in the center of the circle, and that there are one or more clusters in the same circular area. The process of transmitting data from regular sensor nodes to the sink node is to that of transmitting the data to the corresponding cluster heads, and then aggregating on the sink node along the way. Thus, the transmission paths form a hierarchical network. Assuming also that the center of the circular area that is covered by cluster C is denoted by a node (x_0, y_0) and the distance that the sensor nodes in cluster C can transmit data to the sink node is S .

In the following formula, assuming that a is the radius of the circular area C , α' is the energy consumption coefficient, α_1 is the loop energy consumption coefficient, α_2 is the antenna energy consumption coefficient, n is the number of wireless sensor nodes in the circular area, r is the rate of data transmission, δ is the routing influence coefficient, k is the number of cluster heads in the circular area, d_{char} is the regional characteristic radius, γ is the compression ratio, c is the number of over-compression, and β is data compression coefficient, then, if one circle with radius a consists of k number of circular clusters and the radius of each of the clusters is x , we can get the following formula:

$$k\pi x^2 = \pi a^2 \tag{16}$$

$$x = \frac{a}{\sqrt{k}}. \tag{17}$$

We can further get the formula for distance S as follows:

$$\begin{aligned}
 S &= \frac{n \times \delta}{\pi a^2} \times \iint_{(x,y \in C)} \sqrt{(x-x_0)^2 + (y-y_0)^2} d_x d_y \\
 &= \frac{n \times \delta}{\pi a^2} \times \int_0^{2\pi} d_\theta \int_0^{a/\sqrt{k}} \sqrt{r^2} \times r dr \\
 &= \frac{n \times \delta}{\pi a^2} \int_0^{2\pi} d_\theta \times \frac{1}{3} \times r^3 \Big|_0^{a/\sqrt{k}} \\
 &= \frac{n \times \delta}{\pi a^2} \times \frac{1}{3} \times \frac{a^3}{k^{3/2}} \times \theta \Big|_0^{2\pi} \\
 &= \frac{2an\delta}{3k^{3/2}}.
 \end{aligned} \tag{18}$$

The amount of energy consumed by a network consists of three parts: P_1 , which is for the wireless sensor nodes in each circular area to transmit data to the cluster heads, P_2 , which is for the cluster heads to receive the data, and P_3 , which is for the cluster heads to transmit the data to the sink node. The formula for P_1 in terms of k fusion nodes can be expressed as follows:

$$P_1 = \frac{2\alpha' anr\delta}{3k^{1/2}}. \tag{19}$$

With an acceptable distortion D , the minimum amount of data $R(D)$ is the amount of data transmitted and received by cluster heads. If β is the energy coefficient of data fusion at the cluster heads, the energy consumption for classical fusion is proportional to the amount of compressed data; that is,

$$P_2 = \beta R(D). \quad (20)$$

If the density of cluster heads is $k/\pi a^2$ in a circular area with k cluster heads, with the assumption of a linear compression model for the cluster heads, the following formula would hold:

$$P_3 = (\gamma R(D) + c) \times \frac{2k\alpha'a}{3}. \quad (21)$$

We could then calculate energy consumption E based on Formulas (19), (20), and (21) as follows:

$$E = P_1 + P_2 + P_3. \quad (22)$$

$$E = \frac{2\alpha' anr\delta}{3k^{1/2}} + \beta R(D) + (\gamma R(D) + c) \times \frac{2k\alpha'a}{3}. \quad (23)$$

Since our purpose is to calculate k in order to minimize E , we can force $E = 0$ and the process is as follows:

$$\begin{aligned} E &= \frac{2\alpha' anr\delta}{3k^{1/2}} + \beta R(D) + (\gamma R(D) + c) \times \frac{2k\alpha'a}{3} \\ &- \frac{1}{2} \times \frac{2\alpha' anr\delta}{3k^{3/2}} + (\gamma R(D) + c) \times \frac{2\alpha'a}{3} = 0 \\ \frac{\alpha' anr\delta}{3k^{3/2}} &= \frac{2\alpha'a(\gamma R(D) + c)}{3} \\ \frac{nr\delta}{k^{3/2}} &= 2(\gamma R(D) + c) \\ k^{3/2} &= \frac{nr\delta}{2(\gamma R(D) + c)} \\ k &= \left(\frac{nr\delta}{2(\gamma R(D) + c)} \right)^{2/3}, \end{aligned} \quad (24)$$

where there exists $\alpha' = (\alpha_1 + \alpha_2 d_{\text{char}})/d_{\text{char}}$.

3.2. Examples Using the Rate-Distortion Function

(1) When there exist two element probability sources such that $[p(x)] = \{1/2 \ 1/2\}$ and $D = \{0 \ \alpha\}$, we get the following:

$$D_{\max} = \frac{1}{2} \times \alpha + \frac{1}{2} \times 0 = \frac{\alpha}{2}, \quad D_{\min} = 0.$$

$$\begin{aligned} R(D) &= \ln 2 - H\left(\frac{D}{\alpha}\right) \\ &= \ln 2 - \left[\frac{D}{\alpha} \ln \frac{D}{\alpha} + \left(1 - \frac{D}{\alpha}\right) \ln \left(1 - \frac{D}{\alpha}\right) \right]. \end{aligned} \quad (25)$$

When $D_{\max} = \alpha/2$, $R(D_{\max}) = 2 \ln 2$ and $k = (nr\delta/2(2\gamma \ln 2 + c))^{2/3}$.

When $D_{\min} = 0$, $R(D_{\min}) = R(0) = R(D)_{\max} = 1$ and $k = (nr\delta/2c)^{2/3}$.

When $R(D)_{\min} = 0$, $k = (nr\delta/2c)^{2/3}$.

(2) When there exists a one-dimensional Gauss source that meets the mean square error distortion criterion $R(D) = \begin{cases} (1/2)\log_2(\sigma^2/D) & D < \sigma^2 \\ 0 & D \geq \sigma^2 \end{cases}$, we get the following:

When $D = 0$, $R(D) \rightarrow \infty$.

When $D = \sigma^2$, $R(D) = 0$.

4. A Model for Energy Consumption Based on the Rate-Distortion Function

4.1. The Network Energy Consumption Model. We reference the energy model in the LEACH protocol in our study, which consists of two phases: cluster establishment phase and stable data transmission phase. Regarding the different types of energy consumption, we assume that there are electron energy consumption, energy consumption of the power amplifier when a node transmits data, and electron energy consumption which can occur only when a node receives data in a WSN. If S_{elec} is the energy consumption for transmitting or receiving one bit of data, lS_{elec} is then the energy consumption for transmitting or receiving an l -bit message. Our power amplifier consumption adopts the free space model (FS) and the multipath fading model (MP) according to the distance between the sources and the sink node. When the distance d between two nodes is shorter than a threshold value, the FS model is applied. When the distance d between two nodes is longer than a threshold value but shorter than the maximum communication distance d' , the MP model is applied. Therefore, energy consumption S_s of a node's sending an l -bit message is as follows:

$$S_s = \begin{cases} lS_{\text{elec}} + l\epsilon_{fs}d^2 & d < d_0 \\ lS_{\text{elec}} + l\epsilon_{mp}d_{\text{to-BS}}^4 & d_0 \leq d < d'. \end{cases} \quad (26)$$

Assuming that there are n nodes and k clusters in a WSN, the distance from a node in the circular area to the base station is $d_{\text{to-BS}}$, where $d_0 \leq d_{\text{to-BS}} < d'$, and the distance from the same node in the circular area to the cluster head is $d_{\text{to-clus}}$, where $0 < d_{\text{to-clus}} < d_0$; if energy consumption during the establishment phase is S_0 , then $S_0 = (k/n)S_1 + (1 - (k/n))S_2$, where S_1 is the energy consumption of the cluster head and S_2 is the energy consumption of the nodes within the cluster during the cluster establishment phase.

During the stable data transmission phase, if the energy consumption of a cluster head for receiving the information from the nodes within the cluster is S_{re} , then

$$S_{\text{re}} = lS_{\text{elec}}. \quad (27)$$

Similarly, if the energy consumption for data transmission from the cluster head to the base station is $S_{f\text{-BS}}$, then

$$S_{f\text{-BS}} = lS_{\text{elec}} + l\epsilon_{mp}d_{\text{to-BS}}^4. \quad (28)$$

If the energy consumption for data transmission from sensor nodes to the cluster head is $S_{f\text{-clus}}$, then

$$S_{f\text{-clus}} = l s_{\text{elec}} + l \varepsilon_{fs} d_{\text{to-clus}}^2. \quad (29)$$

The whole WSN will die when the first node exits the network due to the depletion of energy. Therefore, we consider the amount of energy of the node that first dies as the amount energy of the whole network.

4.2. Proof of the Validity of the Model. We now show that the topological structure of data fusion based on the rate-distortion function is more energy-efficient than those not allowing any distortion in WSNs.

The energy consumption of the entire network consists of three parts: energy consumption for cluster establishment, energy consumption of the cluster heads for receiving data, and energy consumption of the cluster head for sending the data to the sink. Therefore, the following formula holds:

$$E_i = r_n + q \left[\left(\frac{n}{k} - 1 \right) S_{\text{re}} + \frac{n}{k} S_{\text{ag}} + S_{f\text{-BS}} \right] + (r_n - q) S_{f\text{-clus}}, \quad (30)$$

where E_i is the initial energy of the node that dies first, r_n is the total number of rounds, q is the number of elected cluster heads, S_{ag} is the energy consumption of a cluster head for compressing one message, $S_{f\text{-BS}}$ is the energy consumption for transmitting data from a cluster head to the base station, and $S_{f\text{-clus}}$ is the energy consumption for transmitting data from a node to the cluster head. According to formula (22), the following formula holds:

$$\begin{aligned} E_i &= r_n + q \left[\left(\frac{n}{(nr\delta/2(\gamma R(D) + c))^{2/3}} - 1 \right) S_{\text{re}} \right. \\ &\quad \left. + \frac{n}{(nr\delta/2(\gamma R(D) + c))^{2/3}} S_{\text{ag}} + S_{f\text{-BS}} \right] \\ &\quad + (r_n - q) S_{f\text{-clus}} \\ &= r_n + q \left[\left(\frac{n^{1/3} \times [2(\gamma R(D) + c)]^{2/3}}{(r\delta)^{2/3}} - 1 \right) S_{\text{re}} \right. \\ &\quad \left. + \frac{n^{1/3} \times [2(\gamma R(D) + c)]^{2/3}}{(r\delta)^{2/3}} S_{\text{ag}} + S_{f\text{-BS}} \right] \\ &\quad + (r_n - q) S_{f\text{-clus}}. \end{aligned} \quad (31)$$

From the above formula, the total amount of energy consumption of the whole network is proportional to $R(D)$. If there exist $D \in [0, D_{\text{max}}]$ and $R(D)_{\text{max}} = R(D_{\text{min}}) = R(0)$, $R(D)$ would get the maximum value when $D = 0$. That is, when there is no distortion requirement, E_i attains its maximum value. Therefore, a WSN design based on the rate-distortion function is better than a one without considering distortion for the purpose of saving energy.

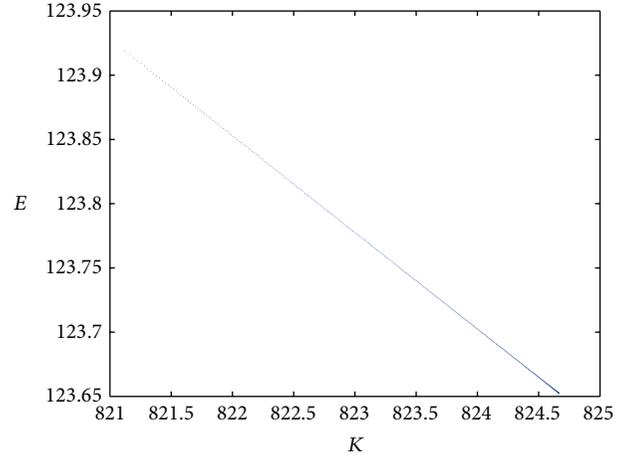


FIGURE 2: The total amount of network energy consumption of two-element source based on the rate-distortion function.

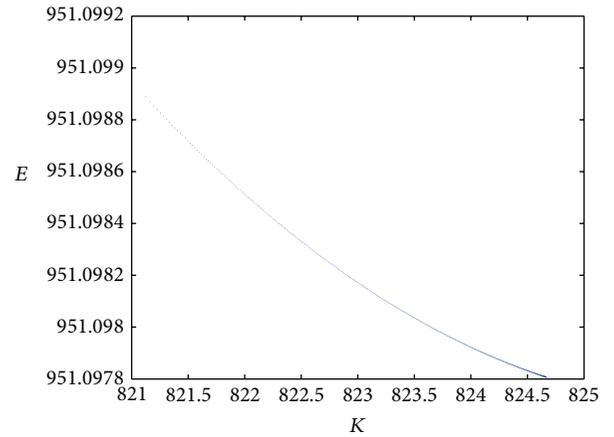


FIGURE 3: The total amount of network energy consumption without considering the rate-distortion function.

5. Experiment and Simulation Results

The purpose of our experiment, which was performed using Matlab, is to evaluate our proposed method by comparing it with the method proposed by Yang et al. [6] in terms of the total amount of network energy consumption. The simulation parameters are set as follows: $n = 10000$, $a = 1000$ m, $\alpha_1 = 5 \times 10^{-8}$ J/b, $\alpha_2 = 1 \times 10^{-10}$ J/b, $d_{\text{char}} = 22.36$, $\beta = 5 \times 10^{-8}$ J/b, $\gamma = 30\%$, $c = 32$ bit, $\delta = 0.95$, and $r = 160$ b/sec.

From Figures 2 and 3, we can see that the amount of energy that is consumed decreases along with the increase of K ; that is, the energy consumption decreases along with an increase in the number of fusion nodes or cluster nodes. For the same number K in the range [821, 825], the total amount of network energy consumption of in the case of the two-element source based on the rate-distortion function shown in Figure 2 is lower than that shown in Figure 3 which does not employ the rate-distortion function. The experiment result is in line with the conclusion of the last section. Moreover, under the same simulation environment, the best

result occurs when the number of fusion nodes is 821 for the Gaussian source. In a word, the method proposed in this paper is very suitable for the two-element source from the viewpoint of network energy consumption.

6. Conclusion

In this paper, we proposed a method for calculating the number of cluster heads based on the rate-distortion function. According to different requirements on information distortion and an established energy consumption model, the exact number of the cluster heads can be calculated for the purpose of data fusion. We showed that the proposed method is more effective through the means of mathematical proof. We also performed some analysis on the simulation results by using Matlab to demonstrate that the energy consumption of the model based on the rate-distortion function would consume less energy than the one that does not consider the factor of information distortion. In the future, we will perform the experiment and analysis based on some real network data to further improve the efficiency as well as energy consumption of our model for data fusion in WSNs.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work presented in this paper has been supported by the National Natural Science Foundation of China (61272500), Beijing Natural Science Foundation (4142008), and Prelaunch of Beijing City Government Major Tasks and District Government Emergency Projects (Z131100005613030).

References

- [1] F. Castanedo, "A review of data fusion techniques," *The Scientific World Journal*, vol. 2013, Article ID 704504, 19 pages, 2013.
- [2] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.
- [3] H. P. Huang, L. Chen, X. Cao, R. C. Wang, and Q. Y. Wang, "Weight-based clustering decision fusion algorithm for distributed target detection in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 192675, 9 pages, 2013.
- [4] H. M. Abdulsalam and B. A. Ali, "W-LEACH based dynamic adaptive data aggregation algorithm for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 289527, 11 pages, 2013.
- [5] E. Ahvar, S. Ahvar, G. M. Lee, and N. Crespi, "An energy-aware routing protocol for query-based applications in wireless sensor networks," *The Scientific World Journal*, vol. 2014, Article ID 359897, 9 pages, 2014.
- [6] J. Yang, S. Q. Zhang, X. L. Zhang, and F. Ding, "Study on method of determining data aggregators number for wireless sensor network," *Microcomputer Information (Control & Automation)*, vol. 22, no. 1–9, pp. 161–163, 2006.
- [7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [8] X. Y. Deng and J. Huang, "Optimal data acquisition scheme about LEACH," *Journal of Southeast University*, vol. 42, no. 1, pp. 20–24, 2012.
- [9] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [10] K. Yang, Y.-M. Wu, and H.-B. Zhou, "Research of optimal energy consumption model in wireless sensor network," in *Proceedings of the 2nd International Conference on Computer Engineering and Technology (ICCET '10)*, pp. V7-421–V7-424, Chengdu, China, April 2010.
- [11] R. Y. Yu, X. W. Wang, and X. S. Yi, "Energy-balancing data collection based on the multiplicatively weighted voronoi diagram in sensor networks," *Journal of Northeastern University (Natural Science)*, vol. 30, no. 12, pp. 1718–1722, 2009.
- [12] Z. Sheng and S. Q. Xie, *Probability Theory and Mathematical Statistics*, Higher Education Press, Beijing, China, 1979.
- [13] M. Q. Zhou, *Information Theory (3)*, Aerospace University Press, Beijing, China, 2006.

Research Article

Countermeasures to Avoid Noncooperation in Fully Self-Organized VANETs

Jezabel Molina-Gil, Pino Caballero-Gil, and Cándido Caballero-Gil

Faculty of Mathematics, University of La Laguna, 38271 Tenerife, Spain

Correspondence should be addressed to Pino Caballero-Gil; pcaballe@ull.es

Received 10 March 2014; Revised 1 June 2014; Accepted 2 June 2014; Published 26 June 2014

Academic Editor: Fei Yu

Copyright © 2014 Jezabel Molina-Gil et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The secure and efficient exchange of information in vehicular ad hoc networks (VANETs) involves more challenges than in any other type of ad hoc networks. This paper proposes a new vehicular communication system based on mobile phones for fully distributed and decentralized networks. In these networks, communications depend on individual nodes, which could decrease the efficiency and reliability of transmitted information. Besides, the limitation in the resources of mobile devices is an additional obstacle in the forwarding problem, and the content of the information generated by individual nodes must be considered inherently unreliable. In particular, this paper proposes the application of groups as a basis structure for a cooperation mechanism useful in event generation and in packet retransmission. Its aim is to promote the involvement of nodes in network performance. Given that such participation involves consumption of node resources, a group-based structure is here used not only to reduce communication overload but also to prevent sending false information and to encourage nodes in relaying packets. Several simulations of the proposal have been done, and the results have confirmed that this is a promising approach to increase network efficiency and trust in transmitted information, while reducing the number of selfish nodes in VANETs.

1. Introduction

Vehicular ad hoc networks (VANETs) have been proposed as a solution based on intervehicles communication to prevent adverse circumstances on the roads and to increase efficiency in traffic management. In order to turn this type of networks into a reality that helps to improve road safety, several security communication tools are necessary to protect them from many possible types of attacks such as attacks to jeopardize the connectivity and attacks to modify forwarded information.

The main goal of this paper is the proposal of new tools that allow the protection of VANETs against such attacks, ensuring as far as possible that generation and retransmission of information are done correctly. This paper, based on two previous works of the authors [1, 2], includes an analysis of various cooperation mechanisms needed to deploy a reliable and functional VANET where the nodes are fully autonomous and independent. It describes different techniques for promoting cooperation. Specifically, different

strategies to motivate nodes to participate in relaying packets properly and to ensure greater availability and quality of the network are here presented. Besides, it deals with the need to protect the content of the relayed information by using reactive groups that ensure the accuracy of the information without involving any significant delay. In order to reduce the time required to generate the information, a group-based structure is proposed to handle more efficiently the generated packets.

The rest of the paper is organized as follows. Related work is overviewed in Section 2. Section 3 describes secure event generation through cooperative groups and includes an analysis of implementation results from simulations. Section 4 addresses relevant cooperation factors in packet forwarding. Finally, conclusions are given in Section 5.

2. Related Work

Research on cooperation in VANETs can be classified mainly into two categories according to the way of encouraging

participation of nodes in the network: through incentives or through reputation. In incentive-based systems, nodes pay and charge for participating in the network. The work in [3] proposes the use of virtual credit in incentive schemes to stimulate packet forwarding in general mobile ad hoc networks. A system where vehicles receive an incentive for forwarding and carrying advertisements is described in [4]. Besides, [5] discusses some unique characteristics of incentive schemes specifically thought for VANETs and focuses on a receipt-based reward scheme. On the other hand, in reputation systems, malicious nodes are detected and isolated from the network. In [6], an event-based system is used to prevent nodes from spreading false traffic messages by determining whether incoming traffic messages are significant and trustworthy to the driver. The work in [7] describes a mechanism for detecting possible malicious nodes through the use of three different modules whose sum determines node reputation. The proposal called VARS [8] uses direct and indirect trust as well as appended opinions to enable confident decisions on event packets. Another interesting reputation system is described in [9], where trust relationships and packet-acceptance decisions are based on instant observation and relaying behaviour of nodes.

However, all the aforementioned tools, including the reputation system proposed in [10], require certification authorities, which are responsible for delivering public/private keys and certificates. In particular, [11] proposes that a regional transportation authority plays such a role. Therefore, none of those solutions can be considered applicable to fully distributed and decentralized networks such as the ones discussed in this work. The closest references to this paper are [12, 13], where cooperative groups coordinate actions taken by multiple vehicles to make optimal decisions. Reference [14] proposes a novel approach to compose a trustworthy group in a P2P model, which is fully distributed and scalable. On the other hand, the work in [15] uses routing for communications and introduces cooperation as a service, based on a cluster structure. Finally, [16] proposes a flocking scheme for a group of vehicles, which focuses on their decentralized coordination so that they can cooperate in complex environments.

Many references on VANETs discuss the problem of packet forwarding. Those works assume the existence of a sender and a receiver because the routing issue where intermediate nodes are used to perform connections between any pair of nodes is one of the most basic networking issues in ad hoc networks. Reference [17] classifies routing protocols into single-black hole attack and collaborative-black hole attack, analyses the categories of these solutions, and provides a comparison table. A remarkable result of the research is a tool for driving assistance that allows creating a real and safe vehicular ad hoc network by using only smartphones called VAIpho (VANET in Phones) [1, 2]. This scheme offers the possibility of deploying a real and secure VANET quickly and economically. In VAIpho, no specific routing protocol is required because broadcast is used to disseminate information.

3. Cooperative Groups

Most proposals to improve security in VANETs use cryptographic tools such as authentication, key management, or pseudonyms. However, these schemes do not provide any solution to ensure that event information generated within the network is true, which means that, even if all nodes are cooperative and forward messages, it is possible that such messages are announcements of false events. In this section, we analyse the possibility of using cooperative groups to prevent generating and forwarding false announcements. A group is here defined as a set of vehicles that are located in a close geographical area. Many papers propose the formation of groups to avoid overloading the network by reducing communication overload, and so forth. In [18], the authors propose to divide roads into cells so that groups are formed by vehicles within the same cell. However, the tasks of dividing a highway into cells by using GPS to determine the group of each vehicle and choosing a leader for each group are time-consuming tasks and require heavy computation. Thus, they are nonappropriate in fully self-organized VANETs.

The proposal here presented uses the idea of reactive group formation to generate events cooperatively so that groups are formed when an event is detected. This not only allows improving the reliability of transmitting information, but also reduces channel overload because the number of generated packets decreases.

3.1. Security Mechanisms. Since our proposal is self-organized, the device associated to each network node n must be able to generate a pair of private and public keys in a decentralized way. In order to obtain a signed certificate for its public key, it contacts with several legitimate network nodes and if these nodes confirm that node n is trustworthy, a certificate with its private key is generated. Like what was presented in [19], a similarity can be found between the proposed security mechanisms and social networking sites like Facebook, Twitter, and so forth.

The system asks the new node for information about its contacts in some social network and looks for registered and trustworthy nodes in the system. The results are showed to the new node, which selects the nodes to sign its certificate. During the signing process, the system exchanges the generated certificates so that each user holds the signature generated by itself and the signatures produced by its friends.

3.2. Group Formation and Leader Election. An important issue about the formation of groups is the choice of the group leader and the nodes that belong to each group. The lack of centralized structures and the dynamism of the environment imply a complex challenge.

When a node detects an incident, it sends a message to all nodes within its transmission range, to form a reactive group. This node will be the leader of the group (see Figure 1). G denotes a group of nodes n_i with location $\text{loc}(n_i, T)$, which are

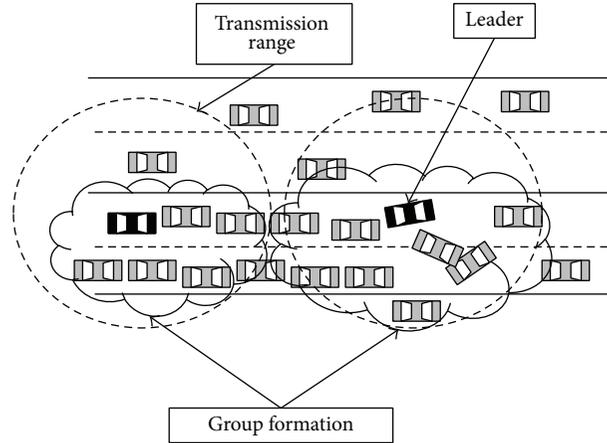


FIGURE 1: Group formation and leader election.

within the leader’s transmission range R at the time T when it detects an incident:

$$\begin{aligned}
 N &= \{n_1, n_2, n_3, \dots\}, \\
 G &= \{n_i \in N; \text{loc}(n_i, T) \leq R\}.
 \end{aligned}
 \tag{1}$$

During group formation, three different types of messages can be distinguished.

- (i) *Detection Message*. It is sent by a node when it detects an incident, in order to form a group. It contains information used by nodes to determine whether the event is true. It will be detailed in the next subsection. Nodes receiving it will assume that the origin node is the group leader.
- (ii) *Signature Message*. It is signed by neighbouring nodes and sent to the group leader in response to a detection message, indicating agreement with received data.
- (iii) *Event Message*. It is sent by the group leader to all nodes, containing all the received signatures and the incident information. This message is stored and broadcast by all receiving nodes.

It is possible that two or more nodes begin the group formation process. This situation could produce more than one leader in a group. The solution to this problem can be reached through a cooperative decision between leaders. In particular, each node that generates a detection message must store the timestamp of the generation. If any node within its transmission range receives this packet and has previously generated an identical packet, meaning that it is another potential leader, it compares both packet timestamps. In this way, the oldest one corresponds to the leader. This solution has been tested in VAIpho implementation with successful results.

3.3. Event Generation. Before sending information about an event that has been detected, a verification process is necessary to prevent forwarding fraudulent information. This process is performed in three steps, which correspond to the

three types of messages described above. This process is part of the proposed scheme and is performed within each group as described below.

The proposal allows nodes inside a group to decide whether some received data are true or not, by matching them with the perceived information. If both pieces of information match, the node signs the received message indicating that it agrees with the content of the message. Thus, in order to generate an event, all vehicles that are part of the cooperative group must agree with the information they receive from their neighbourhood. The protocols to make this process would imply an overload of the channel and a significant delay of communications if no mechanism to minimize them was adopted. In our proposal, each vehicle processes locally the received events before making a decision based on a three-stage sequential scheme. The first one is the decision-making stage, corresponding to detection and signature message, where each vehicle locally processes the message received about an event from the leader, verifies it, and compares it with the information it receives directly from its environment. The second stage is data aggregation. Once the leader concludes that it has received enough signed messages about the existence of an event, it applies an aggregation method to combine them. This process was described in [20]. Finally, an event message is generated and forwarded through the network to reach as many users as possible.

In order to decide whether the received data match the information perceived by the receiver, the proposed decision-making stage is based on four dimensions: time, location, direction of travel, and speed. Firstly, the time interval in which an event is valid must be taken into account. Secondly, the location of an event, on the road or on the map, is an essential parameter. Finally, the direction of travel in which the event is located must be distinguished due to the usual existence of two-way traffic. Additionally, since VAIpho can detect traffic jams, another interesting parameter is speed. Thus, in the decisions-making stage, each node follows rules based on the above four parameters.

The first verification is the date of the event. If it is an expired event, the node drops the packet. The second test

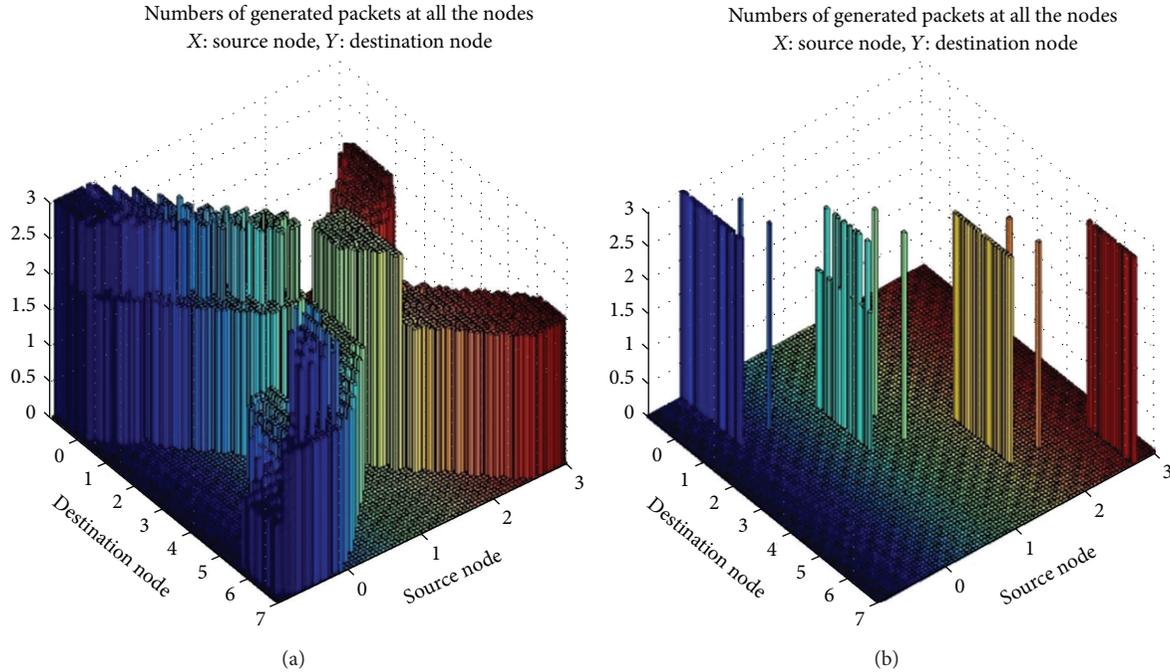


FIGURE 2: Number of generated packets with and without cooperation groups.

parameter is the location of the event. The node checks the coordinates (X, Y, Z) of the received packet and compares them with its current coordinates (X, Y, Z) . If the vehicle is within the event range, it must be able to detect the event. The third test is the direction of travel. In two-way traffic roads, a traffic jam is possible in a direction where the opposite direction is completely free of vehicles. In order to do this test, nodes must verify the direction of travel included in the packet and compare it with their direction of travel. Finally, a speed check is performed. If a node receives a packet reporting the existence of a traffic jam on the road in which it is circulating, the node checks for a period of time if its speed ($Speed_{current}$) is below the maximum speed of the road ($Speed_{max}$) divided by a factor, which in the implementation was equal to 4. If so, it concludes that there is a traffic jam:

$$\text{if } Speed_{current} \geq \frac{Speed_{max}}{4}, \quad \text{there is no traffic jam,} \quad (2)$$

otherwise, there is a traffic jam.

In the last step, after receiving enough of signed messages, the leader adds all the signatures, generates an event message, and sends it to all nodes within its transmission range. At this moment, all nodes that receive this message must validate the information through the verification of signatures. Nodes that receive this validated event store it in their database and follow the store-and-carry paradigm. In this way, they forward the packet within the network and prevent the spread of false information.

3.4. Performance Evaluation. In order to analyse how fast and effective the event generation module is and how the

proposed group-based structure affects the implementation, several NS-2 simulations have been done based on data got from a real device implementation. This section presents some details and results obtained by averaging 100 simulations using different network sizes over the same area of 2000 square meters, that is to say, considering different situations regarding traffic density. Simulations were done in networks between 1000 and 10000 nodes. The most relevant parameters selected for the simulation have been given as follows: total number of lanes for each direction = 3, simulation time = 1000 s, moment when motions start = 0 s, moment when relaying begins = 40 s, transmission period = 10 s, transmission range = 100 m, and travelled distance before the event happens = 800 m. Speeds and directions of nodes in the NS-2 simulations were random.

The main goal of the simulations was to evaluate, on the one hand, the number of generated packets using reactive groups, and, on the other hand, the effects of our proposal in the computational complexity, that is to say, the time our cooperative mechanism takes to warn all network nodes about the existence of an event. For the purpose of these simulations, we used and compared the proposal with a basic scheme without any cooperation mechanism, where each node that detects or receives an event sends it to all nodes in its transmission range.

The first simulations correspond to a traffic jam and the corresponding warning packet forwarding with and without reactive groups. Figure 2 shows that the number of generated packets in the simulation without groups is much higher than when using reactive groups, even though in this case such a number includes the packets generated by the group formation algorithm. The decrease in the number of

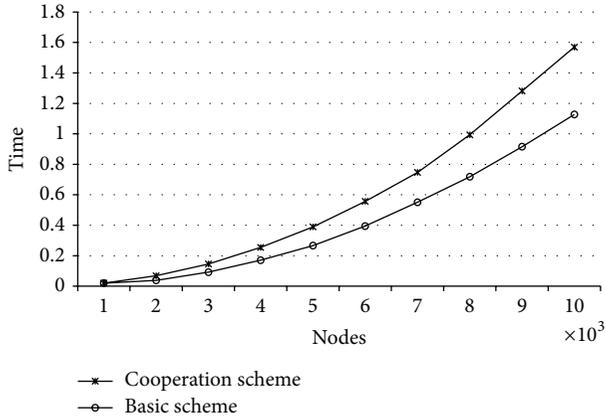


FIGURE 3: Time cost.

generated packets allows making better use of the channel. Thus, Figure 2 shows how using reactive groups can reduce the number of generated packets and improve the channel usage.

Connections between vehicles in VANETs are usually short, so any proposed mechanism requiring communication between vehicles has to be fast enough to prevent data loss in communications. In Figure 3, we can see the impact of node density in time cost of communication both with the basic scheme and with the proposed cooperation scheme. We can conclude that the use of the group-based mechanism does not involve a significant increase in time cost of management when the packet is received.

The difference between the basic scheme and the cooperation scheme depends on the computational cost required to form the group structure when a new event is detected. However, the reduction in the number of generated packets involves a relevant communication improvement. Another sensitive aspect of the implementation is that nodes must check the accuracy of the received information in order to avoid attacks. This leads to the verification of some signatures, which explains the delay it causes. However, this delay does not involve a significant increase in time. Finally, when the number of nodes increases, the time for processing packets also increases.

4. Cooperation in Packet Forwarding

4.1. The Problem. In order to spread information, VAIpho users relay packets to all the vehicles in the same group. This implies an important resource consumption, which in the specific case of VAIpho involves the problem of using devices with limited battery. Furthermore, in VANETs, nodes can move at very high speeds, so links can be intermittent. This requires that, according to the store-and-carry approach [5], nodes store the events and exchange them when they meet other vehicles, which could produce also a problem of storage space.

As we see below, if the number of malicious users who do not cooperate in forwarding packets is not high compared to the total number of users, they will not pose a big threat to

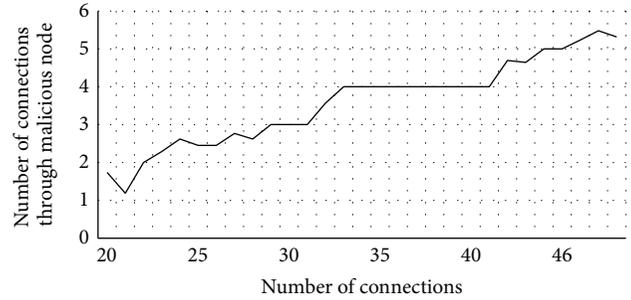


FIGURE 4: Connections with 5% of malicious nodes.

the network. Figure 4 shows the effect of these users in the performance of the network. In particular, we have simulated a network with 100 nodes where around 5% of them do not forward packets. Nodes in the simulations move randomly, and communications between them are also random, making around 20 and 50 connections for each simulation. From the obtained data, we estimate that the probability of meeting a malicious node is 12% in the worst case. Therefore, under these conditions, malicious nodes cannot be considered a major threat to the proper performance of the network. However, VAIpho users might choose not to relay packets or to turn off their devices to save power, and this would seriously affect the network because these vehicles would no longer be part of the network. Indeed, the greater the number of nodes that collaborate in forwarding packets, the better the performance of the VANET.

Given the aforementioned limitation of battery and storage of mobile phones, it is possible that nodes try to act passively by only receiving information from the network rather than storing and forwarding packets. These users would benefit from information relayed in the network without participating in it, so they passively damage and degrade network performance, endangering its connectivity.

Hence, a specific module is necessary in VAIpho to encourage nodes in network cooperation. Users must be motivated to keep their devices turned on and forward packets properly, which is accomplished by offering a new incentive scheme to encourage participation of nodes as part of VAIpho. This aim deals with two factors that may influence packet forwarding due to selfish and/or passive behaviour. One of these factors is battery consumption, which can affect packet forwarding during sending packets to inform neighbouring nodes. Another factor is the limitation of the storage space, which is required to store all the events that are generated and/or received.

4.2. Forwarding Cooperation. The proposed protocol uses the idea of reactive groups where nodes in the same transmission range join together in the same group. This idea of groups allows reducing the number of packets in the network because the leader is in charge of making decisions about the packets that reach its group. This prevents from having multiple retransmissions of the same packet.

When a node detects a traffic jam, it sends the information, and the nearby receiving vehicles start to disseminate it

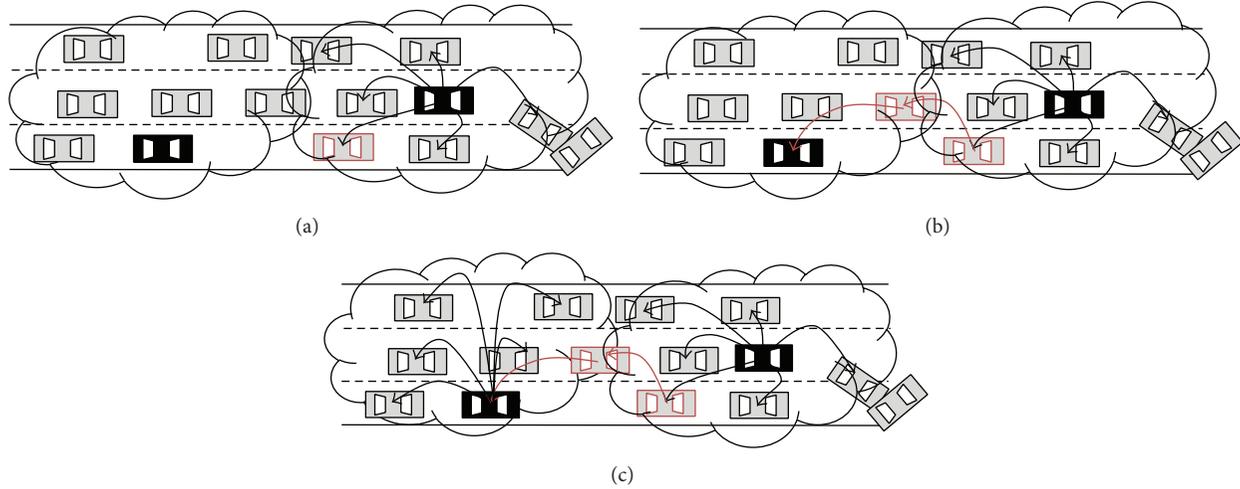


FIGURE 5: Packet forwarding.

to other vehicles while they are moving by using the leaders of the groups. These packets are forwarded for a certain period of time and distance from the source provider.

For this type of packets, the detector node sends the packet to all vehicles that are in its scope. It will be the leader of the group in its range of coverage; see Figure 5(a). In this case, the number of generated packets is relatively high because the aim of such packets is to provide information to as many vehicles as possible. If all vehicles were devoted to covering this type of packets without any control, the network would be overloaded. Although the number of generated packets is large, thanks to the idea of groups, the leader manages to relay them in an orderly manner. The leader will receive the packet and will be responsible for broadcasting it within its group. The result is that no nodes will receive the same packet many times from its neighbours because if a leader receives a packet that had been previously received, it will not relay it again to its group. Moreover, the leader will seek a route within its group to a neighbour group through one of the members of its group nodes (see Figure 5(b)). Hence, it achieves a reduction of both the number of retransmissions between groups and the number of retransmissions inside the group (see Figure 5).

In this proposal, both the leader of the group and the node selected to send the packet to other groups consume more resources than other vehicles inside the group. For this reason, an incentive scheme is necessary to encourage the nodes to be leaders of the groups and to relay information to other groups when they are chosen to do it. In order to reach this aim, a new incentive scheme is proposed inspired by a micropayment scheme [21], where each payment for a forwarding service can be thought as a lottery ticket where, for example, the award could be gas vouchers.

As we are going to explain below, nodes relaying packets will receive a lottery ticket. This ticket allows that both the authority and the winner node can determine whether it is a winning ticket or not. Besides, this method encourages relaying packets because the prizes will not be only for the

node with the winning ticket but also for the node that received the forwarded packet.

4.2.1. Incentive Scheme. The presented model proposes a scheme based on a type of lottery in which each node has a non-null probability of winning. We denote by n the current node, N_i the child nodes that n broadcasts the packet. The node that detects an event generates a packet that contains a unique identifier $PackID$, the information $Information$, and a hash code H computed randomly with a certain size:

$$[PackID | Information | H]. \quad (3)$$

When a node n receives the packet, it checks the information. If n decides to participate in the forwarding, it sends the message to other nodes and waits for a receipt rec_{N_i} justifying that it sent the packet to their children nodes N_i . Then, the node n computes for each child node a receipt rec_{N_i} with a hash on $PackID$, $NodeID_N$ y rec_{N_i} and checks the result against H :

$$h(PackID | NodeID_N | rec_{N_i}) = H. \quad (4)$$

If equality (4) is fulfilled in one of these verifications, then the node n is a winner. We denote by $Prob_h$ the probability that a hash on $PackID$ concatenated with $NodeID$ and the receipts rec_{N_i} that child nodes send to a relaying node collide with a value H :

$$Prob_h = Prob [h(PackID | NodeID_N | rec_{N_i}) = H]. \quad (5)$$

It is assumed that a node can receive only one reward. The probability of a relaying node winning a prize $Prob_p$ in forwarding packets to N_c nodes, where it received the packet from a number of nodes N_f , can be defined as follows:

$$Prob_p = (N_c \cdot N_f) \cdot Prob_h. \quad (6)$$

As showed in (6), a node can get a reward if it computes the hash of the packet with the receipts from some of its

children and gets a winning code. Furthermore, a node can get also a reward if it sends the winner receipt to its father. Therefore, a node can transmit packets or receipts to get an award. Hence, the greater the number of retransmissions is, the greater the probability of winning is. In this way, nodes are motivated to cooperate. Moreover, this mechanism will motivate child nodes to send the receipts to node N . However, if we analyse the probability, we find that, when a node gets a winner receipt, it will not broadcast more packets since the previous function is restricted to an only one reward for winner. This behaviour would not be desirable since the objective is to motivate relaying all packets. In order to solve this problem, we propose the use of hash function with nonnegligible probability of collision, which leads to the possibility of having more than one winner receipt for each packet:

$$\begin{aligned} \exists i \neq j : h(\text{PackID} \mid \text{NodeID}_N \mid \text{rec}_{Ni}) \\ = h(\text{PackID} \mid \text{NodeID}_N \mid \text{rec}_{Nj}) = H. \end{aligned} \quad (7)$$

In this case, a node could win the lottery for each packet it relays. Thus, nodes can win more than once with the same packet. On the other hand, nodes can also win the lottery with every receipt they return to their parent node. Similar to the previous case, nodes could win the lottery with one or more receipts. Hence, the problem that once a node wins the award it could stop retransmitting the same packet is solved because if a node wins a prize, this does not mean that it cannot win another prize.

4.2.2. The Leader Problem. It is not difficult to assume that no node wants to be leader because in such a case the number of packets it would have to handle is greater than being any other node belonging to the group. However, if we analyse the used mechanisms, we will conclude that in both cases being a leader provides a greater reward than being another node in the group.

According to the introduced group structure, a leader receives all the packets in its group. Hence, it will have a bigger probability to receive receipts that produce a hash collision with H so that it could be a winner node. The probability of a leader to win a prize in a group consisting of $|G|$ nodes, which receive the packet from one node of the group, could be defined as follows:

$$|G| \cdot \text{Prob}_h. \quad (8)$$

Therefore, this provides an incentive for leaders to propagate packets because the higher the number of retransmissions, the greater the probability of winning a reward. As a leader, the packets are broadcast to all members of its group, the model promotes that nodes prefer to be leaders, and consequently this mechanism motivates nodes to become leaders and cooperate.

4.3. Battery Consumption. In order to ensure proper network functionality, it is necessary to provide users with updated information. Thus, the nodes must cooperate in the delivery

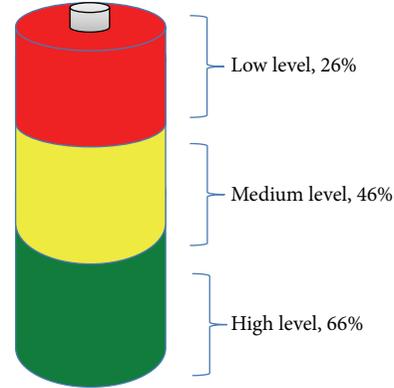


FIGURE 6: Limit of battery level.

of events to all nodes that are met during their life on the network.

If users detect that a mobile phone application consumes too much battery and it does not allow a normal use of the phone, they will turn off the application. This situation would seriously affect the performance of the network. Hence, users must be motivated to keep VAIpho application turned on as much as possible. An advantage of VAIpho regarding battery consumption compared to other existing mobile phone applications is that VAIpho does not use any 3G connections, which is currently the largest battery consuming service. Besides, the offer of an incentive mechanism, the reception of useful information in real time, and the possibility to choose a battery level below which the application automatically stops working are three VAIpho features that motivate users to keep their devices turned on as long as possible.

Three battery levels are here proposed for the decision about whether to cooperate in relaying packets or not. The user may establish these levels, which, as shown in Figure 6, in the implementation, have been defined: high (66%), medium (46%), and low (26%). Thus, when the battery level drops below the threshold selected by the user called limit of battery level (LBL), nodes do not cooperate in packet forwarding and other network operations, and the remaining level of battery is used for other purposes.

In order to determine the approximate battery consumption involved in running VAIpho, we made a study with real devices. Considering that the used services that represent the highest battery consumption on the application are GPS, Wi-Fi, and Bluetooth, we analysed and got the following data. GPS has a consumption of about 25% battery every hour, while the use of Wi-Fi and Bluetooth gives a sum of 10% every 8 hours, representing a total of 1.25% per hour. Taking these data as a starting point, we made some simulations. In particular, nodes with different battery levels were considered. In the first case, all nodes initially started the simulation with a high level of battery between 75 and 100%; in the second case, they had medium levels between 46 and 75%, and the third case was with low battery levels: 25–46%. Moreover, nodes had different LBL uniformly distributed according to the three levels defined above. During the simulation, nodes consumed battery, so when they reached

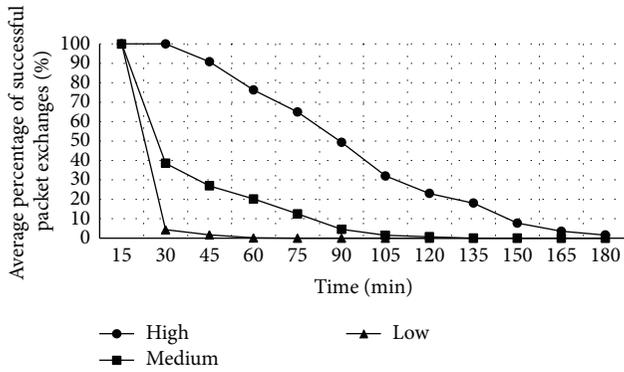


FIGURE 7: Battery consumption of VAIpho.

their chosen LBL, they turned off the VAIpho application and did not cooperate more. The main aim of the simulation consisted in analyzing the network performance for three different cases. The first one is a simulation where nodes are in an optimal case because they have very high levels of battery. The worst scene is where all nodes start with very low battery level. Finally, an intermediate state is also considered.

Figure 7 shows the relationship between the time of application usage without recharging the battery and the average percentage of successful packet exchange, depending on the three defined battery levels. In the simulations, the number of nodes was 100, and about 1000 connections were established every 15 minutes. In particular, Figure 7 shows the averages obtained from simulations. In particular, in every case, the battery level set by the users in order to stop VAIpho was as follows:

- (i) when the battery drops below the level considered high, for 33 nodes,
- (ii) when the battery drops below the level considered medium, for 33 nodes,
- (iii) when the battery drops below the level considered low, for 34 nodes.

As we can see in Figure 7, the percentage of successful connections decreases as battery decreases. The worst case is when all nodes have initially very low battery level. However, considering that only 34 vehicles or fewer were running the application, because the remaining 66 stopped it according to their stipulated battery level threshold, the results were pretty good. In particular, the simulations showed that users could freely set their preferred thresholds to stop VAIpho without compromising the network performance, so it may be considered an effective countermeasure to prevent noncooperation. Besides, taking into account that the average trip time by car is about 22 minutes each way, reaching 50% already after 90 minutes is a sufficiently high percentage that guarantees an appropriate quality of the offered services.

Although the results regarding battery consumption shown by the simulations are good, the use of GPS with a 25% of battery consumption per hour could be considered inadequate for the proposal if the application has to run for a long time with no recharge. A solution to this problem can be

based on the proposal presented in [22] based on the use of Wi-Fi and cell tower triangulation for location, which provide fast and lower energy location methods.

4.4. Storage Space. A decision about whether to store or to drop packets requires taking into account several complex features such as traffic density, type of road (conventional or highway), and time (rush hours). These measures are quite complex and would require a scheme to implement some artificial intelligence method. However, we can get advantages of the group-based proposal where nodes belong to a group and a leader node exists in each group.

Besides, in order to ensure an optimal use of storage space, we define two decision parameters about whether to store or discard a packet: time and distance. These parameters are used to decide whether to save or delete packets from storage space, which prevents the fact that events can remain stored indefinitely.

The time parameter implies that the source node has to add a timestamp to each event. This timestamp indicates the moment when it detected and/or generated a packet. It prevents events from remaining indefinitely stored in memory because they are removed from storage space when they expire. This happens when the time exceeds a threshold. The source node sets this threshold $Expiration$ according to a radius R and its speed as defined below:

$$\frac{R}{speed} \cdot 60 = Expiration_{min}. \quad (9)$$

This radius R depends on the type of event E and the speed. In most cases, information generated at a certain location in a VANET is not interesting out of a radius distance. For example, the event might be a free parking lot space in the centre of a city, which would not be interesting for a driver that is two miles away. However, if the event were about a traffic jam, then the information would be very helpful for the same driver. This means that the basic radio R must be greater or lesser depending on the type of event.

It is possible that nodes remain indefinitely inside the same radius so a time parameter is necessary to delete old packets. Besides the radio, the node speed also influences the total time that must be considered for event storage. This is due to the fact that a vehicle does not use the same period of time travelling on a highway as on a conventional road, as speed is completely different. Therefore, time of storage expiration for each event must depend both on radius and on speed. Figure 8 shows the scheme idea through an example of the radius-based idea.

Vehicles leaving the radius zone will drop the packet. However, those nodes that stay inside the radio will keep carrying and forwarding the packet to the vehicles they meet. The operating flow of the proposal scheme depends on the role of the nodes.

- (i) **Leader Node.** In the proposal, when a leader receives a packet, before making a decision, it validates the information. Figure 9 illustrates the routing flow algorithm for validation. First, it checks whether its

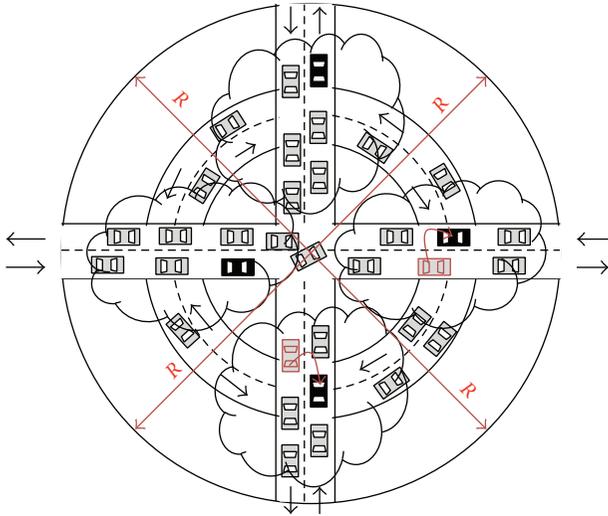


FIGURE 8: Event generation and action range.

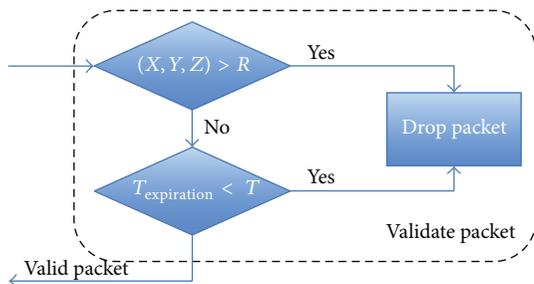


FIGURE 9: Checking event.

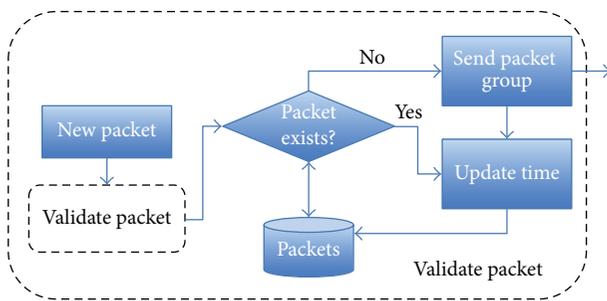


FIGURE 10: Leader flow when a packet is received.

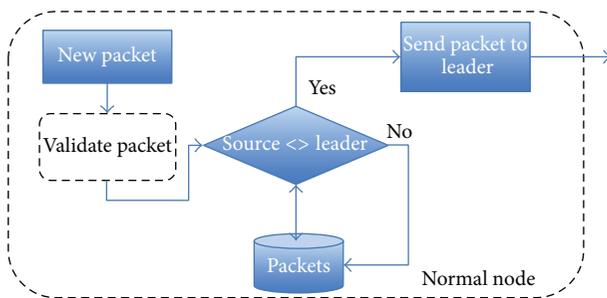


FIGURE 11: Node flow when a packet is received.

position is inside the radius R ; then, it checks the expiration time. If both alternatives are correct, the flow carries on the next step. Otherwise, it drops the message. In order to avoid unnecessary message forwarding, the leader has to check whether this information has just been sent to nodes in its group, and, in such a case, the leader does not store and relay the information. On the other hand, if the information is new or the timestamp has been updated, the leader will forward the message and store the new information. Besides, if the leader detects a new vehicle in its group, it sends all the updated information that is stored in the database (see Figure 10).

- (ii) *Normal Node*. A normal node has two operation modes. It could receive a packet from the leader or from a normal node. In the first case, it validates the information, checks the source node, and stores the information in its database. In order to minimize the broadcast overhead, in the second option, it validates the information. If the validation is right, it sends the packet to the leader, which will send the new packet to all nodes inside its group. Figure 11 illustrates this flow. Finally, when a node receives the information from the leader, it stores it in its database.

5. Conclusions

This paper has addressed the cooperation issue in fully distributed and decentralized VANETs. Several countermeasures to ensure cooperative behaviour have been proposed. In order to ensure that road traffic information available to drivers is trustful, a new cooperative scheme that allows generating true event warning packets has been described here. It was demonstrated that not only it provides truthful information but it also reduces the number of generated packets and hence the computational complexity. Besides, a new incentive scheme inspired by the lottery idea was proposed to encourage nodes to be relay information. Finally, different aspects that can lead to failure in cooperation, such as storage or battery consumption, have been taken into account when designing the proposal. We have evaluated the performance of cooperative groups when generating events and detecting misbehaviour, and the results confirm that this is a promising approach for increasing channel efficiency and decreasing false information generation in vehicular ad hoc networks.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the Spanish MINECO and the FEDER Fund under Projects TIN2011-25452 and IPT-2012-0585-370000.

References

- [1] P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil, "Design and implementation of an application for deploying vehicular networks with smartphones," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 834596, 10 pages, 2013.
- [2] P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil, "System for securely communicating in a spontaneous self-managed vehicular ad-hoc network," Patent P201000865, University of La Laguna, Tenerife, Spain, 2010.
- [3] L. Buttyán and J. Hubaux, "Stimulating cooperation in self-organizing mobile Ad Hoc networks," *Mobile Networks and Applications*, vol. 8, no. 5, pp. 579–592, 2003.
- [4] S. Lee, G. Pan, J. Park, M. Gerla, and S. Lu, "Secure incentives for commercial ad dissemination in vehicular networks," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 150–159, Montreal, Canada, September 2007.
- [5] L. Feng and W. Jie, "FRAME: an innovative incentive scheme in vehicular networks," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–6, June 2009.
- [6] N. W. Lo and H. C. Tsai, "A reputation system for traffic safety event on vehicular AD Hoc networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, Article ID 125348, 2009.
- [7] R. K. Schmidt, T. Leinmüller, E. Schoch, A. Held, and G. Schäfer, "Vehicle behavior analysis to enhance security in VANETs," in *Proceedings of the 4th Workshop on Vehicle to Vehicle Communications (V2VCOM '08)*, Eindhoven, The Netherlands, 2008.
- [8] F. Dotzer, L. Fischer, and P. Magiera, "VARS: a vehicle Ad-Hoc network reputation system," in *Proceedings of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pp. 454–456, 2005.
- [9] Z. Wang and C. Chigan, "Cooperation enhancement for message transmission in VANETs," *Wireless Personal Communications*, vol. 43, no. 1, pp. 141–156, 2007.
- [10] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J. P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1557–1568, 2007.
- [11] J. Sun and Y. Fang, "A defense technique against misbehavior in VANETs based on threshold authentication," in *Proceedings of the IEEE Military Communications Conference (MILCOM '08)*, pp. 1–7, San Diego, Calif, USA, November 2008.
- [12] C. Frese, J. Beyerer, and P. Zimmer, "Cooperation of cars and formation of cooperative groups," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '07)*, pp. 227–232, Istanbul, Turkey, June 2007.
- [13] T. Batz, K. Watson, and J. Beyerer, "Recognition of dangerous situations within a cooperative group of vehicles," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 907–912, June 2009.
- [14] A. Aikebaier, T. Enokido, and M. Takizawa, "Trustworthy group making algorithm in distributed systems," *Human-Centric Computing and Information Sciences*, vol. 1, no. 1, p. 15, 2011.
- [15] H. Mousannif, I. Khalil, and H. Al Moatassime, "Cooperation as a service in VANETs," *Journal of Universal Computer Science*, vol. 17, no. 8, pp. 1202–1218, 2011.
- [16] N. Xiong, A. V. Vasilakos, L. T. Yang, W. Pedrycz, Y. Zhang, and Y. Li, "A resilient and scalable flocking scheme in autonomous vehicular networks," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 126–136, 2010.
- [17] F. H. Tseng, L. D. Chou, and H. C. Chao, "A survey of black hole attacks in wireless mobile ad hoc networks," *Human-Centric Computing and Information Sciences*, vol. 1, no. 1, pp. 1–16, 2011.
- [18] M. Raya, A. Aziz, and J. Hubaux, "Efficient secure aggregation in VANETs," in *Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET '06)*, pp. 67–75, Los Angeles, Calif, USA, September 2006.
- [19] P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil, "How to build vehicular ad-hoc networks on smartphones," *Journal of Systems Architecture*, vol. 59, no. 10, pp. 996–1004, 2013.
- [20] J. M. Molina-Gil, P. Caballero-Gil, C. Hernández-Goya, and C. Caballero-Gil, "Data aggregation for information authentication in VANETs," in *Proceedings of the 6th International Conference on Information Assurance and Security (IAS '10)*, pp. 282–287, August 2010.
- [21] M. Jakobsson, J.-P. Hubaux, and L. Buttyan, "A micro-payment scheme encouraging collaboration in multi-hop cellular networks," in *Proceedings of the Financial Cryptography Conference*, pp. 15–33, 2003.
- [22] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in *Proceedings of the ACM International Conference on Mobile Systems, Applications and Services (MobiSys '10)*, pp. 285–297, San Francisco, Calif, USA, June 2010.

Research Article

A Regev-Type Fully Homomorphic Encryption Scheme Using Modulus Switching

Zhigang Chen,^{1,2,3} Jian Wang,¹ Liqun Chen,⁴ and Xinxia Song⁵

¹ College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

² College of Computer and Information, Zhejiang Wanli University, Ningbo, Zhejiang 315100, China

³ Information Security Group, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK

⁴ HP Labs, Bristol BS34 8QZ, UK

⁵ College of Junior, Zhejiang Wanli University, Ningbo, Zhejiang 315101, China

Correspondence should be addressed to Zhigang Chen; chzg99@gmail.com

Received 9 March 2014; Accepted 14 May 2014; Published 25 June 2014

Academic Editor: Tianjie Cao

Copyright © 2014 Zhigang Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A critical challenge in a fully homomorphic encryption (FHE) scheme is to manage noise. Modulus switching technique is currently the most efficient noise management technique. When using the modulus switching technique to design and implement a FHE scheme, how to choose concrete parameters is an important step, but to our best knowledge, this step has drawn very little attention to the existing FHE researches in the literature. The contributions of this paper are twofold. On one hand, we propose a function of the lower bound of dimension value in the switching techniques depending on the LWE specific security levels. On the other hand, as a case study, we modify the Brakerski FHE scheme (in Crypto 2012) by using the modulus switching technique. We recommend concrete parameter values of our proposed scheme and provide security analysis. Our result shows that the modified FHE scheme is more efficient than the original Brakerski scheme in the same security level.

1. Introduction

A fully homomorphic encryption (FHE) scheme allows arbitrary functions on certain data (referred to as plaintexts) to be performed via their ciphertexts (the encrypted version of the plaintexts) without decrypting the ciphertexts first; therefore, performing these functions does not require one to hold the secret decryption key corresponding to the encryption algorithm. This cryptographic primitive has shown a variety of attractive applications both in theory and in practice. A typical application example is to outsource a computational job to a mistrusted remote server without compromising data privacy.

Since Gentry constructed the first FHE scheme in 2009 [1], a number of FHE schemes including various optimizations of the Gentry original scheme have been proposed. Gentry and colleagues developed several FHE schemes with different improvement, for example, [2–6]; one of them is how to bootstrap “packed” ciphertexts [6]. Smart and

Vercauteren modified the Gentry scheme with the purpose of reducing the key and ciphertext sizes [7]. Stehlé and Steinfeld provides two improvements, respectively, on more aggressive analysis and probabilistic decryption algorithm in order to make the Gentry type of FHE schemes faster [8]. Brakerski et al. made a number of important contributions to this research field, such as [9–13], the details of which will be discussed more in the late part of this paper. Furthermore, van Dijk et al. proposed a new FHE construction over the integers [14], and Coron et al. further suggested on how to optimize this idea with shorter keys [15, 16]. López-Alt et al. constructed a multikey FHE scheme, which allows multiple ciphertexts under different keys to be decrypted jointly [17]. Alperin-Sheriff and Peikert introduced a method to achieve practical bootstrapping in Quasilinear time [18].

One critical challenge when constructing a FHE scheme is managing the noise growth in the process of homomorphic additions and multiplications. To our best knowledge, so far,

there exist three techniques to manage the noise growth as follows.

The first technique is bootstrapping that was used in the first FHE scheme introduced by Gentry. Bootstrapping means to evaluate its own decryption circuit homomorphically. One can use a bootstrapping process to get a new ciphertext after each homomorphic addition or homomorphic multiplication. The noise level in the new ciphertext is maintained in a fixed level. As long as this noise level permits, one can handle the next homomorphic addition or multiplication. By recursing this process a leveled FHE scheme can be developed, and the number of levels (although say the depth of the levels) for a computational circuit could be arbitrary with an assumption of circular security. A FHE scheme with the property of having an arbitrary depth of leveled circuits is referred to as a “pure” FHE scheme.

The second technique is modulus switching. This technique was developed by Brakerski and Vaikuntanathan in [10] and improved in [11]. The main idea of modulus switching is to scale down the ciphertext vector \mathbf{c} over \mathbb{Z}_q or a factor B after each multiplication, which results in a new ciphertext vector \mathbf{c}/B over $\mathbb{Z}_{q/B}$. A scaling process switches the first modulus q to the second modulus q/B and also reduces the noise E in the ciphertext vector \mathbf{c} to the new noise E/B in the new ciphertext vector \mathbf{c}/B . By following this process, the absolute magnitude of the new noise in the new ciphertext actually decreases. Modulus switching therefore can be used to manage noise at the cost of sacrificing the size of modulus. A leveled FHE scheme without bootstrapping can be achieved by modulus switching. In this technique, the depth of leveled computational circuits is prearranged before the computation starts. The depth is presented as a polynomial. For any prearranged polynomial denoted by L , one can evaluate circuits of depth L by carefully choosing the ladder of decreasing modulus.

The third technique is called Flatten, developed by Gentry et al. in [19]. It is designed for the case that an encryption key is presented as a vector and a ciphertext is presented as a matrix. It makes the coefficients of a vector or matrix small by using a flattening technique.

Among the three techniques for noise management, bootstrapping is a general technique that can be used to manage noise in any FHE scheme, but it is very costly! The technique of Flatten is only used in the case where ciphertexts are matrices and the secret keys are vectors. Modulus switching is a lightweight and very powerful way to manage noise and one can efficiently evaluate an arithmetic circuit with an arbitrary polynomial size without resorting to bootstrapping. In this paper, we will focus on modulus switching for noise management and consider the case that ciphertexts and the secret keys are both vectors.

In terms of noise growth, the noise grows from E to E^2 with every multiplication in most of the existing FHE schemes, where E denotes the noise magnitude in ciphertext. However, in the FHE scheme [12] by Brakerski in 2012 (we call it Bra12 for short), each homomorphic multiplication does not square the noise, and instead of the noise grows from E to $\text{poly}(n) \cdot E$ after each homomorphic multiplication. From

this point of view, it looks like that the Bra12 scheme is more efficient, but in fact it is not true. Since the Bra12 scheme makes use of bootstrapping to manage noise, it requires modulus q must be big in order to achieve the result that the scheme has a circuit with enough depth to evaluate its own decryption circuit, for example, $q = \tilde{O}(2^{n/2})$. The security of the scheme depends on the ratio q/B , where B is an initial magnitude of noise; therefore, B cannot be small. These reasons result in the secret key \mathbf{s} sampled uniformly from \mathbb{Z}_q^n rather than from the error distribution χ in the Bra12 scheme. In addition, the noise mainly depends on one norm of \mathbf{s} writing as $\|\mathbf{s}\|_1$ in homomorphic multiplication in the Bra12 scheme. In order to reduce the noise, the scheme uses binary decomposition of the secret key \mathbf{s} to reduce the norm $\|\mathbf{s}\|_1$. This means that a ciphertext \mathbf{c} under the key \mathbf{s} is converted into a new form of ciphertext, denoted by $\text{Powerof2}(\mathbf{c})$ under key $\text{BitDecomp}(\mathbf{s})$. Although the new form of the ciphertext and the secret key can effectively reduce the noise, it increases the dimension of ciphertext and the secret key. In particular, the dimension of the ciphertext and secret key can further blow up in homomorphic multiplication and key switching, which lead to a fatal result when evaluating deep circuits, since it may need too much memory to compute. This feature considerably affects efficiency in the Bra12 scheme.

In this paper, we use modulus switching and an additional technique to improve the efficiency of the Bra12 scheme. Our scheme has the following properties:

- (1) There is lower dimension of the ciphertext and the secret key in homomorphic multiplication and key switching than in the Bra12 scheme. The ciphertext for homomorphic multiplication is defined as $\lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \rfloor$ that corresponds to the secret key $\mathbf{s} \otimes \mathbf{s}$ in our scheme, while the ciphertext for homomorphic multiplication is defined as $\lfloor 2/q \cdot (\text{Powerof2}(\mathbf{c}_1) \otimes \text{Powerof2}(\mathbf{c}_2)) \rfloor$ that corresponds to the secret key $\text{BitDecomp}(\mathbf{s}) \otimes \text{BitDecomp}(\mathbf{s})$ in the Bra12 scheme.
- (2) The secret key \mathbf{s} is sampled from a Gaussian distribution χ in our scheme, which can enable us to get small coefficients of \mathbf{s} . In the Bra12 scheme, the secret key \mathbf{s} is sampled uniformly from \mathbb{Z}_q^n .
- (3) Our scheme uses modulus switching to manage noise, while the Bra12 scheme uses bootstrapping to manage noise.
- (4) In our scheme the initial modulus is that $q \approx 2^{n^\epsilon}$ for every $\epsilon < 1$, while in the Bra12 scheme the modulus is that $q \approx \tilde{O}(2^{n/2})$. The small modulus q makes our scheme considerably efficient.

For a FHE scheme using modulus switching, it is very important to choose a ladder of gradually decreasing moduli $\{q_i\}$. However, so far there has not been a concrete method to tell how to choose these parameters in terms of a certain security level, even in the BGV scheme [11] that just provided a general method to choose moduli $\{q_i\}$. In this paper, we provide a solution to this problem. We first derive a function between the lower bound on the dimension of the LWE problem and the security level. Then we can choose every

concrete modulus q_i and other parameters for a certain security level (e.g., the security level is 80 bit) according to this function.

The rest of this paper is organized as follows. Section 2 defines notational conventions, introduces the LWE assumption, and defines homomorphic encryption and its related terms. Section 3 introduces the Regev encryption scheme that our scheme is based on and defines invariant structure. There is a minor change in the Regev encryption scheme that we describe here. We sample the secret key from a Gauss distribution rather than sample uniformly from \mathbb{Z}_q^n in the Regev encryption scheme. Section 4 analyzes the homomorphic properties by the opinion of invariant structure and the noise growth in homomorphic addition and multiplication. Section 5 introduces key switching and modulus switching. Our FHE scheme based on the modified Regev encryption scheme is presented in Section 6. We analyze how to enable the correctness of our scheme in Section 7. The security and the parameters of our scheme are presented in Section 8. We conclude the paper with a performance comparison between our scheme and the Bar12 scheme in Section 9.

2. Preliminaries

2.1. Basic Notation. For an integer q , we define the set $\mathbb{Z}_q = (-q/2, q/2] \cap \mathbb{Z}$. For any $x \in \mathbb{Z}$, let $y = [x]_q$ denote the unique value $y \in (-q/2, q/2]$. We use $\lfloor x \rfloor$ to indicate rounding x to the nearest integer, and $\lfloor x \rfloor, \lceil x \rceil$ (for $x \geq 0$) to indicate rounding down or up. When q is not a power of two, we will use $\lceil \log q \rceil$ to denote $1 + \lfloor \log q \rfloor$.

We use $x \leftarrow \mathcal{D}$ to denote that x is a sample from a distribution \mathcal{D} . We define B -bounded distributions as ones whose magnitudes never exceed B .

The inner product of two vectors \mathbf{v}, \mathbf{u} of dimension n is denoted by $\langle \mathbf{v}, \mathbf{u} \rangle$, recalling that $\langle \mathbf{v}, \mathbf{u} \rangle = \mathbf{v}^T \cdot \mathbf{u}$. The tensor product of two vectors \mathbf{v}, \mathbf{u} of dimension n , denoted by $\mathbf{v} \otimes \mathbf{u}$, is the n^2 dimensional vector containing all elements of the form $\mathbf{v}[i]\mathbf{u}[j]$. Note that $\langle \mathbf{v} \otimes \mathbf{u}, \mathbf{x} \otimes \mathbf{y} \rangle = \langle \mathbf{v}, \mathbf{x} \rangle \cdot \langle \mathbf{u}, \mathbf{y} \rangle$.

A lattice is defined as the set of all integer combinations $\Lambda = \mathcal{L}(\mathbf{b}_1, \dots, \mathbf{b}_n) = \{ \sum_{i=1}^n x_i \mathbf{b}_i : x_i \in \mathbb{Z} \text{ for } 1 \leq i \leq n \}$ of n linearly independent vectors $\mathbf{b}_1, \dots, \mathbf{b}_n$ in \mathbb{R}^n . The set of vectors $\mathbf{b}_1, \dots, \mathbf{b}_n$ is called a basis for the lattice. A basis can be represented by the matrix $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n] \in \mathbb{R}^{n \times n}$. The determinant of a lattice is the absolute value of the determinant of the basis matrix $\det(\Lambda) = |\det(\mathbf{B})|$.

q -ary lattices are most important in lattice-based cryptography. Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for integers q, m, n , there are two kinds of m -dimensional q -ary lattices

$$\begin{aligned} \Lambda_q(\mathbf{A}) &= \{ \mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = \mathbf{A}^T \mathbf{s} \text{ mod } q \text{ for some } \mathbf{s} \in \mathbb{Z}^n \}, \\ \Lambda_q^\perp(\mathbf{A}) &= \{ \mathbf{y} \in \mathbb{Z}^m : \mathbf{A}^T \mathbf{s} = 0 \text{ mod } q \}. \end{aligned} \tag{1}$$

The two kinds of q -ary lattices are dual to each other, namely, $\Lambda_q(\mathbf{A}) = q \cdot \Lambda_q^\perp(\mathbf{A})^*$ and $\Lambda_q^\perp(\mathbf{A}) = q \cdot \Lambda_q(\mathbf{A})^*$.

2.2. Learning with Errors (LWE). The learning with errors (LWE) problem was introduced by Regev [20]. This problem

was later generalized as the ring learning with errors (RLWE) problem by Lyubashevsky et al. [21]. For security parameter λ , let $n = n(\lambda)$ be an integer dimension, let $q = q(\lambda) \geq 2$ be an integer, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and let $\chi = \chi(\lambda)$ be a distribution over \mathbb{Z} . Let $\mathcal{A}_{\mathbf{s}, \chi}$ be the distribution obtained by choosing a vector \mathbf{a} from \mathbb{Z}_q^n uniformly at random and a noise term $e \leftarrow \chi$, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The LWE problem includes the search-LWE problem and the decision-LWE problem. The search-LWE problem is giving an arbitrary number of independent samples from $\mathcal{A}_{\mathbf{s}, \chi}$, output \mathbf{s} with a high probability. We are primarily interested in the decision-LWE (DLWE) problem for cryptographic applications. The DLWE problem is defined as follows.

Definition 1 (DLWE). For an integer $q = q(\lambda)$ and an error distribution $\chi = \chi(\lambda)$ over \mathbb{Z} , the decision-LWE problem, denoted by $\text{DLWE}_{n, q, \chi}$, is to distinguish the following two distributions: in the first distribution, one sample from $\mathcal{A}_{\mathbf{s}, \chi}$; in the second distribution, one sample uniformly from \mathbb{Z}_q^{n+1} . The $\text{DLWE}_{n, q, \chi}$ assumption is that solving $\text{DLWE}_{n, q, \chi}$ is computationally infeasible.

Two kinds of reductions are known, namely, the quantum reduction [20] and classical [22, 23] reduction, between $\text{DLWE}_{n, q, \chi}$ and approximating short vector problems in lattices. Particularly, a probability distribution χ is taken to be the Gaussian distribution, which is statistically indistinguishable from the B -bound distribution for an appropriate value B .

Note that the DLWE problem can be seen as a bound distance decoding problem in q -ary lattices. The second component of LWE instance can be seen as a perturbed lattice point in $\Lambda_q(\mathbf{A}^\dagger)$, to be decoded.

We now state the quantum reduction from worst-case lattice problems to the LWE problem introduced in [20].

Theorem 2. For any integer dimension n , prime integer $q = q(\lambda)$, and $B = B(\lambda) \geq 2n$, there is an efficiently samplable B -bound distribution such that if there exists an efficient (possibly quantum) algorithm that solves $\text{DLWE}_{n, q, \chi}$, then there is an efficient quantum algorithm for solving $\widetilde{O}(q \cdot n^{1.5} / B)$ -approximate worst-case SIVP and gapSVP.

There are other forms of q (see [24, 25]). In addition, if the vector \mathbf{s} is sampled from the distribution χ , then the LWE problem is still hard. We sample \mathbf{s} from the Gaussian distribution χ in our scheme.

2.3. Leveled Fully Homomorphic Encryption. A homomorphic encryption scheme $\text{HE} = (\text{Keygen}, \text{Enc}, \text{Dec}, \text{Eval})$ includes a quadruple of PPT algorithms. For the definition of full homomorphic encryption, readers can refer to these papers [1, 12].

At present, there are two types of fully homomorphic encryption schemes. One is leveled fully homomorphic encryption schemes, in which the parameters of a scheme depend on the depth of the circuits that the scheme can evaluate. In that case any circuit with a polynomial depth can be evaluated. The other is pure fully homomorphic

encryption schemes, which can be built from a leveled fully homomorphic encryption scheme with the assumption of circular security. A pure fully homomorphic encryption scheme can evaluate the circuit whose depth is not limited. The following definitions are taken from [12].

Definition 3 (*L-homomorphism*). A scheme HE is *L-homomorphic*, for $L = L(\lambda)$, if for any depth L arithmetic circuit f (over $\text{GF}(2)$) and any set of inputs, m_1, \dots, m_l , it holds that

$$\begin{aligned} & \Pr [\text{HE.Dec}_{sk}(\text{HE.Eval}_{evk}(f, c_1, \dots, c_l)) \neq f(m_1, \dots, m_l)] \\ & = \text{negl}(\lambda), \end{aligned} \quad (2)$$

where $(pk, evk, sk) \leftarrow \text{HE.Keygen}(1^\lambda)$ and $c_i \leftarrow \text{HE.Enc}_{pk}(m_i)$.

Definition 4 (*compactness, full homomorphism, and leveled full homomorphism*). A homomorphic scheme is *compact* if its decryption circuit is independent of the evaluated function. A compact scheme is *fully homomorphic* if it is *L-homomorphic* for any polynomial L . The scheme is *leveled fully homomorphic* if it takes 1^L as additional input in key generation.

3. The Basic Encryption Scheme

As same as the Bra12 scheme, our scheme is based on Regev's encryption scheme [20]. We now describe the Regev encryption scheme, but we sample the secret key \mathbf{s} from a Gauss distribution while it was sampled uniformly from \mathbb{Z}_q^n in the Regev encryption scheme. This modification allows us to achieve our goal that the error distribution χ can be set to be as small as possible in our scheme. We call this modified Regev encryption scheme the basic encryption scheme.

Let $n = n(\lambda)$ be the dimension of lattice, an odd modulus $q = q(\lambda)$, and an error distribution $\chi = \chi(\lambda)$. The basic encryption scheme is described as follows.

E.SecretKeygen(1^λ): sample $s' \leftarrow \chi^n$. Output $sk = \mathbf{s} \leftarrow (1, s')$.

E.PublicKeygen(\mathbf{s}): let $N \geq 2(n \log q)$. Sample $\mathbf{A}' \leftarrow \mathbb{Z}_q^{N \times n}$ and $\mathbf{e} \leftarrow \chi^N$. Compute $\mathbf{b} \leftarrow \mathbf{A}'s' + \mathbf{e}$. Set \mathbf{A} to be the $(n+1)$ -column matrix consisting of \mathbf{b} followed by the n columns of $-\mathbf{A}'$, namely $\mathbf{A} = [\mathbf{b} \mid -\mathbf{A}'] \in \mathbb{Z}_q^{N \times (n+1)}$. Note that $\mathbf{A} \cdot \mathbf{s} = \mathbf{e}$. Set the public key $\mathbf{pk} = \mathbf{A}$.

E.Enc(\mathbf{pk}, m): to encrypt a message $m \in \{0, 1\}$, set $\mathbf{m} \leftarrow (m, 0, \dots, 0) \in \{0, 1\}^{n+1}$, sample $\mathbf{r} \in \{0, 1\}^N$, and output $\mathbf{c} \leftarrow \lfloor q/2 \rfloor \cdot \mathbf{m} + \mathbf{A}^T \cdot \mathbf{r} \in \mathbb{Z}_q^{n+1}$.

E.Dec(\mathbf{sk}, \mathbf{c}): output $m \leftarrow \lfloor (2/q) \lfloor \langle \mathbf{c}, \mathbf{s} \rangle \rfloor_q \rfloor \bmod 2$.

The basic encryption scheme above is semantic security based on the hardness of the LWE problem. The proof of this statement follows the proof of security of the original Regev encryption scheme given in [20].

A FHE scheme needs to maintain an invariant structure in decryption that is composed of plaintext and noise. The scheme must keep the invariant structure in the process of homomorphic addition and homomorphic multiplication in order to achieve homomorphism. Next, we define the invariant structure in the above basic encryption scheme and explain the relationship between the correctness of decryption and the noise magnitude in ciphertext.

Lemma 5. Let $\mathbf{c} \in \mathbb{Z}_q^{n+1}$ and $\mathbf{s} \in \mathbb{Z}_q^{n+1}$ be two vectors such that

$$\langle \mathbf{c}, \mathbf{s} \rangle = \left\lfloor \frac{q}{2} \right\rfloor \cdot m + e \pmod{q}, \quad (3)$$

where $m \in \{0, 1\}$. If $|e| < \lfloor q/2 \rfloor / 2$, then we have $m \leftarrow \text{E.Dec}(\mathbf{s}, \mathbf{c})$.

Proof. By definition

$$\begin{aligned} \langle \mathbf{c}, \mathbf{s} \rangle &= \left\langle \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{m} + \mathbf{A}^T \cdot \mathbf{r}, \mathbf{s} \right\rangle \pmod{q} \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \mathbf{r}^T \cdot \mathbf{A} \cdot \mathbf{s} \pmod{q} \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \langle \mathbf{r}, \mathbf{e} \rangle \pmod{q} \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + e \pmod{q}. \end{aligned} \quad (4)$$

Since the coefficients of \mathbf{e} are taken from a Gaussian distribution χ , $e = \langle \mathbf{r}, \mathbf{e} \rangle$ is also subject to a Gaussian distribution according to the standard fact from the Gaussian distribution. The Claim 5.2 in [20] showed that $|e| < \lfloor q/2 \rfloor / 2$ with high probability. Consider an encryption of 0 now; it is closer to 0 than to $\lfloor q/2 \rfloor$ in this case and therefore the decryption is correct. The proof for an encryption of 1 is similar.

The term e is called the noise. $\lfloor q/2 \rfloor \cdot m + e \pmod{q}$ is called the invariant structure. The above Lemma 5 shows that the invariant structure will be hold as long as $|e| < \lfloor q/2 \rfloor / 2$, which can ensure the correctness of decryption. Note that it is very important to keep the invariant structure in ciphertexts generated in homomorphic evaluation. \square

4. Homomorphic Properties and Noise Analysis

We take the definition of homomorphic addition and homomorphic multiplication from the Bra12 scheme, but here we analyze the homomorphic properties of the above scheme by the approach of the invariant structure. Now we analyze the noise growth in the homomorphic addition and multiplication.

Let \mathbf{c}_1 and \mathbf{c}_2 be two ciphertexts under the same secret key \mathbf{s} for modulus q such that

$$\begin{aligned} \langle \mathbf{c}_1, \mathbf{s} \rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 + e_1 \pmod{q} = \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 + e_1 + k_1 q, \\ \langle \mathbf{c}_2, \mathbf{s} \rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_2 + e_2 \pmod{q} = \left\lfloor \frac{q}{2} \right\rfloor \cdot m_2 + e_2 + k_2 q, \end{aligned} \quad (5)$$

for some e_1 and e_2 .

4.1. *Homomorphic Addition.* Let $\mathbf{c}^{\text{add}} = \mathbf{c}_1 + \mathbf{c}_2$. If the invariant structure $\langle \mathbf{c}_1 + \mathbf{c}_2, \mathbf{s} \rangle = \lfloor q/2 \rfloor \cdot (m_1 + m_2) + e \pmod{q}$ can be held during the decryption of \mathbf{c}^{add} for some e , the decryption would be correct such that homomorphic addition is obtained.

By definition

$$\begin{aligned} \langle \mathbf{c}_1 + \mathbf{c}_2, \mathbf{s} \rangle &= \langle \mathbf{c}_1, \mathbf{s} \rangle + \langle \mathbf{c}_2, \mathbf{s} \rangle \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot (m_1 + m_2) \\ &\quad + e_1 + e_2 + k_1q + k_2q. \end{aligned} \tag{6}$$

Let $e^{\text{add}} = e_1 + e_2$. According to the Lemma 5, if $|e^{\text{add}}| < \lfloor q/2 \rfloor / 2$, then $m_1 + m_2 \leftarrow \mathbf{E.Dec}(\mathbf{sk}, \mathbf{c}_1 + \mathbf{c}_2)$. It also means that the invariant structure $\lfloor q/2 \rfloor \cdot (m_1 + m_2) + e_1 + e_2 \pmod{q}$ can be kept in the decryption of \mathbf{c}^{add} . We note that the noise term of output is the sum of input noises.

4.2. *Homomorphic Multiplication.* Multiplicative homomorphism cannot be straightforwardly achieved. We need to construct a form of the two input ciphertexts to represent the homomorphic multiplication such that we can get the product of the two plaintexts with respect to the input ciphertexts after decrypting the homomorphic multiplication. For this purpose, we now focus on the invariant structure in the process of decryption. If the invariant structure $\lfloor q/2 \rfloor \cdot (m_1 \cdot m_2) + e$ for some e is kept in the decryption of the homomorphic multiplication, we could achieve multiplicative homomorphism. Next, we describe how to achieve multiplicative homomorphism by the approach of the invariant structure.

Consider the multiplication of $\langle \mathbf{c}_1, \mathbf{s} \rangle$ and $\langle \mathbf{c}_2, \mathbf{s} \rangle$ now, we have:

$$\begin{aligned} \langle \mathbf{c}_1, \mathbf{s} \rangle \cdot \langle \mathbf{c}_2, \mathbf{s} \rangle &= \left(\left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 + e_1 + k_1q \right) \\ &\quad \times \left(\left\lfloor \frac{q}{2} \right\rfloor \cdot m_2 + e_2 + k_2q \right) \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot \left\lfloor \frac{q}{2} \right\rfloor \cdot (m_1 m_2) + \dots \end{aligned} \tag{7}$$

In order to keep the invariant structure $\lfloor q/2 \rfloor \cdot (m_1 \cdot m_2) + e$, we multiply the above equation by $2/q$:

$$\begin{aligned} \frac{2}{q} \cdot \langle \mathbf{c}_1, \mathbf{s} \rangle \cdot \langle \mathbf{c}_2, \mathbf{s} \rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 + m_1 e_2 + m_2 e_1 + 2(e_1 k_2 + k_1 e_2) \\ &\quad + q \cdot (m_1 k_2 + k_1 m_2 + 2k_1 k_2) - [q]_2 \cdot (m_1 k_2 + k_1 m_2) \\ &\quad + \frac{[q]_2}{q} \cdot (m_1 e_2 - m_2 e_1 - \left\lfloor \frac{q}{2} \right\rfloor \cdot (m_1 m_2)) + \frac{2}{q} \cdot e_1 e_2 \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 + e, \end{aligned} \tag{8}$$

where $e = m_1 e_2 + m_2 e_1 + 2(e_1 k_2 + k_1 e_2) + q \cdot (m_1 k_2 + k_1 m_2 + 2k_1 k_2) - [q]_2 \cdot (m_1 k_2 + k_1 m_2) + [q]_2 / q \cdot (m_1 e_2 - m_2 e_1 - \lfloor q/2 \rfloor \cdot (m_1 m_2)) + 2/q \cdot e_1 e_2$.

The invariant structure appears in (8). Since $2/q \cdot \langle \mathbf{c}_1, \mathbf{s} \rangle \cdot \langle \mathbf{c}_2, \mathbf{s} \rangle = \langle 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2), \mathbf{s} \otimes \mathbf{s} \rangle$, multiplicative homomorphism is achieved by tensoring the input ciphertext $2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2)$. We note the ciphertext is fraction. For the sake of simplicity, we round the ciphertext for multiplication to the nearest integer ciphertext $\lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \rfloor$, which will bring out an error $\mathbf{r} = \lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \rfloor - 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2)$. Thus we get

$$\begin{aligned} &\left\langle \left\lfloor \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \right\rfloor, \mathbf{s} \otimes \mathbf{s} \right\rangle \\ &= \left\langle \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) + \mathbf{r}, \mathbf{s} \otimes \mathbf{s} \right\rangle \\ &= \left\langle \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2), \mathbf{s} \otimes \mathbf{s} \right\rangle + \langle \mathbf{r}, \mathbf{s} \otimes \mathbf{s} \rangle. \end{aligned} \tag{9}$$

Plugging (8) into above equation, we have

$$\begin{aligned} &\left\langle \left\lfloor \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \right\rfloor, \mathbf{s} \otimes \mathbf{s} \right\rangle \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 + m_1 e_2 + m_2 e_1 + 2(e_1 k_2 + k_1 e_2) \\ &\quad - [q]_2 \cdot (m_1 k_2 + k_1 m_2) + \frac{[q]_2}{q} \\ &\quad \cdot (m_1 e_2 - m_2 e_1 - \left\lfloor \frac{q}{2} \right\rfloor \cdot (m_1 m_2)) + \frac{2}{q} \cdot e_1 e_2 \\ &\quad + \langle \mathbf{r}, \mathbf{s} \otimes \mathbf{s} \rangle \pmod{q} \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 + e_1^{\text{mult}} + e_2^{\text{mult}} \pmod{q}, \end{aligned} \tag{10}$$

where $e_1^{\text{mult}} = m_1 e_2 + m_2 e_1 + 2(e_1 k_2 + k_1 e_2) - [q]_2 \cdot (m_1 k_2 + k_1 m_2) + [q]_2 / q \cdot (m_1 e_2 - m_2 e_1 - \lfloor q/2 \rfloor \cdot (m_1 m_2)) + 2/q \cdot e_1 e_2$ and $e_2^{\text{mult}} = |\langle \mathbf{r}, \mathbf{s} \otimes \mathbf{s} \rangle|$. The noise is $e_1^{\text{mult}} + e_2^{\text{mult}}$ in the ciphertext for multiplication. Particularly, the significant noise term of e_1^{mult} is $2(e_1 k_2 + k_1 e_2)$, which is not like the many previous FHE schemes whose homomorphic multiplication operation squares the noise.

The ciphertext for multiplication can thus be defined as $\lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \rfloor$ that can be decrypted using a tensored secret key $\mathbf{s} \otimes \mathbf{s}$. The invariant structure in the decryption of the homomorphic multiplication is $\lfloor q/2 \rfloor \cdot m_1 m_2 + e_1^{\text{mult}} + e_2^{\text{mult}}$. If $|e_1^{\text{mult}} + e_2^{\text{mult}}| < \lfloor q/2 \rfloor / 2$, according to Lemma 5, the invariant structure can be kept such that the correctness of decryption can hold. So we have $m_1 m_2 \leftarrow \mathbf{E.Dec}(\mathbf{sk}, \lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \rfloor)$, where \mathbf{sk} is $\mathbf{s} \otimes \mathbf{s}$. So far, we have finished the construction for the ciphertext for multiplication. We have achieved homomorphic addition and homomorphic multiplication. However, the noise growth is caused in the homomorphic addition and homomorphic multiplication.

The problem of noise growth in the homomorphic evaluation affects directly the homomorphic ability of the above

basic encryption scheme, so it is critical to manage noise growth for constructing the FHE scheme. Before we solve the problem of noise growth, we in the next subsection analyze the noise growth in a homomorphic addition and homomorphic multiplication. Note that our analysis method for the noise growth is different from the one used in the Bral2 scheme, as the secret key \mathbf{s} is sampled from a Gaussian distribution which results in the secret key \mathbf{s} is B -bounded. In addition, we give a tighter noise analysis than it in [12].

4.3. Noise Analysis

Lemma 6. Let $q, n, |\chi| \leq B, N$ be parameters as described in the basic encryption scheme. Let $\mathbf{c}_1, \mathbf{c}_2$ be the ciphertexts under the secret key \mathbf{s} such that

$$\begin{aligned} \langle \mathbf{c}_1, \mathbf{s} \rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 + e_1 \pmod{q}, \\ \langle \mathbf{c}_2, \mathbf{s} \rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_2 + e_2 \pmod{q}, \end{aligned} \tag{11}$$

with $|e_1|, |e_2| \leq E < \lfloor q/2 \rfloor / 2$. Then

$$\begin{aligned} \langle \mathbf{c}_1 + \mathbf{c}_2, \mathbf{s} \rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot [m_1 + m_2]_2 + e^{add} \pmod{q}, \\ \left\langle \left\lfloor \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \right\rfloor, \mathbf{s} \otimes \mathbf{s} \right\rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 + e^{mult} \pmod{q}, \end{aligned} \tag{12}$$

where $|e^{add}| \leq 1 + 2E, |e^{mult}| \leq 12nBE$.

Proof

Analysis for Addition. By definition

$$\begin{aligned} \langle \mathbf{c}_1 + \mathbf{c}_2, \mathbf{s} \rangle &= \langle \mathbf{c}_1, \mathbf{s} \rangle + \langle \mathbf{c}_2, \mathbf{s} \rangle \pmod{q} \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 + e_1 + \left\lfloor \frac{q}{2} \right\rfloor \cdot m_2 + e_2 \pmod{q} \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot (m_1 + m_2) + e_1 + e_2 \pmod{q} \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot [m_1 + m_2]_2 + 2 \cdot \left\lfloor \frac{q}{2} \right\rfloor \cdot \left\lfloor \frac{m_1 + m_2}{2} \right\rfloor \\ &\quad + e_1 + e_2 \pmod{q}. \end{aligned} \tag{13}$$

Then we get $|e^{add}| = |2 \cdot \lfloor q/2 \rfloor \cdot \lfloor (m_1 + m_2)/2 \rfloor + e_1 + e_2| \leq 1 + 2 \cdot E$.

Analysis for Multiplication. By (10)

$$\begin{aligned} \left\langle \left\lfloor \frac{2}{q} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \right\rfloor, \mathbf{s} \otimes \mathbf{s} \right\rangle &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_1 m_2 \\ &\quad + e_1^{mult} + e_2^{mult} \pmod{q}. \end{aligned} \tag{14}$$

We first analyze the bound of e_1^{mult} . The magnitude of e_1^{mult} mainly depends on the term $2(e_1 k_2 + k_1 e_2)$, so we check

the bound of the absolute value of k_1 (the same bound also holds for k_2):

$$\begin{aligned} |k_1| &= \frac{|\langle \mathbf{c}_1, \mathbf{s} \rangle - \lfloor q/2 \rfloor \cdot m_1 - e_1|}{q} \\ &\leq \frac{|\langle \mathbf{c}_1, \mathbf{s} \rangle|}{q} + 1 \\ &\leq \left(\frac{\|\mathbf{c}_1\|_\infty}{q} \right) \cdot \|\mathbf{s}\|_1 + 1 \\ &\leq \left(\frac{1}{2} \right) \cdot \|\mathbf{s}\|_1 + 1 \\ &\leq nB. \end{aligned} \tag{15}$$

The absolute value of k_1 depends on $\|\mathbf{s}\|_1$ from above inequality, then the bound of k_1 is $O(E \cdot \|\mathbf{s}\|_1)$. The tighter bound is described as follows:

$$|e_1^{mult}| \leq 2E + 4nBE + 2nB + 3 \leq 8nBE. \tag{16}$$

Next, we analyze the bound of e_2^{mult} . According to the definition of an error $\mathbf{r} = \lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \rfloor - 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2)$ and the secret key sampled from a B -bounded Gaussian distribution, we get $\|\mathbf{r}\|_\infty \leq 1/2$ and $|\mathbf{s} \otimes \mathbf{s}| \leq (n+1)^2 B^2$. Then

$$\begin{aligned} |e_2^{mult}| &= |\langle \mathbf{r}, \mathbf{s} \otimes \mathbf{s} \rangle| \leq \|\mathbf{r}\|_\infty \cdot |\mathbf{s} \otimes \mathbf{s}| \\ &\leq (n+1)^2 B^2. \end{aligned} \tag{17}$$

Putting these together, we get

$$\begin{aligned} |e_1^{mult} + e_2^{mult}| &\leq 8nBE + (n+1)^2 B^2 \\ &\leq 8nBE + 4nBE \\ &\leq 12nBE. \end{aligned} \tag{18}$$

We see that the significant noise term in the homomorphic multiplication depends on $\|\mathbf{s}\|_1$ from Lemma 6, which also happens in the Bral2 scheme. In order to reduce the norm, the secret key \mathbf{s} is expressed in the form of binary, namely, $\text{BitDecomp}(\mathbf{s})$, then the ciphertext corresponding to the \mathbf{s} is expressed in $\text{Powerof2}(\mathbf{c})$. The side effect is to produce the ciphertext vector and the secret key vector of a high dimension. In particularly, the ciphertext is the form of $\lfloor 2/q \cdot (\text{Powerof2}(\mathbf{c}_1) \otimes \text{Powerof2}(\mathbf{c}_2)) \rfloor$ under the key $\text{BitDecomp}(\mathbf{s}_1) \otimes \text{BitDecomp}(\mathbf{s}_2)$ after homomorphic multiplication, which results in a large amount of computation that requires a large memory. The process cannot be practical. However, our scheme does not have this result. Since we sample the secret key from a Gaussian distribution that enables the coefficients of the secret key to be as small as possible, the secret key \mathbf{s} needs not to be expressed in the form of binary, so the ciphertext. That is the reason why it can improve performance.

Under the above definition of homomorphic addition and homomorphic multiplication, we can perform only

a bounded number of homomorphic operations (namely, a somewhat homomorphic encryption scheme), because the noise and the dimension grow as a result of performing homomorphic operations. Therefore, there are two problems that should be solved in order to achieve a FHE scheme based on the somewhat homomorphic encryption scheme.

First, we need to control the dimension of the ciphertext that increases from $n + 1$ to $(n + 1)^2$ after a homomorphic multiplication. We use the key switching technique to solve this problem.

Second, we need to manage the noise growth in homomorphic operations. We use modulus switching to solve this problem. \square

5. Key Switching and Modulus Switching

We describe the two techniques: key switching and modulus switching. Our notation is adopted from [11].

5.1. Key Switching. Key switching can transform a ciphertext \mathbf{c}_1 under a secret key \mathbf{s}_1 to a new ciphertext \mathbf{c}_2 under a secret key \mathbf{s}_2 , in which \mathbf{c}_1 and \mathbf{c}_2 encrypt the same message. If the dimension of \mathbf{c}_2 and \mathbf{s}_2 is lower than the dimension of \mathbf{c}_1 and \mathbf{s}_1 , the dimension of the key and ciphertext vectors is reduced by key switching.

Key switching consists of two procedures. The first procedure is denoted by **SwitchKeyGen**($\mathbf{s}_1, \mathbf{s}_2, n_1, n_2, q$), which takes as input the two secret key vectors, the respective dimension of these vectors, the corresponding modulus q , and outputs some auxiliary information τ that is a matrix. The second procedure is denoted by **SwitchKey**($\tau, \mathbf{c}_1, n_1, n_2, q$), which takes as input the auxiliary information τ , a ciphertext \mathbf{c}_1 , and its dimension n_1 , the dimension of the output ciphertext n_2 , and the modulus q , and outputs a new ciphertext \mathbf{c}_2 whose dimension is n_2 .

SwitchKeyGen($\mathbf{s}_1 \in \mathbb{Z}_q^{n_1}, \mathbf{s}_2 \in \mathbb{Z}_q^{n_2}$):

- (1) Run $\mathbf{A} \leftarrow \mathbf{E.PublicKeyGen}(\mathbf{s}_2)$ for $N = n_1 \cdot \lceil \log q \rceil$, namely, $\mathbf{A} = [\mathbf{b} \mid -\mathbf{A}']$.
- (2) Set $\mathbf{B} \leftarrow [(\text{Powerof2}(\mathbf{s}_1) + \mathbf{b}) - \mathbf{A}']$, which means to add the $\text{Powerof2}(\mathbf{s}_1) \in \mathbb{Z}_q^N$ to $-\mathbf{A}'$'s first column and add \mathbf{b} to $-\mathbf{A}'$'s second column. Output $\tau_{\mathbf{s}_1 \rightarrow \mathbf{s}_2} = \mathbf{B}$.

SwitchKey($\tau_{\mathbf{s}_1 \rightarrow \mathbf{s}_2}, \mathbf{c}_1, q$): output $\mathbf{c}_2 = \text{BitDecomp}(\mathbf{c}_1)^T \cdot \mathbf{B} \in \mathbb{Z}_q^{n_2}$.

Key switching is essentially the product of a high dimension vector and a high dimension matrix. Next, we describe the correctness of key switching; namely, the decryption of the new ciphertext can preserve correctness. The proof is based on the definition (see [11]).

Lemma 7. Let $\mathbf{s}_1, \mathbf{s}_2, q, \mathbf{A}, \mathbf{B} = \tau_{\mathbf{s}_1 \rightarrow \mathbf{s}_2}$ be parameters as described in **SwitchKeyGen** and have $\mathbf{A} \cdot \mathbf{s}_2 = \mathbf{e}_2 \in \mathbb{Z}_q^N$. Let $\mathbf{c}_1 \in \mathbb{Z}_q^N$ and $\mathbf{c}_2 \leftarrow \mathbf{SwitchKey}(\tau_{\mathbf{s}_1 \rightarrow \mathbf{s}_2}, \mathbf{c}_1)$. Then,

$$\langle \mathbf{c}_2, \mathbf{s}_2 \rangle = \langle \text{BitDecomp}(\mathbf{c}_1), \mathbf{e}_2 \rangle + \langle \mathbf{c}_1, \mathbf{s}_1 \rangle \pmod{q}. \tag{19}$$

5.2. Modulus Switching

Definition 8 (Scale). For integer vector \mathbf{x} and integers $q > p > m$, we define $\mathbf{x}' \leftarrow \mathbf{Scale}(\mathbf{x}, q, p, 2)$ to be the vector closest to $(q/p) \cdot \mathbf{x}$ that satisfies $\mathbf{x} = \mathbf{x}' \pmod{2}$.

The next lemma shows that it is possible to transform a ciphertext \mathbf{c} that encrypts m under key \mathbf{s} for modulus q into a ciphertext \mathbf{c}' that encrypts m under the same key \mathbf{s} for modulus p . Since our basic encryption scheme is different from the basic scheme in the BGV scheme [11], the proof of Lemma 9 is slightly different from the proof in [11].

Lemma 9. Let q, p be odd and $q > p > 2$. Let $\mathbf{c} \in \mathbb{Z}_q^{n+1}$ and $\mathbf{c}' \leftarrow \mathbf{Scale}(\mathbf{c}, q, p, 2)$. Then, for any $\mathbf{s} \in \mathbb{Z}_q^{n+1}$, if $\langle \mathbf{c}, \mathbf{s} \rangle = \lfloor q/2 \rfloor \cdot m + E \pmod{q}$ and $\langle \mathbf{c}', \mathbf{s} \rangle = \lfloor p/2 \rfloor \cdot m + E' \pmod{p}$, with $|E| < q/4 - (q/p) \cdot \|\mathbf{s}\|_1 - q/(2p)$, we have

$$\begin{aligned} |E'| &< \left(\frac{p}{q}\right) \cdot |E| + \|\mathbf{s}\|_1 + \frac{1}{2}, \\ \left\lfloor \frac{2}{p} \left(\langle \mathbf{c}', \mathbf{s} \rangle \pmod{p} \right) \right\rfloor &= \left\lfloor \frac{2}{q} \langle \mathbf{c}, \mathbf{s} \rangle \pmod{q} \right\rfloor \pmod{2}. \end{aligned} \tag{20}$$

Proof. By $\langle \mathbf{c}, \mathbf{s} \rangle = \lfloor q/2 \rfloor \cdot m + E \pmod{q}$, we have $\langle \mathbf{c}, \mathbf{s} \rangle - \lfloor q/2 \rfloor \cdot m - kq = E$ for some $k \in \mathbb{Z}$. For the same k , let $\langle \mathbf{c}', \mathbf{s} \rangle - \lfloor p/2 \rfloor \cdot m - kp$. Next we just prove $|\langle \mathbf{c}', \mathbf{s} \rangle - \lfloor p/2 \rfloor \cdot m - kp| < p/2$ in order to prove $\langle \mathbf{c}', \mathbf{s} \rangle - \lfloor p/2 \rfloor \cdot m - kp = E'$.

Since $\langle \mathbf{c}', \mathbf{s} \rangle = \langle (p/q) \cdot \mathbf{c}, \mathbf{s} \rangle + \langle \varepsilon, \mathbf{s} \rangle$, where $\|\varepsilon\|_\infty < 1$, we have

$$\begin{aligned} &\left| \langle \mathbf{c}', \mathbf{s} \rangle - \left\lfloor \frac{q}{2} \right\rfloor \cdot m - kp \right| \\ &= \left| \left\langle \left(\frac{p}{q}\right) \cdot \mathbf{c}, \mathbf{s} \right\rangle - \left\lfloor \frac{q}{2} \right\rfloor \cdot m - kp + \langle \varepsilon, \mathbf{s} \rangle \right| \\ &= \left| \left(\frac{p}{q}\right) \cdot \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \left(\frac{p}{q}\right) \cdot E \right. \\ &\quad \left. + kp - \left\lfloor \frac{p}{2} \right\rfloor \cdot m - kp + \langle \varepsilon, \mathbf{s} \rangle \right| \\ &= \left| \left\lfloor \frac{p}{2} \right\rfloor \cdot m + \left(\frac{1}{2} - \frac{p}{2q}\right) \cdot m + \left(\frac{p}{q}\right) \cdot E \right. \\ &\quad \left. - \left\lfloor \frac{p}{2} \right\rfloor \cdot m + \langle \varepsilon, \mathbf{s} \rangle \right| \\ &= \left| \left(\frac{1}{2} - \frac{p}{2q}\right) \cdot m + \left(\frac{p}{q}\right) \cdot E + \langle \varepsilon, \mathbf{s} \rangle \right| \\ &< \left(\frac{p}{q}\right) \cdot |E| + \|\mathbf{s}\|_1 + \frac{1}{2} \\ &< \frac{p}{4}. \end{aligned} \tag{21}$$

We thus have $\langle \mathbf{c}', \mathbf{s} \rangle - \lfloor p/2 \rfloor \cdot m - kp = E'$ and $|E'| < (p/q)|E| + \|\mathbf{s}\|_1 + 1/2$.

Since $\langle \mathbf{c}', \mathbf{s} \rangle \pmod{p} = \langle \mathbf{c}', \mathbf{s} \rangle - kp$ and $\mathbf{c} = \mathbf{c}' \pmod{2}$, we have $\langle \mathbf{c}', \mathbf{s} \rangle \pmod{p} = \langle \mathbf{c}', \mathbf{s} \rangle - kp = \langle \mathbf{c}, \mathbf{s} \rangle - kq \pmod{2} = \langle \mathbf{c}, \mathbf{s} \rangle \pmod{q} \pmod{2}$. By definition, $2p = 2q \pmod{2}$. Since pq and 2 are coprime, it follows that $2/q = 2/p \pmod{2}$. Modulo 2 ,

we have $(2/p) \cdot (\langle \mathbf{c}', \mathbf{s} \rangle \bmod p) = (2/q) \cdot (\langle \mathbf{c}, \mathbf{s} \rangle \bmod q) \pmod{2}$. We thus get $\lfloor (2/p) \cdot (\langle \mathbf{c}', \mathbf{s} \rangle \bmod p) \rfloor = \lfloor (2/q) \cdot (\langle \mathbf{c}, \mathbf{s} \rangle \bmod q) \rfloor \pmod{2}$. \square

The following corollary follows immediately from Lemma 9.

Corollary 10. *Let q and p be two odd moduli. Let \mathbf{c} be a ciphertext under the key \mathbf{s} for the modulus q , where $m \leftarrow \lfloor (2/q) \cdot (\langle \mathbf{c}, \mathbf{s} \rangle \bmod q) \rfloor \pmod{2}$. Suppose that \mathbf{s} is a completely short key, and assume that $|E| < q/4 - (q/p) \cdot \|\mathbf{s}\|_1 - q/(2p)$. Then we have $\mathbf{c}' \leftarrow \mathbf{Scale}(\mathbf{x}, q, p, 2)$, where \mathbf{c}' is a ciphertext that encrypts the same message m under the keys \mathbf{s} for the modulus p , namely, $m \leftarrow \lfloor (2/p) \cdot (\langle \mathbf{c}', \mathbf{s} \rangle \bmod p) \rfloor \pmod{2}$. The noise of the new ciphertext \mathbf{c}' has magnitude at most $(p/q) \cdot |E| + \|\mathbf{s}\|_1 + 1/2$.*

Since the noise magnitude in the ciphertext \mathbf{c} depends on the length of the key vector \mathbf{s} , we must make the length of the key vector \mathbf{s} short in order to use modulus switching to reduce the magnitude of the noise. For this purpose, we sample the key \mathbf{s} from Gaussian distribution that is set to be as small as possible.

6. A Regev-Type FHE Scheme Using Modulus Switching

Next, we use modulus switching to construct a Regev-type FHE. This scheme is a leveled FHE scheme, in which the i th level needs a modulus q_i . The parameters in our scheme includes a ladder of decreasing modulus $\{q_i\}$ ($i = L-1, \dots, 0$), where a parameter L indicates the depth that a circuit can be evaluated. It is very important to choose reasonable modulus from q_L to q_0 , and we will focus on the details on how to choose reasonable modulus in Section 8. Since the magnitude of q_L is related to the security parameter λ and different circuit depths result in different magnitude values of q_L , the performance of our scheme depends on the security parameter λ and the circuit depth L .

FHE.Setup(λ, L): input the security parameter λ and the circuit level L , output a ladder of decreasing modulus $\{q_i\}$ ($i = L-1, \dots, 0$), the noise distribution χ , and the dimension n . Note that χ and n are the same as in the previous basic encryption scheme.

FHE.KeyGen(n, i): For $i = L-1$ down to 0, do the following.

- (1) Run $\mathbf{s}_i \leftarrow \mathbf{E.SecretKeyGen}(1^n)$. Let $\mathbf{sk} = \{\mathbf{s}_i\}$.
- (2) Run $\mathbf{A}_i \leftarrow \mathbf{E.PublicKeyGen}(\mathbf{s}_i)$. Let $\mathbf{pk}_1 = \{\mathbf{A}_i\}$ ($i = L-1, \dots, 0$).
- (3) Set $\mathbf{s}'_i \leftarrow \mathbf{s}_i \otimes \mathbf{s}_i \in \mathbb{Z}_q^{(n+1)^2}$.
- (4) Run $\tau_{\mathbf{s}'_{i+1} \rightarrow \mathbf{s}_i} \leftarrow \mathbf{SwitchKeyGen}(\mathbf{s}'_{i+1}, \mathbf{s}_i)$. (Omit this step when $i = L-1$.) Let $\mathbf{pk}_2 = \{\tau_{\mathbf{s}'_{i+1} \rightarrow \mathbf{s}_i}\}$ ($i = L-1, \dots, 0$).

Then output $\mathbf{sk} = \{\mathbf{s}_i\}$ and $\mathbf{pk} = (\mathbf{pk}_1, \mathbf{pk}_2)$.

FHE.Enc(\mathbf{pk}, m): take a message $m \in \{0, 1\}$. Run $\mathbf{E.Enc}(\mathbf{A}_{L-1}, m)$.

FHE.Dec($\mathbf{sk}, \mathbf{c}_i$): assume that \mathbf{c}_i is a ciphertext under the secret key \mathbf{s}_i . Run $\mathbf{E.Dec}(\mathbf{sk}, \mathbf{c}_i)$.

FHE.Add($\mathbf{pk}, \mathbf{c}_1, \mathbf{c}_2$): input two ciphertexts $\mathbf{c}_1, \mathbf{c}_2$ under the same secret key \mathbf{s}_i . If the two secret keys are different, we can use **FHE.Refresh** to refresh the two ciphertexts to the new two ciphertexts under the same secret key. Then, output $\mathbf{c}_3 \leftarrow \mathbf{c}_1 + \mathbf{c}_2$.

FHE.Mult($\mathbf{pk}, \mathbf{c}_1, \mathbf{c}_2$): input two ciphertexts $\mathbf{c}_1, \mathbf{c}_2$ under the same secret key \mathbf{s}_i . If the two secret keys are different, we can use **FHE.Refresh** to make it so. Compute $\mathbf{c}_3 \leftarrow \lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \rfloor$, and the relative secret key is $\mathbf{s}'_i = \mathbf{s}_i \otimes \mathbf{s}_i$. Then, output $\mathbf{c}_4 \leftarrow \mathbf{FHE.Refresh}(\mathbf{c}_3, \tau_{\mathbf{s}'_{i+1} \rightarrow \mathbf{s}_i}, q_i, q_{i-1})$.

FHE.Refresh($\mathbf{c}, \tau_{\mathbf{s}'_i \rightarrow \mathbf{s}_{i-1}}, q_i, q_{i-1}$): input ciphertext \mathbf{c} under the secret key \mathbf{s}'_i for modulus q_i . $\tau_{\mathbf{s}'_i \rightarrow \mathbf{s}_{i-1}}$ is the auxiliary information for key switching. The current and next modulus are q_i and q_{i-1} . Do the following.

- (1) Key switching: compute $\mathbf{c}_1 \leftarrow \mathbf{SwitchKey}(\tau_{\mathbf{s}'_i \rightarrow \mathbf{s}_{i-1}}, \mathbf{c}, q_i)$, a ciphertext under the key \mathbf{s}_{i-1} for q_i .
- (2) Modulus switching: compute $\mathbf{c}_2 \leftarrow \mathbf{Scale}(\mathbf{c}_1, q_i, q_{i-1}, 2)$, a ciphertext under the key \mathbf{s}_{i-1} for q_{i-1} .

In order to enable the correctness of the above leveled FHE scheme, we must choose the correct parameters. Next, we describe how to enable the correctness of this scheme.

7. Correctness

The correctness of the above leveled FHE scheme comes from the correctness of each step in homomorphic operations, that is, each step in **FHE.Add** and **FHE.Mult**. If the noise magnitude in ciphertext is below $q_{i-1}/4$ or $q_i/4$ after each step in homomorphic operations, correct decryption is guaranteed.

7.1. The Initial Noise. The initial ciphertext is output by **FHE.Enc** that just invokes **E.Enc**.

Lemma 11. *Let q_{L-1}, n, N be the parameters associated with **FHE.Enc**. χ is a B -bounded Gaussian distribution. The length of the noise in ciphertexts output by **FHE.Enc** is at most NB . If $NB < \lfloor q_{L-1}/2 \rfloor / 2$, correct decryption is guaranteed.*

Proof. **E.Enc**(\mathbf{A}_{L-1}, m) shows that $\lfloor q/2 \rfloor \cdot m + \mathbf{A}_{L-1}^T \cdot \mathbf{r} \in \mathbb{Z}_{q_{L-1}}^{(n+1)}$, where $\mathbf{r} \in \{0, 1\}^N$. We have $\mathbf{A}_{L-1} \cdot \mathbf{s}_{L-1} = \mathbf{e}$, where $\mathbf{e} \leftarrow \chi^N$ and $\mathbf{s}_{L-1} \leftarrow \mathbf{FHE.KeyGen}$. Then we get

$$\begin{aligned} & \langle \mathbf{c}, \mathbf{s}_{L-1} \rangle \pmod{q_{L-1}} \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \mathbf{r}^T \mathbf{A}_{L-1} \mathbf{s}_{L-1} \pmod{q_{L-1}} \quad (22) \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + NB \pmod{q_{L-1}}. \end{aligned}$$

According to Lemma 5, if $NB < \lfloor q_{L-1}/2 \rfloor / 2$, then correct decryption is guaranteed. \square

7.2. The Correctness of Homomorphic Operations

Lemma 12. Let \mathbf{c}_1 and \mathbf{c}_2 be two ciphertexts under \mathbf{s}_i for q_i , where $\langle \mathbf{c}_1, \mathbf{s}_i \rangle = \lfloor q_i/2 \rfloor \cdot m_1 + e_1 \pmod{q_i}$ and $\langle \mathbf{c}_2, \mathbf{s}_i \rangle = \lfloor q_i/2 \rfloor \cdot m_2 + e_2 \pmod{q_i}$ with $|e_1|, |e_2| \leq E < \lfloor q_i/2 \rfloor/2$. Let $\mathbf{c}_3 = \mathbf{c}_1 + \mathbf{c}_2$. The noise magnitude of \mathbf{c}_3 is at most $2E$. If $2E < \lfloor q_i/2 \rfloor/2$, we have $m_1 + m_2 \leftarrow \text{FHE.Dec}(\mathbf{s}_i, \mathbf{c}_3)$; namely, \mathbf{c}_3 can be correctly decrypted.

Proof. The proof can be obtained easily from Lemma 6. \square

The procedure of **FHE.Mult** consists of three steps, namely, the multiplication, and then the key switching and modulus switching. Next, we analyze the correctness of each step.

Lemma 13. Let \mathbf{c}_1 and \mathbf{c}_2 be two ciphertexts under \mathbf{s}_i for q_i , where $\langle \mathbf{c}_1, \mathbf{s}_i \rangle = \lfloor q_i/2 \rfloor \cdot m_1 + e_1 \pmod{q_i}$ and $\langle \mathbf{c}_2, \mathbf{s}_i \rangle = \lfloor q_i/2 \rfloor \cdot m_2 + e_2 \pmod{q_i}$ with $|e_1|, |e_2| \leq E < \lfloor q_i/2 \rfloor/2$. Let $\mathbf{c}_3 = \lfloor 2/q \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \rfloor$, and let $\mathbf{s}'_i = \mathbf{s}_i \otimes \mathbf{s}_i$. The noise magnitude of \mathbf{c}_3 is at most $12nBE$. If $12nBE < \lfloor q_i/2 \rfloor/2$, we have $m_1 m_2 \leftarrow \text{FHE.Dec}(\mathbf{s}'_i, \mathbf{c}_3)$; that is, \mathbf{c}_3 can be correctly decrypted.

Proof. The proof can be obtained easily from Lemma 6. \square

We note that the noise after multiplication is $12nBE$ rather than E^2 like in many of the previous FHE schemes.

Lemma 14. Let \mathbf{c} be a ciphertext under $\mathbf{s}'_i = \mathbf{s}_i \otimes \mathbf{s}_i$ for q_i , where $\langle \mathbf{c}, \mathbf{s}'_i \rangle = \lfloor q_i/2 \rfloor \cdot m + e \pmod{q_i}$ with $|e| \leq E < \lfloor q_i/2 \rfloor/2$. Let $\mathbf{c}_1 \leftarrow \text{SwitchKey}(\tau_{\mathbf{s}'_i \rightarrow \mathbf{s}_{i-1}}, \mathbf{c}, q_i)$. The noise magnitude of \mathbf{c}_1 is at most $E + B(n+1)^2 \log q_i$. If $E + B(n+1)^2 \log q_i < \lfloor q_i/2 \rfloor/2$, we have $m \leftarrow \text{FHE.Dec}(\mathbf{s}_{i-1}, \mathbf{c}_1)$; namely, \mathbf{c}_1 can be correctly decrypted.

Proof. By Lemma 7

$$\begin{aligned} \langle \mathbf{c}_1, \mathbf{s}_{i-1} \rangle &= \langle \text{BitDecomp}(\mathbf{c}), \mathbf{e}_1 \rangle + \langle \mathbf{c}, \mathbf{s}'_i \rangle \pmod{q_i} \\ &= B(n+1)^2 \cdot \lceil \log q_i \rceil + \langle \mathbf{c}, \mathbf{s}'_i \rangle \pmod{q_i} \\ &= B(n+1)^2 \cdot \lceil \log q_i \rceil + E + \left\lfloor \frac{q_i}{2} \right\rfloor \cdot m \pmod{q_i}, \end{aligned} \tag{23}$$

where $\mathbf{e}_1 \leftarrow \chi^N$ and $N = (n+1)^2 \cdot \lceil \log q_i \rceil$. If $E + B(n+1)^2 \log q_i < \lfloor q_i/2 \rfloor/2$, we have $m \leftarrow \text{FHE.Dec}(\mathbf{s}_{i-1}, \mathbf{c}_1)$; namely, \mathbf{c}_1 can be correctly decrypted. \square

Lemma 15. Let \mathbf{c}_1 be a ciphertext of dimension $n+1$ under \mathbf{s}_{i-1} for q_i , where $\langle \mathbf{c}_1, \mathbf{s}_i \rangle = \lfloor q_i/2 \rfloor \cdot m_1 + e_1 \pmod{q_i}$ with $|e_1| \leq E < \lfloor q_i/2 \rfloor/2$. Let $\mathbf{c}_2 \leftarrow \text{Scale}(\mathbf{c}_1, q_i, q_{i-1}, 2)$. The noise magnitude of \mathbf{c}_2 is at most $(q_{i-1}/q_i) \cdot E + (n+1) \cdot B + 1/2$. If $(q_{i-1}/q_i) \cdot E + (n+1) \cdot B + 1/2 < \lfloor q_{i-1}/2 \rfloor/2$, we have $m \leftarrow \text{FHE.Dec}(\mathbf{s}_{i-1}, \mathbf{c}_2)$; that is, \mathbf{c}_2 can be correctly decrypted.

Proof. By Corollary 10

$$\begin{aligned} |E'| &< \left(\frac{q_{i-1}}{q_i} \right) \cdot |E| + \|\mathbf{s}_{i-1}\|_1 + \frac{1}{2} \\ &< \left(\frac{q_{i-1}}{q_i} \right) \cdot E + (n+1) \cdot B + \frac{1}{2}. \end{aligned} \tag{24}$$

If $(q_{i-1}/q_i) \cdot E + (n+1) \cdot B + 1/2 < \lfloor q_{i-1}/2 \rfloor/2$, we have $m \leftarrow \text{FHE.Dec}(\mathbf{s}_{i-1}, \mathbf{c}_2)$. \square

8. Security and Parameters Settings

For a FHE scheme using modulus switching, it is most important to set up a reasonable ladder of decreasing modulus. The size of modulus is related to the dimension n of the LWE problem and the circuit depth L . Furthermore, the underlying security parameter λ is related to the dimension n of the LWE problem. However, it does not provide the concrete connection between the underlying security parameter and the dimension of the LWE problem in Regev's paper, nor the concrete parameters setting on its encryption scheme. It also does not provide the concrete method to set a concrete ladder of decreasing modulus based on a concrete security level and other parameters in the BGV scheme, even though BGV scheme is the first FHE scheme using modulus switching.

In this section, we will analyze the function between the lower bound in the dimension n of the LWE problem and the security level. Then we will give the method how to set the concrete ladder of decreasing modulus based on a certain security level and other parameters in our scheme.

8.1. The Dimension of the LWE Problem and the Security Level.

In order to estimate the hardness of LWE for a concert set of parameters, we first consider the distinguishing attack LWE; namely, the adversary distinguishes (with some noticeable advantage) an LWE instance from uniformly random, which can result in that the semantic security of an LWE-based cryptosystem is to be broken with the same advantage. Given a point \mathbf{b} that is either LWE instance or uniformly random. In order to do this attack, the adversary needs to find a short nonzero integral vector \mathbf{v} such that $\mathbf{A}\mathbf{v} = 0 \pmod{q}$; namely, \mathbf{v} is a short vector in $\Lambda_q^\perp(\mathbf{A}^t)$. Since $\Lambda_q^\perp(\mathbf{A}^t) = q \cdot \Lambda_q(\mathbf{A}^t)^*$, we have $\mathbf{v} = q \cdot \mathbf{y}$, where \mathbf{y} is a short vector in the dual of the lattice $\Lambda_q^\perp(\mathbf{A}^t)$. Then the adversary tries to test whether the inner product $\langle \mathbf{v}, \mathbf{b} \rangle$ is close to zero modulo q . When \mathbf{b} is a uniformly random instance, the test accepts with the probability exactly $1/2$. When $\mathbf{b} = \mathbf{A}^t \mathbf{s} + \mathbf{e}$, where \mathbf{e} is sampled from a Gaussian distribution with standard deviation σ , we have $\langle \mathbf{v}, \mathbf{b} \rangle = \langle \mathbf{v}, \mathbf{A}^t \mathbf{s} \rangle + \langle \mathbf{v}, \mathbf{e} \rangle = \langle q \cdot \mathbf{y}, \mathbf{A}^t \mathbf{s} \rangle + \langle \mathbf{v}, \mathbf{e} \rangle = \langle \mathbf{v}, \mathbf{e} \rangle \pmod{q}$, which is essentially Gaussian with standard deviation $\|\mathbf{v}\| \cdot \sigma$. When $\|\mathbf{v}\| \cdot \sigma$ is not much larger than q , the adversary can distinguish the Gaussian from the uniform with advantage of being very close to $\exp(-\pi \cdot (\|\mathbf{v}\| \cdot \sigma/q)^2)$. In general, in order to do the distinguishing attack with high confidence, one needs $\|\mathbf{v}\| \leq q/(2\sigma)$, which need to reduce the basis well enough such that the shortest vector is of size roughly q/σ . We assume that the security depends on the ratio

q/σ . Furthermore, we assume that the adversary will spend all the attack running time doing lattice reduction according to the paper [26].

The key point is to compute inner product $\langle \mathbf{v}, \mathbf{e} \rangle$ modulo q for a enough short vector in the distinguishing attack described above, which do not use the secret \mathbf{s} of LWE sample. It means that the distinguishing attack still work whether the secret \mathbf{s} is sampled from a Gaussian distribution or uniform. Next, we analyze the relation between the dimension of LWE and the security level.

A short vector used in the distinguishing attack can be got from lattice reduction algorithm. From the analysis of lattice reduction algorithms by Gama and Nguyen [27], the Hermite factor is regarded as the dominant parameter in the runtime of the reduction and the quality of the reduced basis. A reduced basis $\mathbf{B}(\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\|, \dots, \leq \|\mathbf{b}_m\|)$ of an m -dimensional lattice Λ has the Hermite factor δ^m for $\delta \geq 1$ if $\|\mathbf{b}_1\| = \delta^m \cdot \det(\Lambda)^{1/m}$. The term δ is called a quality parameter. In addition, Lindner and Peikert perform the experiments in the paper [26], which predict the runtime required to achieve a given root-Hermite factor δ in random q -ary lattices arising from LWE. The result of their experiments show that the logarithm of the runtime should grow roughly linearly in $1/\log(\delta)$. In particular, for a random q -ary lattices arising from LWE, the time (in seconds) that is spent to compute a reduced basis of quality δ is conservatively estimated at least as follows:

$$\log(\text{time}) \geq \frac{1.8}{\log(\delta)} - 110. \quad (25)$$

We note that the runtime estimated in (25) can be also applied in here to analyze our scheme. First, the random q -ary lattices for experiments in the paper [26] include the random q -ary lattices arising from LWE where the secret was sampled from a Gaussian distribution. Second, the encryption scheme described in the paper [26] is also based on the same LWE problem like our scheme; namely, the secret is choose from a Gaussian distribution.

Recall that the basis is required to be reduced well enough such that the shortest vector is of size roughly q/σ in the distinguishing attack. Thus the adversary needs to reduce the basis enough so that $\|\mathbf{b}_1\| = q/\sigma$. Moreover, for a random q -ary lattice of rank n , the determinant is q^n with high probability. By the definition of quality parameter δ , a basis \mathbf{B} that has quality parameter δ has $\|\mathbf{b}_1\| = \delta^m \cdot \det(\Lambda)^{1/m} = \delta^m \cdot q^{n/m}$. From the result in paper [28], when lattice reduction algorithms is applied to $\Lambda^\perp(\mathbf{A}^t)$, the shortest vectors are produced when $m = (n \cdot \log q / \log \delta)^{1/2}$. For simplicity, we take $\sigma = 1$ such that $\|\mathbf{b}_1\| = q/\sigma = q$, then we have

$$\begin{aligned} \log q &= \log(\delta^m \cdot q^{n/m}) \\ &= m \cdot \log \delta + \left(\frac{n}{m}\right) \log q \\ &= 2 \left(\frac{n \cdot \log q}{\log \delta}\right)^{1/2}. \end{aligned} \quad (26)$$

We can solve for n and plug Equation (25) into it, then get $n = \log q \cdot (\log(\text{time}) + 110)/7.2$ which is a function between

n and q/σ (recall $q = q/\sigma$). In order to ensure the time that is spent to reduce the basis at least 10^k , we need to set n to be at least

$$n \geq \frac{(\log(q/\sigma) \cdot (k + 110))}{7.2}. \quad (27)$$

We thus obtain the relation between the dimension of LWE and the security level. If we want to get 80 bit security level we need to set $n \geq \log(q/\sigma) \cdot 26.4$, for 128 bit security level we need to set $n \geq \log(q/\sigma) \cdot 33.1$.

8.2. Setting Concrete Parameters. Based on our scheme, we first set a concrete ladder of decreasing modulus. For a certain security level, we recommend specific dimension and modulus values for a specific circuit level L .

8.2.1. The Upper Bound of Noise. In order to obtain a suitable modulus, we need to find a common upper bound of noise for each circuit level.

Assume that we have a common upper bound E on noise magnitude, which means that the noise magnitude is at most E for all ciphertexts in all levels. Let \mathbf{c}_1 and \mathbf{c}_2 be two ciphertexts at level i . The noise magnitude is at most $12nBE$ after multiplication by following Lemma 11. Then, we apply the key switching, and the noise magnitude is at most $12nBE + B(n+1)^2 \log q_i$ by following Lemma 12. Finally, we apply modulus switching, and the noise magnitude in this stage is at most

$$\left(\frac{q_{i-1}}{q_i}\right) \cdot (12nBE + B(n+1)^2 \log q_i) + (n+1) \cdot B + \frac{1}{2}. \quad (28)$$

According to our assumption, the above equation is less than E . The dominant term is $12nBE$; thus, we have

$$\left(\frac{q_{i-1}}{q_i}\right) \cdot 12nBE < \frac{E}{2}, \quad (29)$$

$$\left(\frac{q_{i-1}}{q_i}\right) \cdot (B(n+1)^2 \log q_i) + (n+1) \cdot B + \frac{1}{2} < \frac{E}{2}. \quad (30)$$

We get $q_{i-1}/q_i < 1/(24nB)$ from Inequality (29), and we plug it into Inequality (30); then we have

$$\begin{aligned} &\left(\frac{1}{24nB}\right) \cdot (B(n+1)^2 \log q_i) + (n+1) \cdot B + \frac{1}{2} \\ &\approx \left(\frac{1}{24}\right) \cdot (n+1) \cdot \log q_i + (n+1) \cdot B \\ &\approx 2(n+1) \cdot B \\ &< \frac{E}{2}. \end{aligned} \quad (31)$$

We thus set $E \approx 8(n+1) \cdot B$, which is the approximate common upper bound. We also get the ratio of q_{i-1} and q_i that is approximately $1/(24nB)$. Next we can set a concrete ladder of decreasing modulus.

TABLE 1: The parameters of our scheme.

L	n	$\log q_0$	$\log q_{L-1}$
10	9400	63	149
20	19100	67	714
30	29200	70	1092
40	39500	72	1446

8.2.2. *A Concrete Ladder of Decreasing Modulus.* We first consider the smallest modulus. At the level 0, the noise magnitude is at most $12nBE$ after multiplication. In order for the correction of decryption to occur, we need to ensure $12nBE < q_0/4$. We can take $q_0 \approx 48nBE \approx 384n^2B^2$, which is approximately the smallest modulus.

Since $q_{i-1}/q_i \approx 1/(24nB)$, we can derive q_1, \dots, q_{L-1} ; for example, $q_1 = 24nB q_0$, $q_2 = (24nB)^2 \cdot q_0, \dots, q_{L-1} = (24nB)^{L-1} \cdot q_0$. We thus obtain a concrete ladder of decreasing modulus.

8.2.3. *The Concrete Parameters of Our Scheme.* According to (27) and the largest modulus q_{L-1} , we have $n \geq (\log(q_{L-1}/\sigma) \cdot (k + 110))/7.2$, which is the lower bound of the dimension of LWE. We use the Gaussian parameter $\sigma = 7$ from the experiment in [28]. Since $\log(q_{L-1}/\sigma) = \log((384 \cdot 24^{L-1} \cdot (nB)^{L+1})/7) \approx 8 + 5(L - 1) + (L + 1)(\log n + \log B)$, we have

$$n \geq \frac{(3 + 5L + (L + 1)(\log n + \log B)) \cdot (k + 110)}{7.2}. \quad (32)$$

B is the bound of Gaussian, and we use $B = 2n$ from the statement in [20]. We then can obtain the lower bound of the dimension n from the circuit depth L as well as the security level.

For an 80-bit security level ($k = 80$) and different circuit depth L , we derive the parameters of our scheme, as shown in the Table 1.

8.3. *Performance.* The computational complexity of our scheme comes from homomorphic multiplication which includes three steps. The computational cost that computes the tensored ciphertext is $\tilde{O}(n^2 \log^2 q_j)$. The computational cost in the step of key switching is $\tilde{O}(n^3 \log^2 q_j)$. The computational cost in the step of modulus switching is $\tilde{O}(n \log q_j)$. As a result, the per-gate computation in our scheme is $\tilde{O}(n^3 \log^2 q_j) = \tilde{O}(n^3 L^2)$. As a comparison, in the Bra12 scheme the per-gate computation is $\tilde{O}(n^3 \log^4 q_j)$. This shows that our scheme is more efficient than the Bra12 scheme.

8.4. *Bootstrapping.* We also can use bootstrapping to achieve a leveled FHE scheme. Furthermore, by using bootstrapping, we can obtain a pure FHE scheme with an assumption of circular security. There is a detailed explanation about bootstrapping in paper [29].

In our scheme the depth of a decryption circuit is $O(\log n + \log \log q)$. We can regard the above leveled FHE scheme as a somewhat homomorphic encryption scheme. As

long as we set the depth of circuit $L > O(\log n + \log \log q)$, our scheme is bootstrappable.

9. Conclusions

We have constructed a leveled FHE scheme using modulus switching based on the Bra12 scheme, and our scheme improves the efficiency of the Bra12 scheme. The per-gate computation in our scheme is $\tilde{O}(n^3 \log^2 q_j) = \tilde{O}(n^3 L^2)$, while it is $\tilde{O}(n^3 \log^4 q_j)$ in the Bra12 scheme. Furthermore, we have derived a function of the lower bound in the dimension n of the LWE problem and the security parameter. For an 80-bit security level and several different depth parameters, we have shown the concrete values of the dimension n of the LWE problem and the modulus q in each level. These concrete values for different parameters are very important in the fully homomorphic scheme that leverages modulus switching technique for noise management, which cannot be solved before.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The first author would like to thank the Fund of Jiangsu Innovation Program for Graduate Education (no. CXLX12.0162), the Fundamental Research Funds for the Central Universities, Ningbo Natural Science Foundation (no. 2012A610067), and the Chinese National Scholarship Fund and also appreciate the benefit to this work from Projects in science and technique of Ningbo municipal (no. 2012B82003). The fourth author would like to thank Ningbo Natural Science Foundation (no. 2013A610071).

References

- [1] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09)*, pp. 169–178, ACM, Bethesda, Md, USA, June 2009.
- [2] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits," in *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS '11)*, pp. 107–109, IEEE Computer Society, October 2011.
- [3] C. Gentry and S. Halevi, "Implementing Gentry's fully-homomorphic encryption scheme," in *Advances in Cryptology—Eurocrypt 2011*, K. Paterson, Ed., pp. 129–148, Springer, Berlin, Germany, 2011.
- [4] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," in *Advances in Cryptology—Crypto 2012*, R. Safavi-Naini and R. Canetti, Eds., pp. 850–867, Springer, Berlin, Germany, 2012.
- [5] C. Gentry, S. Halevi, C. Peikert, and N. P. Smart, "Ring switching in BGV-style homomorphic encryption," in *Security*

- and *Cryptography for Networks*, I. Visconti and R. Prisco, Eds., pp. 19–37, Springer, Berlin, Germany, 2012.
- [6] C. Gentry, S. Halevi, and N. P. Smart, “Better bootstrapping in fully homomorphic encryption,” in *Public Key Cryptography—Pkc 2012*, M. Fischlin, J. Buchmann, and M. Manulis, Eds., pp. 1–16, Springer, Berlin, Germany, 2012.
- [7] N. P. Smart and F. Vercauteren, “Fully homomorphic encryption with relatively small key and ciphertext sizes,” in *Public Key Cryptography—Pkc 2010*, P. Nguyen and D. Pointcheval, Eds., pp. 420–443, Springer, Berlin, Germany, 2010.
- [8] D. Stehlé and R. Steinfeld, “Faster fully homomorphic encryption,” in *Advances in Cryptology—Asiacrypt 2010*, M. Abe, Ed., pp. 377–394, Springer, Berlin, Germany, 2010.
- [9] Z. Brakerski and V. Vaikuntanathan, “Fully homomorphic encryption from ring-LWE and security for key dependent messages,” in *Advances in Cryptology—Crypto 2011*, P. Rogaway, Ed., pp. 505–524, Springer, Berlin, Germany, 2011.
- [10] Z. Brakerski and V. Vaikuntanathan, “Efficient fully homomorphic encryption from (standard) LWE,” in *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS '11)*, pp. 97–106, IEEE Computer Society, October 2011.
- [11] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “(Leveled) fully homomorphic encryption without bootstrapping,” in *Proceedings of the 3rd Conference on Innovations in Theoretical Computer Science*, pp. 309–325, ACM, Cambridge, Mass, USA, January 2012.
- [12] Z. Brakerski, “Fully homomorphic encryption without modulus switching from classical GapSVP,” in *Advances in Cryptology—Crypto 2012*, R. Safavi-Naini and R. Canetti, Eds., pp. 868–886, Springer, Berlin, Germany, 2012.
- [13] Z. Brakerski, C. Gentry, and S. Halevi, “Packed ciphertexts in LWE-based homomorphic encryption,” in *Public-Key Cryptography—Pkc 2013*, K. Kurosawa and G. Hanaoka, Eds., pp. 1–13, Springer, Berlin, Germany, 2013.
- [14] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, “Fully Homomorphic Encryption over the Integers,” in *Advances in Cryptology—Eurocrypt 2010*, H. Gilbert, Ed., pp. 24–43, Springer, Berlin, Germany, 2010.
- [15] J. Coron, A. Mandal, D. Naccache, and M. Tibouchi, “Fully homomorphic encryption over the integers with shorter public keys,” in *Advances in Cryptology—Crypto 2011*, P. Rogaway, Ed., pp. 487–504, Springer, Berlin, Germany, 2011.
- [16] J. Coron, D. Naccache, and M. Tibouchi, “Public key compression and modulus switching for fully homomorphic encryption over the integers,” in *Advances in Cryptology—Eurocrypt 2012*, D. Pointcheval and T. Johansson, Eds., pp. 446–464, Springer, Berlin, Germany, 2012.
- [17] A. López-Alt, E. Tromer, and V. Vaikuntanathan, “On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption,” in *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pp. 1219–1234, ACM, New York, NY, USA, May 2012.
- [18] J. Alperin-Sheriff and C. Peikert, “Practical bootstrapping in quasilinear time,” in *Advances in Cryptology—Crypto 2013*, R. Canetti and J. Garay, Eds., pp. 1–20, Springer, Berlin, Germany, 2013.
- [19] C. Gentry, A. Sahai, and B. Waters, “Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based,” in *Advances in Cryptology—Crypto 2013*, R. Canetti and J. Garay, Eds., pp. 75–92, Springer, Berlin, Germany, 2013.
- [20] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pp. 84–93, ACM, Baltimore, Md, USA, November 2005.
- [21] V. Lyubashevsky, C. Peikert, and O. Regev, “On ideal lattices and learning with errors over rings,” in *Advances in Cryptology—Eurocrypt 2010*, H. Gilbert, Ed., pp. 1–23, Springer, Berlin, Germany, 2010.
- [22] C. Peikert, “Public-key cryptosystems from the worst-case shortest vector problem: extended abstract,” in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09)*, pp. 333–342, ACM, Bethesda, Md, USA, June 2009.
- [23] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, “Classical hardness of learning with errors,” in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pp. 575–584, ACM, June 2013.
- [24] D. Micciancio and P. Mol, “Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions,” in *Advances in Cryptology—Crypto 2011*, P. Rogaway, Ed., pp. 465–484, Springer, Berlin, Germany, 2011.
- [25] D. Micciancio and C. Peikert, “Trapdoors for lattices: simpler, tighter, faster, smaller,” in *Advances in Cryptology—Eurocrypt 2012*, D. Pointcheval and T. Johansson, Eds., pp. 700–718, Springer, Berlin, Germany, 2012.
- [26] R. Lindner and C. Peikert, “Better key sizes (and Attacks) for LWE-based encryption,” in *Topics in Cryptology—Ct-Rsa 2011*, A. Kiayias, Ed., pp. 319–339, Springer, Berlin, Germany, 2011.
- [27] N. Gama and P. Q. Nguyen, “Predicting lattice reduction,” in *Advances in Cryptology—Eurocrypt 2008*, N. Smart, Ed., pp. 31–51, Springer, Berlin, Germany, 2008.
- [28] D. Micciancio and O. Regev, “Lattice-based cryptography,” in *Post-Quantum Cryptography*, D. Bernstein, J. Buchmann, and E. Dahmen, Eds., pp. 147–191, Springer, Berlin, Germany, 2009.
- [29] C. Zhigang, W. Jian, C. Liquan, and S. Xinxia, “Review of how to construct a fully homomorphic encryption scheme,” *International Journal of Security and Its Applications*, vol. 8, no. 2, pp. 221–230, 2014.

Research Article

Automating Risk Analysis of Software Design Models

Maxime Frydman,¹ Guifré Ruiz,² Elisa Heymann,¹ Eduardo César,¹ and Barton P. Miller³

¹ Computer Architecture and Operating Systems Department, Universitat Autònoma de Barcelona, Campus UAB, Edifici Q, Bellaterra, 08193 Barcelona, Spain

² The Open Web Application Security Project (OWASP), 1200-C Agora Drive, No. 232, Bel Air, MD 21014, USA

³ Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706-1685, USA

Correspondence should be addressed to Maxime Frydman; mb.frydman@gmail.com

Received 14 March 2014; Accepted 20 May 2014; Published 18 June 2014

Academic Editor: Mirjana Ivanovic

Copyright © 2014 Maxime Frydman et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The growth of the internet and networked systems has exposed software to an increased amount of security threats. One of the responses from software developers to these threats is the introduction of security activities in the software development lifecycle. This paper describes an approach to reduce the need for costly human expertise to perform risk analysis in software, which is common in secure development methodologies, by automating threat modeling. Reducing the dependency on security experts aims at reducing the cost of secure development by allowing non-security-aware developers to apply secure development with little to no additional cost, making secure development more accessible. To automate threat modeling two data structures are introduced, identification trees and mitigation trees, to identify threats in software designs and advise mitigation techniques, while taking into account specification requirements and cost concerns. These are the components of our model for automated threat modeling, AutSEC. We validated AutSEC by implementing it in a tool based on data flow diagrams, from the Microsoft security development methodology, and applying it to VOMS, a grid middleware component, to evaluate our model's performance.

1. Introduction

Software supports the information structure of businesses and governments worldwide. The growth of the Internet and networked systems has implied an increase of threats and challenges for software development companies. To address this issue security activities are increasingly being introduced into the software development lifecycle to reduce the number of software defects earlier in the software cycle. Reducing software defects earlier in the software lifecycle offers two main advantages; first it lowers the cost of fixing the software and second it limits the risk of deploying insecure software to users.

There are currently three high-profile approaches to the development of secure software (detailed in Section 2), the OWASP comprehensive lightweight application security process (CLASP) [1], McGraw Touchpoints, [2] and the Microsoft security development lifecycle (SDL) [3].

All of these secure development methodologies share one essential risk analysis activity, called *threat modeling* [4], used

to guide the following steps of the process. In this activity, the architecture of the system is represented and analyzed, generally prior to the implementation, to identify potential security threats to the system and to select appropriate mitigation techniques to address them.

Unfortunately, this activity either must be performed by *security-aware developers* or requires a *core security team* as most developers are not used to thinking and acting as professional attackers [5], nor do they have the necessary security expertise to imagine sophisticated attack scenarios [6] and mitigation strategies. This need for security expertise adds a significant cost to secure software development which reduces the chance that it will be used in many software projects.

In this paper we address the problem of the security expertise required for risk analysis. We created a model, AutSEC (automated security expert consultant), that automates the risk analysis process. The purpose of AutSEC is to enforce *security by design*, where threats are mitigated early in the development process, and automate all security operations

of the threat modeling process to allow non-security-aware engineers to develop secure software.

To validate AutSEC we implemented the model in a tool that integrates with the Microsoft SDL methodology. This implementation is compatible with the Microsoft threat modeling process and tool, facilitating its integration in development environments where SDL is already deployed.

This paper makes the following contributions.

- (i) Two new data structures, *identification trees*, which contain information to identify threats in software design models, and *mitigation trees*, which classify threat countermeasures by their costs.
- (ii) A model, AutSEC, relying on these two data structures, that purges the less relevant threats according to the business policies and estimates the mitigation techniques of least effort that adhere to the software specification.

The rest of this paper is organized as follows. Section 2 describes current methodologies used for threat modeling. Section 3 describes the input expected by our tool in addition to the SDL standard. Section 4 describes the methodology of the model used to automate threat identification, sort risks, and compute least effort countermeasures. Section 5 presents experimental results obtained by applying our tool to the grid middleware component VOMS Admin. Section 6 analyses the experimental results. Finally Section 7 concludes our work.

2. Related Work

There are currently three widely deployed methodologies for secure application development. Each of these methodologies has the same purpose, that is, to detect and eliminate security threats to applications throughout the development lifecycle of the application. This activity begins during the architectural design of the application and ends after the application has been tested and deployed.

The OWASP *comprehensive lightweight application security process (CLASP)* is a set of processes that can be integrated into any software development process and is designed to be both easy to adopt and effective. This makes CLASP more suitable for small organizations. It takes a prescriptive approach, documenting activities that organizations should be doing, and provides an extensive wealth of security resources that make implementing those activities reasonable.

The *McGraw Touchpoints*, a methodology that involves explicitly pondering the security situation throughout the software lifecycle. This means identifying and understanding common risks, designing for security, and subjecting all software artifacts to thorough, objective risk analysis and testing. It is based on industrial experience gathered over many years.

The *Microsoft security development lifecycle (SDL)* is a software development security assurance process consisting of practices grouped in seven phases: training, requirements, design, implementation, verification, release, and response.

All three methodologies share a common activity called *threat modeling* where the software under development is modeled. This model is then used by security experts to identify potential threats to the software and how to best mitigate them. This is a crucial step in secure application development as it orients the security efforts that will be deployed throughout the applications development lifecycle.

Our proposal reduces the reliance on security experts by *automating the threat identification and mitigation step*. Our model was developed to be generic; however our implementation used to validate AutSEC is compatible with the Microsoft SDL methodology. This choice was made as the SDL methodology offers a modeling tool that meets the requirements of threat matching, while having the flexibility to add custom annotations used by our model to refine the analysis (further described in Section 4).

Our model relies on a knowledge base called *attack patterns* to perform threat identification. This knowledge base is composed of threats that AutSEC is capable of identifying. Each threat in our knowledge base is represented by an *identification tree*, a *mitigation tree*, and ranking information. The *identification tree* is used to identify potential threats based on the software model and is based on the work found in [7]. The *mitigation tree* represents all the possible countermeasures that can be used to address a threat. Mitigation trees are a new concept to list and rank possible countermeasure but its representation is based on concepts introduced by attack trees.

3. Software Design Modeling

There are several approaches used to represent software designs for security purposes [8]. As explained in Section 2, our implementation of the AutSEC model is aimed at automating the widely used threat modeling [9] process of the Microsoft security development lifecycle (SDL), which uses data flow diagrams (DFDs) to represent the software architecture. To perform the modeling, Microsoft provides analysts with a modeling tool [10]; our implementation is based on the output of this tool and only requires a few specific additions to the original diagrams.

Our implementation expects the system to be represented as defined in the threat modeling process, which consists of data flows, data stores, processes, and trust boundaries to build the DFDs [3]. In addition, it is expected of the developers to make three small additions to elements in the form of attributes.

- (i) *Asset value* represents the value as a resource, {high, medium, and low}, of a DFD element; for example, a server might be valued as high, while the log files might be valued as low. This is based on the potential damage that would result in the resource being compromised.
- (ii) *Languages* are programming language used, for example, Java or C++.
- (iii) *Frameworks* are frameworks or other external software libraries used, if any, for example, CSRF Guard or ESPAI.

The *asset value* attribute must be defined for each DFD element, but *Languages* and *Frameworks* must be defined only for processes. This information will be used to refine the results in the threat identification and risk sorting steps. Since our tool is implemented on top of the current threat modeling process, it is important to maintain compatibility with the current SDL tool. The addition of the new attributes is performed by utilizing the assumption feature of SDL and allows native integration.

An example of the required additions is shown in Figure 1. Figure 1(a) shows the original DFD modeled according to SDL. Figure 1(b) shows the same model as well as the new added attributes required by AutSEC. In this example, each DFD element of Figure 1(b) has now been assigned an asset value and the only process, VOMS server, has been assigned a language.

The *asset value* is determined based on the damage that can be done if the resource is compromised. The VOMS server was classified as a high value asset, the mail server was classified as a medium value asset, and the log files are classified as low value assets. In the case of VOMS, compromising the main server would allow an attacker to compromise the operations of VOMS while obtaining the log files would at best disclose certain private information.

4. Methodology

The aim of our model, AutSEC, is to automate the threat modeling activity so that non-security-aware developers can perform secure development. The model described in this section takes the diagrams produced by the developers during the requirements and design phase of their software and produces documentation that will identify threats and describe how to mitigate the threats throughout the software's lifecycle.

AutSEC is a 4-step process whose result is to generate three detailed reports, one for each relevant software development activity; these are the design, implementation, and verification reports.

The design report discusses architectural and design decisions that can mitigate or eliminate potential threats. The implementation report shows how to implement certain features in a secure manner. The verification report combines all the threats contained in the design and verification reports and details how to assert that each threat has been properly mitigated. These three reports reflect the stages of the development lifecycle.

Since our model is aimed at developers regardless of their security expertise, we have taken great care as to limit interaction with the developers. When our process requires inputs, the inputs take the form of specific questions that a developer is able to answer, that is, business requirements, implementation details, and general mechanics, about the software he is developing and in the form of a multiple choice or polar question (yes or no). For the same reason the documentation produced as output of our model is presented with all the necessary information to understand each threat and its mitigation technique.

The input to our model is the DFD produced by the threat modeling tool according to Section 3, and the outputs are the three detailed reports mentioned above. AutSEC is a 4-step process as shown in Figure 2.

- (1) *Data flow diagram canonicalization* to interpret the labels of user-defined elements of the diagram.
- (2) *Threat identification* to identify threats relevant to the diagram.
- (3) *Risk ranking and threat purging* to prioritize threats according to risk and dismiss threats depending on business requirements.
- (4) *Mitigation planning* to propose countermeasures for the discovered threats to the developers that are compatible with their requirements.

The combination of these four steps results in the threat evaluation of the user application. To perform the threat evaluation we use two knowledge bases.

C14n Table: C14n is the canonicalization table that contains the information used to map unknown user labels of the diagram to known values.

Attack Patterns: the attack patterns are a collection of information over each threat that contain the *identification tree* used to identify the threat, the *risk attributes* used to rank the threat, and the *mitigation tree* used to mitigate the threat.

The information concerning threats used to build the *attack patterns* was gathered from several relevant security sources and standards, such as *Common Attack Pattern Enumeration and Classification (CAPEC)* [11], *Common Weakness Enumeration (CWE)* [12], and *Open Web Security Project (OWASP)* [13] amongst others. These databases contain generic information that can be applied to any software as long as it is modelled with the methodology described in Section 3.

The following subsections describe each of AutSEC's 4 steps in detail.

4.1. Data Flow Diagram Canonicalization. The first step of AutSEC is the data flow diagram canonicalization; this serves to map unknown user-defined labels to ones that can be automatically interpreted, for example, the identification of a user defined entity called *Apache* as a *web server*. This is accomplished using a data structure called *MultiMap* that allows the mapping of a set of values to a single key to build a *canonicalization table*; see Figure 3.

The purpose of canonicalization is to obtain specific information about the elements contained in the diagram of the application. This increases the precision of the threat identification and reduces the amount of generic threats reported.

While this process performs relatively well, it is not possible to anticipate every declination that can be given to DFD elements. This is addressed by the questioning phase of AutSEC, where unmapped elements can be refined by the developers. This gives flexibility to the tool both in terms of

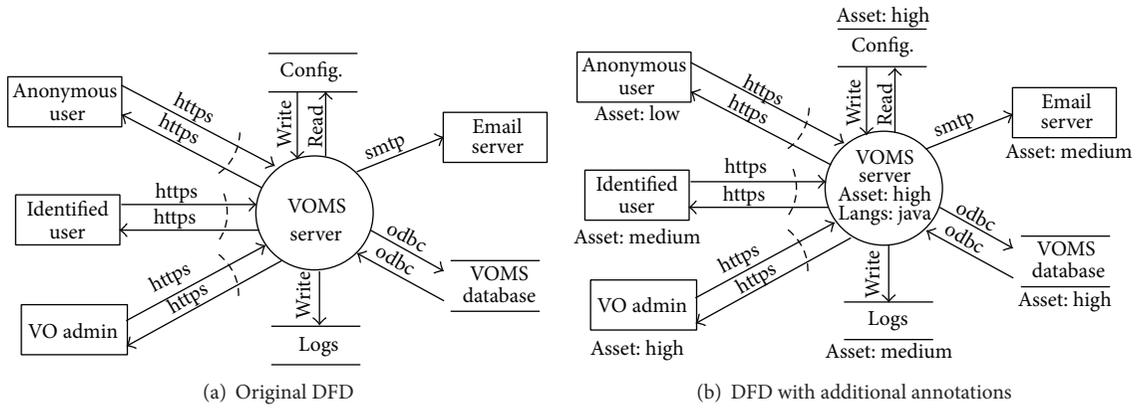


FIGURE 1: Original DFD compared to annotated DFD.

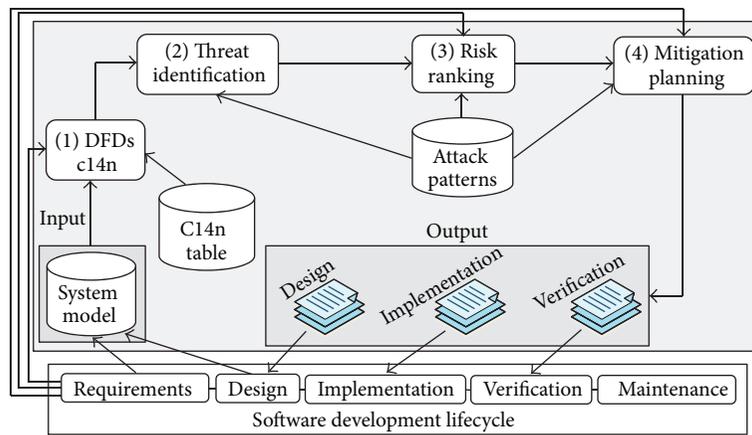


FIGURE 2: General architecture diagram of our approach.

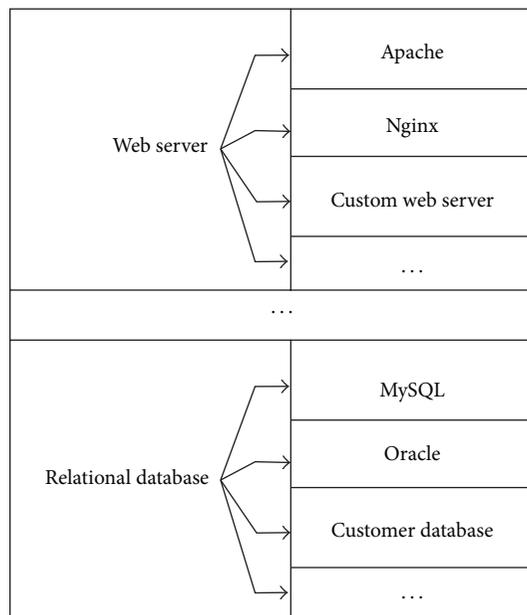


FIGURE 3: Example multimap mapping.

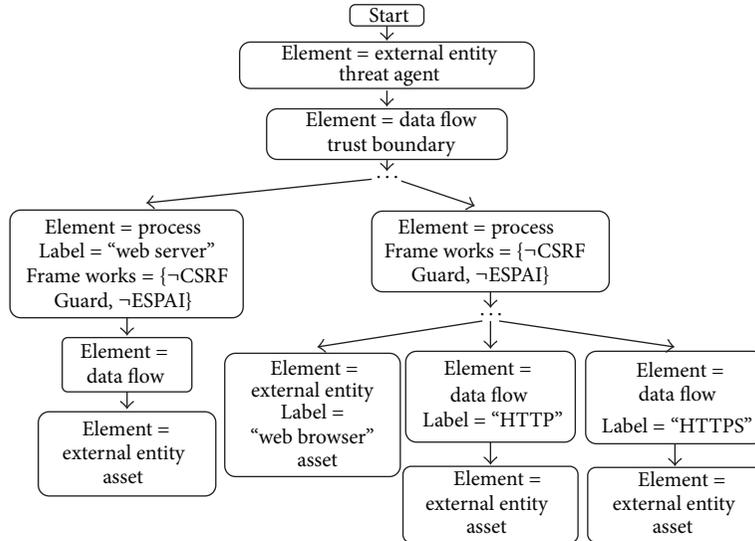


FIGURE 4: Identification graph of CSRF threats.

modeling restrictions as well as usability during the modeling process, only asking for refinements when interpretation has failed and learning from those refinements for further projects.

To interpret string attributes, each named element defined by the developer in the data flow diagram (DFD) is compared with the values contained in the MultiMap. If the mapping is successful then the label of the DFD element is replaced by the mapped key. Otherwise, a number of possible keys are presented to the developer for the unknown DFD element. This comes in the form of a list of generic items that are common in software development, for example, relational databases, web servers, and user interfaces. If one of these items is selected by the developer, the new value is added to the *canonicalization table* and its mapping key is assigned to the element of the DFD. If there were no suitable mappings, a generic value is assigned to the element and it will be treated as a generic element.

The resulting canonicalized DFD serves as input to the second step of the process.

4.2. Threat Identification. The second step of AutSEC is to perform threat identification based on the information contained in the canonicalized DFD.

This step is the core of the analysis; all further steps rely on the accuracy of the threat detection. To identify threats, a set of trees, *identification trees*, were designed. Each threat defined in the *attack patterns* contains an *identification tree* which is used to determine if the threat is relevant to the DFD.

Each branch of an *identification tree* represents a subgraph to be matched in the DFD. If the tree is matched with the DFD, it means that the threat is relevant. Each node in the tree represents an element of the DFD and can indicate additional attributes required for the match to be valid. These attributes can either be a requirement or indicate that a specific element cannot be present in the DFD for the match to be successful.

Figure 4 shows an example of *identification tree* for cross-site request forgery (CSRF) threats.

As shown in the figure, CSRF is a possible threat when there is data crossing a trust boundary (attack surface (the attack surface is the collection of interaction points with a software available to an attacker) [14]) to an HTTP server and the process that handles the HTTP request does not use specific frameworks against CSRF threats. If an anti-CSRF framework was used, it would indicate that the threat is mitigated making the threat irrelevant. This is represented by a key value pair, and the value has a “¬” symbol to indicate the negation in the match. In addition, each threat defines the *Threat Agent*, the component carrying out the attack, and the *asset*, the component compromised. In this case the threat agent is an external entity and the asset is a resource of value present on the web server. Certain threats can also require specific canonical labels; here the requirement for a successful CSRF attack is the presence of a web server.

The tree of Figure 4 is relatively generic for client-side threats of web technologies and can easily be reused for other types of threats.

Each threat identified in this step is added to a list of threats potentially affecting the software. This list serves as input to the next step of the process.

4.3. Risk Ranking and Threat Purging. The third step of AutSEC is to rank the list of identified threats by risk; this serves two purposes.

First purpose is the sorting of discovered threats to be able to prioritize the order in which they should be addressed. Second purpose is to allow the purging of threats; while certain potential threats can be present in a system, they might be considered too unlikely to occur or to have a very insignificant impact. For this purpose the user can set a threshold to eliminate certain threats based on business security policies.

The sorting is performed using the US National Security Telecommunications and Information Systems Security Committee [15] definition of risk in

$$\text{Risk} = \text{Likelihood} \times \text{Impact}. \quad (1)$$

The *likelihood* of exploitation value is taken from the CAPEC security source; its potential values are *very high* (1), *high* (0.75), *medium* (0.5), and *low* (0.25). The *impact* is calculated as shown in

$$\text{Impact} = \text{Asset} \times \text{ThreatAgent} \times \text{CIA Impact}. \quad (2)$$

The *asset* is defined by the developers, as explained in Section 3, and the *ThreatAgent* is the inverse of the *asset*. If a component is a high value asset, it will be a low value threat agent. For instance, if the *asset* is very high such as a database containing confidential information or an administrator of the system, it implies that it is a highly trusted component and the risk of suffering an attack from it is low. The possible values for the *asset* and *ThreatAgent* are *high* (1.2), *medium* (1.0), and *low* (0.8). Combining this information the *CIA impact* is computed as shown in

$$\text{CIA Impact} = \text{Conf Imp} + \text{Int Imp} + \text{Avai Imp}. \quad (3)$$

The *confidentiality*, *integrity*, and *availability* impact information are gathered from the attack patterns of CAPEC. Their value can be *high* (0.33), *medium* (0.22) or *low* (0.11). This puts risks in a range of approximately 0.05 and 1.45.

Using this ranking each threat identified in the previous step is ranked according to risk. The choice is then offered to the developers to elect a threshold; a threshold is a value from 0 to 1.5. Threats ranked below this threshold will be purged and will not be considered for the mitigation planning.

The purged list of threats contains only those that scored above the threshold; this is the input to the final step of the process.

4.4. Mitigation Planning. The final step of AutSEC is the production of the results that will be used by the software developers for secure development. The results take the form of reports that address each threat that was detected.

These reports are separated into three categories. The first addresses the design activity of the development and indicates architecture consideration to mitigate threats. The second addresses the implementation activity where specific guidelines are given to mitigate the threat. The third report concerns testing and serves as a fail-safe measure to ensure that all the detected threats were properly mitigated.

The architecture and implementation reports contain countermeasures for each identified threat. There are usually a number of possible countermeasures available to address a threat. Each of these countermeasures has implications, whether in terms of effort or as limitations to the software. For example, while removing all input to a software might be an efficient and effortless way to mitigate most threats, it is unlikely to conform with business requirements.

To address the election of the proper countermeasure, a new data structure was defined, called *mitigation tree*. The

purpose of *mitigation trees* is to determine the mitigation of least effort, that is, those that require the least monetary and/or time investment, to address the threat while conforming to the business requirements. If the countermeasure is not suitable, alternatives will be proposed that are more flexible but incur a higher effort.

Attack trees have been widely used by the community to represent attacks in a similar way as attack patterns do. Its root is the goal of an attacker, and each branch contains the set of actions that an attacker must carry out to achieve the goal at the root. *Mitigation trees* are similar however with a constructive rather than destructive intent. The root of the *mitigation tree* is the goal of mitigating a determined threat; each branch contains the set of software specifications or features, for the design and implementation activities, needed to accomplish the goal of the root. In addition, each feature contains an *estimated cost* associated to its implementation. This information is stored for each mitigation in the attack pattern of the threat.

Figure 5 shows the mitigation tree of CSRF attacks. It shows that to mitigate CSRF threats it is necessary to first mitigate all persistent cross-site scripting (PXSS) and reflected cross-site scripting (RXSS) threats and then offers four subbranches that represent different mitigation techniques. Each node or software specification of the attack tree has an estimated cost. This cost is calculated by using an *expert judgment* approach [16], where the security expert that builds the tree establishes a relative cost for each software specification using his past experiences as criteria.

During the mitigation planning, a set of polar questions are presented to the developers concerning design specifications, which are not shown in the DFD, and are relevant from a security point of view. These questions aim to identify whether certain threats are already mitigated and need not appear in the mitigation planning.

Since the purpose of this method is to emphasize *security by default* and *security by design*, it might be possible that certain countermeasures degrade the usability of the system, or that they are incompatible with the business requirements. When a certain countermeasure implies limitations, these are presented to the developers to ensure that they are acceptable. When a countermeasure is rejected, the *least effort* mitigation is recomputed excluding the incompatible mitigation. This process is repeated until a suitable countermeasure has been found for each threat.

While the verification and design reports indicate the measures to be taken to mitigate threats, this does not guarantee their proper implementation. This is addressed by the verification report, where each threat is set to be tested. If the threat is not successfully avoided during the elaboration of the architecture and the implementation, it will be detected when carrying out the penetration testing actions of this activity. The verification report contains for each threat the testing activities, some example exploit code, and relevant references.

Due to unclear boundaries between design and implementation [12], we define the boundary here as follows. If it can be modeled in UML, it corresponds to the design activity, otherwise to the implementation.

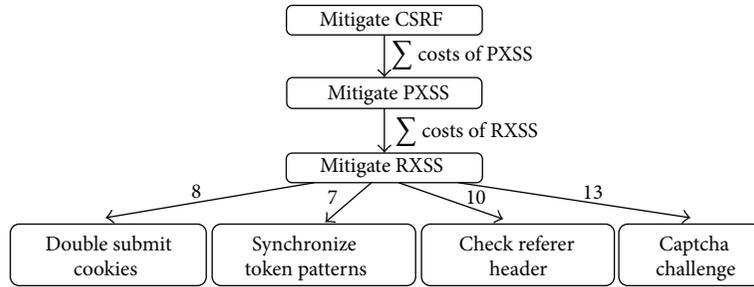


FIGURE 5: Mitigation tree of the CSRF attack pattern.

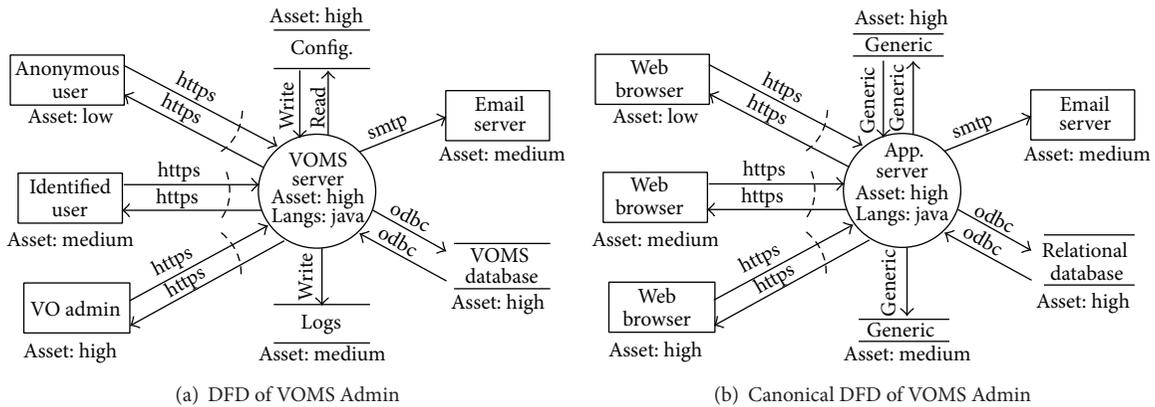


FIGURE 6: VOMS Admin parsing.

5. Experimental Results

To evaluate the validity of AutSEC’s approach, the distributed grid middleware component VOMS was used as case study using our tool that implements AutSEC. VOMS is a grid middleware that manages virtual organizations and user certificate attributes that will later be used by other middleware systems to take authorization decisions.

This section shows how the DFD diagram of VOMS Admin, a component of VOMS, is produced, how this diagram is processed, and the reports that result from this diagram. The examples provided in this section limit themselves to one element of VOMS and a specific threat, the 3 full reports containing all the threats and the DFD compatible with our tool are available in [17].

5.1. VOMS Admin DFD Diagram. Figure 6(a) shows the data flow diagram of VOMS Admin that was built as described in Section 3 using the threat modeling tool, which is the main manual phase of the assessment. After the diagram has been built, limited interaction is required to choose the desired level of security and the willing balance between security by default and usability.

5.2. VOMS Admin DFD Canonicalization. Figure 6(b) shows the canonicalized diagram produced from the original diagram. A few labels could not be automatically mapped, like

the VOMS server, configuration files, and logs, as they do not appear in the canonicalization mapping table. These were set by the developers during the polar question phase where the VOMS server was specified to be an app server and the other resources were assigned as generic entities.

It is from this canonicalized DFD that the threat identification will be performed.

5.3. VOMS Admin Threat Identification. During the threat analysis step, the subgraphs of attack patterns are matched with the canonical DFD of Figure 6(b) to find the potential threats to the system. Table 1 shows each detected vulnerability according to the DFD and the report in which it appears.

As can be seen, a wide range of potential vulnerabilities are detected that correspond to the DFD. The threats are separated between the design and implementation report, depending on where it is most appropriate to mitigate the issue. The verification report covers all detected threats to ensure that they have been properly addressed.

The cross-site request forgery (CSRF) threat is used in this document to provide a complete example of AutSEC’s process.

The detection of the CSRF threat results from the matching of the subgraphs shown in Listing 1 and represented in Figure 7(a).

As can be seen, a number of different types of potential CSRF attacks are identified for each type of web user, that

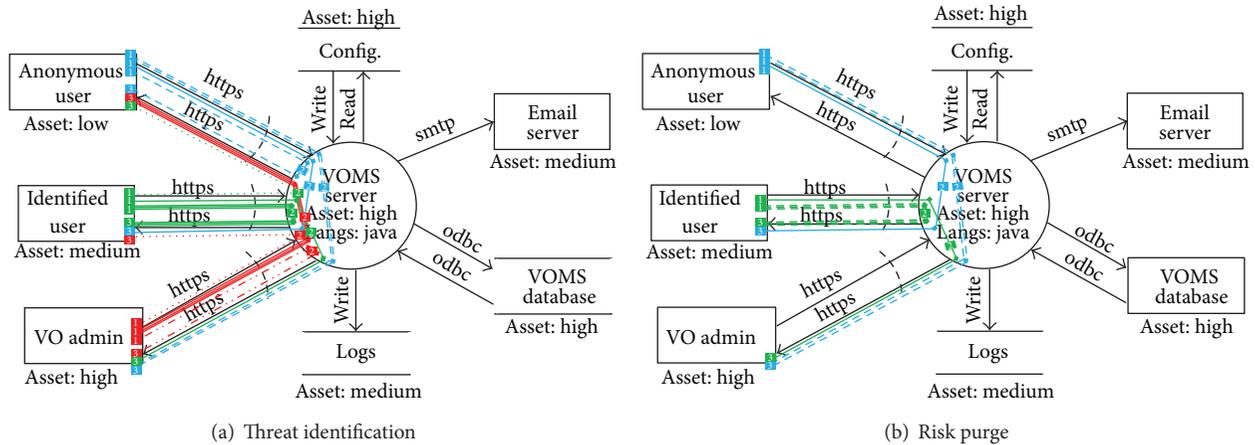


FIGURE 7: VOMS Admin risk analysis.

TABLE 1: Vulnerabilities reported to be found in corresponding AutSEC reports.

Design report	Implementation report	Verification report
Cross-site request forgery (CSRF)	Time of check to time of use	Cross-site request forgery (CSRF)
Insecure cryptographic storage	SQL injection attacks	Insecure cryptographic storage
	Reflected cross-site scripting (RXSS)	Time of check to time of use
	E-mail headers injection	SQL injection attacks
		Reflected cross-site scripting (RXSS)
		E-mail headers injection

is, anonymous, identified, and admin, can perform a CSRF attack that can target every type of user. These represent all the potential CSRF threats to the software.

Once all the potential threats are identified, the next step is to purge those that are least likely to have a significant impact.

5.4. VOMS Admin Risk Ranking. The risk ranking step of this process is where the threats detected in Listing 1 are sorted according to their potential threat. Listing 2 shows the results of the ranking using the CSRF example and is represented in Figure 7(b).

As can be seen, the most likely and damaging scenario for a CSRF attack is the one where an unprivileged user attacks an administrator while the least likely scenario is the one where an administrator attacks an anonymous user.

In Listing 3, the developers are asked for the threshold that is relevant to the activity. With a threshold of 0.5 only the most relevant CSRF attacks are kept discarding threats that score below the threshold.

5.5. VOMS Admin Mitigation Planning. Finally, before the final reports can be produced, the existing mitigation measures have to be identified and the countermeasures proposed by AutSEC have to be evaluated by the developers for compatibility with the requirements.

As described in Section 4.4, this is done using a set of polar questions regarding the design of the system to refine the results. This is shown in lines 1 to 3 of Listing 4. After these

questions have been answered, the countermeasures of *least effort* are estimated and software developers are asked if they are compliant with their software specification. If not, they are recomputed until a balance between security and usability is reached. This is shown in lines 5 to 15 of Listing 4.

This is the final step of the process and the reports are then generated containing the mitigations that have been approved.

It is interesting to notice that this approach has detected that to mitigate a CSRF threat it is first required to mitigate all PXSS and RXSS threats. For this reason, it is asked in line 9 of Listing 4 if it was possible to only allow alphanumeric characters in the HTML forms. Engineers answered “n” (no) because special characters are required in some fields. Therefore, mitigations were recomputed resulting in not only a more permissive but also a more expensive solution which is to HTML encode user supplied data before displaying it back to the web interface.

The mitigation choices are included in the final reports; these reports detail every threat detected that scored above the ranking threshold, the chosen mitigation technique that corresponds to the business requirements, and links that further describe the threat and their possible countermeasures.

6. Experimental Results Validation and Discussion

6.1. Validation. In order to validate our approach we compared the threats reported by our tool with the list of known

- (1) [CSRF] Path: {[VO Admin]-[App. server]-[VO Admin]}
- (2) [CSRF] Path: {[VO Admin]-[App. server]-[Ident. User]}
- (3) [CSRF] Path: {[VO Admin]-[App. server]-[Anon. User]}
- (4) [CSRF] Path: {[Anon. User]-[App. server]-[VO Admin]}
- (5) [CSRF] Path: {[Anon. User]-[App. server]-[Ident. User]}
- (6) [CSRF] Path: {[Anon. User]-[App. server]-[Anon. User]}
- (7) [CSRF] Path: {[Ident. User]-[App. server]-[VO Admin]}
- (8) [CSRF] Path: {[Ident. User]-[App. server]-[Ident. User]}
- (9) [CSRF] Path: {[Ident. User]-[App. server]-[Anon. User]}

LISTING 1: VOMS Admin CSRF threat identification.

- (4) [CSRF] Path: {[Anon. User]-[App. server]-[VO Admin]} Risk: [0.7128]
- (5) [CSRF] Path: {[Anon. User]-[App. server]-[Ident. User]} Risk: [0.6534]
- (7) [CSRF] Path: {[Ident. User]-[App. server]-[VO Admin]} Risk: [0.6534]
- (8) [CSRF] Path: {[Ident. User]-[App. server]-[Ident. User]} Risk: [0.5989]
- (6) [CSRF] Path: {[Anon. User]-[App. server]-[Anon. User]} Risk: [0.3564]
- (1) [CSRF] Path: {[VO Admin]-[App. server]-[VO Admin]} Risk: [0.3564]
- (9) [CSRF] Path: {[Ident. User]-[App. server]-[Anon. User]} Risk: [0.3267]
- (2) [CSRF] Path: {[VO Admin]-[App. server]-[Ident. User]} Risk: [0.3267]
- (3) [CSRF] Path: {[VO Admin]-[App. server]-[Anon. User]} Risk: [0.1782]

LISTING 2: CSRF risk sorting threats.

- What threshold do you want? [0.0-1.0] 0.5
- (4) [CSRF] Path: {[Anon. User]-[App. server]-[VO Admin]} Risk: [0.7128]
 - (5) [CSRF] Path: {[Anon. User]-[App. server]-[Ident. User]} Risk: [0.6534]
 - (7) [CSRF] Path: {[Ident. User]-[App. server]-[VO Admin]} Risk: [0.6534]
 - (8) [CSRF] Path: {[Ident. User]-[App. server]-[Ident. User]} Risk: [0.5989]

LISTING 3: Sorted CSRF risk purging threats.

- (1) (4.1) To refine results please answer a set of polar questions to refine identification in the DFDs:
- (2) *Are you Checking the Refer Header in the requests of all HTML form actions to determine if it originates from a trusted domain? [y/n] n
- (3) *Are you Synchronizing a Secret token pattern in all HTML form requests? [y/n] n
- (4)
- (5) (4.2) Pushed security by default, refining to give the desired usability
- (6) Answer [y/n] if it is OK to perform the following actions:
- (7) *Is it OK to Synchronizing a Secret token pattern in all HTML form requests? [y/n] y
- (8)
- (9) *Is it OK to Allow only alphanumeric characters in all fields of this payload? [y/n] n
- (10)
- (11) Computing best options...
- (12)
- (13) *Is it OK to HTML Encode all user supplied data before displaying it back to the web interface? [y/n] y
- (14)
- (15)
- (16) FINAL COUNTERMEASURES COMPUTED
- (17) (1) Checking Refer Header in the requests of all HTML form actions to determine if it comes from a trusted domain
- (18) (2) HTML Encode all user supplied data before displaying it back to the web interface
- (19)
- (20) # Reports available in out/report_design.pdf, out/report_implementation.pdf and out/report_verification.pdf

LISTING 4: VOMS Admin mitigation planning.

TABLE 2: VOMS vulnerability summary.

Vulnerability type	Count
Persistent cross-site scripting (PXSS)	2
Business specific	2
Cross-site scripting (XSS)	1
Cross-site request forgery (CSRF)	1
Denial of service (DoS)	1
Insecure third party library linking	1

VOMS vulnerabilities. The information on vulnerabilities affecting VOMS was gathered during the security audit carried out by our team, using the manual *First Principles Vulnerability Assessment* [18] methodology, as well as the collection of all the vulnerabilities reported by the community over the 2011–2014 period shared by the VOMS development team. All of the vulnerabilities discussed in this paper have been fixed and disclosed.

A summary of these vulnerabilities can be found in Table 2.

AutSEC was applied to VOMS a posteriori; that is, the process was applied to VOMS after its release and 10 years of activity. Although the main purpose of AutSEC is to be applied during the elaboration and development phases of software, it would be difficult to quantify the validity of our approach without perspective on the security issues that arise after software has been released for a length of time.

There are 3 crucial elements that affect the validity of this tool; they are the quantity of information contained in the *attack patterns*, the quality of the matching, and the accuracy of the ranking. The quantity of information contained in the *attack patterns* increases the security knowledge of the tool. The quality of the matching defines whether the knowledge contained in the attack patterns can be correctly put to use to identify threats. And the accuracy of the ranking is what allows prioritizing the focus of the security effort and discarding irrelevant threats. A tool which can only detect a single vulnerability but with 100% accuracy is of limited use, and so is one that detects every threat incorrectly. A useful security tool must strike a balance between those three factors.

Similarly to other automated security tools, it is of critical importance to present concise information to the developers that cover the widest array of significant risks, while limiting the amount of irrelevant information. This is traditionally called the ratio of false positives, threats that are reported but do not impact the system, to false negatives, threats that do affect the system but are not reported. The value of an automated tool to the developers is linked to the ratio of false positives to false negatives [19] as not reporting a potential vulnerability leads to a sense of false security, but reporting too many irrelevant vulnerabilities can be just as harmful as it conceals true threats.

In accordance with the definition of validity expressed above, this experiment was conducted using a vulnerability database that was not specific to VOMS but contained a variety of vulnerabilities from different programming languages and technologies. The following sections analyse

the results provided by our tool in terms of successfully identified threats, threats that were not identified (possible false negatives) and threats that were identified but have not been reported as affecting VOMS (possible false positives).

6.2. Successfully Detected Threats. From the identified potential vulnerabilities of VOMS, 4 real vulnerabilities match our tool's predictions. These are the 2 persistent cross-site scripting (PXSS), 1 cross-site scripting (XSS), and 1 cross-site request forgery (CSRF) vulnerabilities.

Analysis of these vulnerabilities shows that the mitigation techniques proposed by AutSEC would have successfully neutralized the threat and thus prevented the vulnerability. The verification fail-safe mechanism of AutSEC, in case the implementation is not correctly carried out, was also analyzed and it offered sufficient information to identify the vulnerabilities found in VOMS.

This result shows that our tool was successful in identifying the threat and offers useful mitigation techniques, and later in the application lifecycle it offers useful and relevant information to guarantee that the threat had been addressed. Early detection of vulnerabilities is the original purpose of AutSEC as it limits the financial impact of fixing vulnerabilities after deployment, as well as the impact on the software's users.

6.3. Undetected Threats. Considering the vulnerabilities found in VOMS and not reported by our tool, 2 vulnerability categories appear.

First category is vulnerabilities that lie outside of the scope of automated assessment. This category has been summarized as business-specific vulnerabilities, where the vulnerability is the result of improper implementation of domain specific requirements. While these are considered vulnerabilities, they are in no way related to the architecture or technology of the application and therefore cannot be detected using AutSEC's methodology. An example of this for VOMS is the incorrect check of certain certificate attributes; while this possesses a security threat to VOMS users, it is entirely related to the domain requirements.

The second category is vulnerabilities that lie within the scope of automated assessment. In the case of VOMS there are 2 vulnerabilities that enter this category, *DoS attacks* and *insecure third party library linking*. These vulnerabilities could be added to AutSEC's *attack patterns*. In our current implementation these types of vulnerabilities have not been included in the *attack patterns* as they are not due to specific architectures. *DoS* vulnerabilities, for example, are a generic issue that can affect any host providing service on a network.

This poses a complex issue between exhaustiveness and relevance, as reporting too many generic vulnerabilities can hurt the visibility of those specific to the system, as well as the difficulty of properly ranking these vulnerabilities without further information not found in the current architectural diagrams. These types of vulnerabilities are easily identified in our current representation, that is, a threat whose attack tree requires the presence of only a single DFD element to be detected.

This is a subject that will be explored in the continuation of this research; one option currently under review is to add

a fourth type of report that covers generic vulnerabilities for each technology used in a project.

6.4. Detected Threats Not Found in VOMS. There were a number of vulnerabilities reported by our tool for which no matching vulnerability was found in VOMS, as can be seen by comparing Tables 1 and 2. For example, SQL injections were identified as a potential threat but have not appeared in VOMS.

After analysis, the reported potential threats are considered to be relevant to the architecture of VOMS, they are not false positives, and their mitigation would benefit the VOMS software. That is to say, our team acting as security experts auditing VOMS would have explored whether these vulnerabilities were present or not as they are likely to occur and are potentially damaging.

Considering that the threats are relevant to the architecture, 3 possibilities exist to explain why they did not appear in VOMS. First, the proper mitigation techniques were implemented by the VOMS team. Second, implementation details of VOMS make this threat a nonissue even if the threat was present. Third, vulnerability exists but has not been uncovered.

Regardless of the reason why these vulnerabilities were not uncovered within VOMS, these threats are considered to be relevant to the architecture of the VOMS software, and therefore their reporting is considered valid.

7. Conclusions

In this paper, we addressed the problem of security expertise needed to perform risk assessments by automating the threat modeling process. By allowing nonsecurity developers to perform threat modeling we aimed at reducing the cost of secure development, making it more available.

To this end, we modeled a new data structure called *identification tree* that can be used to identify threats in software designs. We also designed a new model to describe countermeasures of threats called *mitigation tree*, which classifies the set of software specifications that are required to mitigate a specific threat. These data structures, along with ranking information over threats, were combined in a knowledge base called *attack patterns*.

In addition, we designed a new model, AutSEC, to automate threat modeling relying on the information contained in the *attack patterns*. We implemented this model in the form of an automated tool that works on top of the current Microsoft threat modeling methodology. AutSEC uses the *identification trees* to find the potential threats of a given software model. It purges irrelevant threats according to the developers business policies. And finally, it uses the *mitigation trees* to compute the software specifications of least effort needed to mitigate the detected threats during the development lifecycle.

The resulting least effort mitigations are directly related not only to *security by design* but also to *security by default*. This allows AutSEC to reach the willing balance between usability and security by default by asking the developers if the computed features are in good standing with their

requirements. If not, they are recomputed rejecting those that do not comply.

The output of applying the AutSEC model comes in the form of 3 reports: one for the design activity, which contains the architectural modifications needed to be carried out in the system, another one for the implementation phase, which contains implementation details to avoid the threats, and a final report for the verification phase containing a set of actions that are needed to be carried out to verify that detected threats were properly mitigated. These reports were designed so that any security-unaware developer can carry out their recommendations, which are written in terms that developers are accustomed to and provide ample resources in the case further information is required.

Our implementation of AutSEC was designed to be compatible with the current threat modeling tool distributed with the Microsoft SDL methodology. This offers the advantage that it can be easily integrated into development teams who already make use of the Microsoft methodology with minimum modifications to their software models in order to make them compatible with AutSEC's extended SDL attributes.

The experimental results of our tool were validated using the grid middleware component VOMS Admin. The results show that our tool is capable of detecting threats and offers the appropriate mitigation techniques. We have shown how the use of AutSEC during the development of VOMS Admin would have allowed the early detection of certain vulnerabilities. This has the advantage of limiting the financial impact of vulnerabilities, without requiring software developers to be trained in security and eliminate the impact on the software's users.

Further research on automated tools will focus on expanding AutSEC's vulnerability coverage while maintaining a high level of accuracy in their detection and will look into new ways of presenting additional threat information to the developers without undermining the quality of current reports.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank Andrea Ceccanti and Valery Tschopp from the National Institute of Nuclear Physics and Linda Cornwall from the Particle Physics Department at Oxford for making available the vulnerability data regarding VOMS that allowed the validation of our model. This research has been supported by the MINECO (MICINN), Spain, under contract TIN2011-24384.

References

- [1] P. Chandra, T. Wohleber, J. Feragamo, and J. Williams, *OWASP CLASP V1.2. Comprehensive, Lightweight Application Security Process*, OWASP, Rhode Island, RI, USA, 2007.

- [2] G. McGraw, *Software Security: Building Security*, Addison-Wesley, Reading, Mass, USA, 2006.
- [3] H. Michael and L. Steve, *The Security Development Lifecycle*, Microsoft Press, Redmond, Wash, USA, 2006.
- [4] B. De Win, R. Scandariato, K. Buyens, J. Grégoire, and W. Joosen, "On the secure software development process: CLASP, SDL and Touchpoints compared," *Information and Software Technology*, vol. 51, no. 7, pp. 1152–1171, 2009.
- [5] S. Swigart and S. Campbell, *Sdl Series, Article 4: Threat Modeling at Microsoft*, Microsoft Press, Redmond, Wash, USA, 2008.
- [6] S. Swigart and S. Campbell, Sdl series article 7: Evolution of the microsoft security development lifecycle, MSDN Magazine, 2009, <http://download.microsoft.com/download/C/6/4/C64A-557A-F6CD-48DD-B890-9B0C6270665F/SDL.Series.7.pdf>.
- [7] B. Schneier, "Attack trees," *Dr. Dobbs' Journal*, vol. 24, no. 12, pp. 21–29, 1999.
- [8] L. A. Cortés, P. Eles, and Z. Peng, "A survey on hardware/software codesign representation models," 1999.
- [9] F. Swiderski and W. Snyder, *Threat Modeling. Professional Series*, Microsoft Press, Redmond, Wash, USA, 2004.
- [10] Microsoft, 2013, <http://www.microsoft.com/security/sdl/adopt/threatmodeling.aspx>.
- [11] MITRE Corporation, *Common Attack Pattern Enumeration and Classification (Cape) Schema Description*, MITRE Corporation, McLean, Va, USA, 2008.
- [12] R. A. Martin and S. Barnum, "Common weakness enumeration (cwe) status update," *ADA Letters*, vol. 28, no. 1, pp. 88–91, 2008.
- [13] The Open Web Application Security Project, 2005, <http://www.owasp.org/>.
- [14] P. K. Manadhata and J. M. Wing, "An attack surface metric," *IEEE Transactions on Software Engineering*, vol. 37, no. 3, pp. 371–386, 2011.
- [15] NIACAP: National security telecommunications and information systems security committee (2000), <http://www.cnss.gov/Assets/pdf/cnssi.4009.pdf>.
- [16] M. Jorgensen, "Practical guidelines for expert-judgment-based software effort estimation," *IEEE Software*, vol. 22, no. 3, pp. 57–63, 2005.
- [17] G. Ruiz and E. Heymann, Extended resource material for paper (2014), <http://research.cs.wisc.edu/mist/includes/papers.html>.
- [18] J. A. Kupsch, B. P. Miller, E. Heymann, and E. César, "First principles vulnerability assessment," in *Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop (CCSW '10)*, pp. 87–92, ACM, New York, NY, USA, 2010.
- [19] J. A. Kupsch and B. P. Miller, "Manual vs. automated vulnerability assessment: a case study," in *Proceedings of the 1st International Workshop on Managing Insider Security Threats (MIST '09)*, pp. 83–97, 2009.

Research Article

A Provably Secure Revocable ID-Based Authenticated Group Key Exchange Protocol with Identifying Malicious Participants

Tsu-Yang Wu,^{1,2} Tung-Tso Tsai,³ and Yuh-Min Tseng³

¹ Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China

² Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen 518055, China

³ Department of Mathematics, National Changhua University of Education, Jin-De Campus, Changhua City 500, Taiwan

Correspondence should be addressed to Tsu-Yang Wu; wutsuyang@gmail.com

Received 12 March 2014; Accepted 16 May 2014; Published 1 June 2014

Academic Editor: Fei Yu

Copyright © 2014 Tsu-Yang Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The existence of malicious participants is a major threat for authenticated group key exchange (AGKE) protocols. Typically, there are two detecting ways (passive and active) to resist malicious participants in AGKE protocols. In 2012, the revocable identity- (ID-) based public key system (R-IDPKS) was proposed to solve the revocation problem in the ID-based public key system (IDPKS). Afterwards, based on the R-IDPKS, Wu et al. proposed a revocable ID-based AGKE (RID-AGKE) protocol, which adopted a passive detecting way to resist malicious participants. However, it needs three rounds and cannot identify malicious participants. In this paper, we fuse a noninteractive confirmed computation technique to propose the first two-round RID-AGKE protocol with identifying malicious participants, which is an active detecting way. We demonstrate that our protocol is a provably secure AGKE protocol with forward secrecy and can identify malicious participants. When compared with the recently proposed ID/RID-AGKE protocols, our protocol possesses better performance and more robust security properties.

1. Introduction

In the past, group-oriented applications, such as collaboration works and teleconference, were popularly and widely used in the Internet. Authenticated group key exchange (AGKE) protocol [1] is a cryptographic primitive which provides secure group communications for users in cooperative and distributed applications. During executing the protocol, group participants not only cooperatively generate a common key which is used to encrypt the transmitted messages but also authenticate the participants' identities.

The existence of malicious participants is a major threat for AGKE protocols. The goal of malicious participants is to disturb the establishing of common keys. Hence, how to resist malicious participants in AGKE protocols becomes a critical research. Typically, there are two detecting ways to resist malicious participants. (I) Passive detection [2–4]: it involves an explicit key confirmation approach in AGKE protocols. The resulted protocols only detect the existence of malicious participants and an additional round is required.

(II) Active detection [5, 6]: it adopts a noninteractive confirmed computation technique into AGKE protocols. The resulted protocols can identify the identities of malicious participants without additional round. However, the computational cost of active detection is time-consuming than the one of passive detection.

Quite recently, the revocable identity- (ID-) based public key system (R-IDPKS) was proposed to solve the revocation problem of users in the ID-based public key system (IDPKS). The concept of IDPKS was introduced by Shamir [7] in 1984 and was practiced by Bonch and Franklin [8] in 2001. Indeed, they [8] had suggested a solution that the private key generator (PKG) renews these nonrevoked users' private keys periodically to answer the revocation problem in the IDPKS. The approach can be used to revoke the compromised or misbehaving users. Nevertheless, the heavy workload arose from the PKG for renewing users' private keys periodically.

In 2008, Boldyreva et al. [9] proposed a revocable ID-based encryption (RIBE) scheme by using binary tree. Their scheme can reduce the PKG's workload mentioned in the

Boneh-Franklin solution [8]. However, this scheme is based on a weak security model called the relaxed selective-ID model [10]. In 2009, Libert and Vergnaud [11] relied on Boldyreva et al.'s RIBE to present a secure RIBE scheme under an adaptive-ID model. Recently, Seo and Emura [12] demonstrated Boldyreva et al.'s scheme [9] is vulnerable to decryption key exposure and then proposed a provably secure tree based RIBE scheme. Subsequently, Seo and Emura [13] presented a hierarchical RIBE scheme to solve the open problem mentioned in [11].

In 2011, Tseng and Tsai [14] proposed a practical RIBE scheme over a public channel. The key construction of the Tseng-Tsai scheme is different from the previous schemes [9, 11–13]. In [14], each user's private key consists of a fixed initial private key and an update key, where the update key is renewed along with the current period. For an honest (nonrevoked) user, the PKG periodically issues new update key and sends it to the user via a public channel. Upon receiving the new updating key, the user can renew her/his private key by herself/himself. To revoke a malicious user, the PKG only stops issuing the new update key in current period. Thus, the user cannot compute the newest private key. In other words, she/he cannot execute any cryptographic behaviors in later periods. Later on, several revocable ID-based cryptographic schemes based on the Tseng-Tsai R-IDPKS [14] were presented such as encryption [15], signature [16, 17], authenticated group key exchange (AGKE) [4], and signcryption [18].

In 2012, Wu et al. [4] proposed the first provably secure revocable ID-based AGKE (RID-AGKE) protocol. Their protocol adopted a passive detecting way to resist malicious participants. However, it requires three rounds and cannot identify the identities of malicious participants. In this paper, we fuse the key construction of Tseng-Tsai R-IDPKS [14] and a noninteractive confirmed computation technique [6] to present a two-round RID-AGKE protocol with identifying malicious participants. In our protocol, each group participant can confirm whether the broadcast values are correctly computed by other participants. Based on the detecting approach, our protocol can easily identify the participants who maliciously broadcast the incorrect values to disturb the common key establishing. The framework and security notions for RID-AGKE protocols are defined to formalize possible threats and attacks. We demonstrate the security of our protocol in the random oracle model [19] and under two mathematical assumptions (the computational Diffie-Hellman and the decisional bilinear Diffie-Hellman). Finally, we make the comparisons between our protocol and the recently proposed ID/RID-AGKE protocols to show the advantages of the proposed protocol.

The rest of this paper is organized as follows. We briefly review the concepts of bilinear pairings and related mathematical problems in Section 2. The security model and notions of RID-AGKE are presented in Section 3. We propose a concrete RID-AGKE protocol in Section 4. Security analysis of the proposed RID-AGKE protocol is demonstrated in Section 5. We make the performance analysis and comparisons in Section 6. Conclusions are drawn in Section 7.

2. Preliminaries

In this section, we briefly review the properties of bilinear pairings and related mathematical problems. For the details, a reader can refer to [8, 20, 21] for full descriptions.

2.1. Bilinear Pairings. Let G_1 and G_2 be two groups of a large prime order q , where G_1 is an additive cyclic group and G_2 is a multiplicative cyclic group. A bilinear pairing e is a map defined by $e : G_1 \times G_1 \rightarrow G_2$ and satisfies the following three conditions.

- (1) *Bilinearity*: for all $P, Q \in G_1$ and $a, b \in \mathbb{Z}_q$, $e(aP, bQ) = e(P, Q)^{ab}$.
- (2) *Nondegeneracy*: there exist $P, Q \in G_1$ such that $e(P, Q) \neq 1$.
- (3) *Computability*: for all $P, Q \in G_1$, there exists an algorithm to compute $e(P, Q)$.

2.2. Mathematical Hard Problems and Assumptions. Here, we present two mathematical hard problems and define the corresponding assumptions as follows.

- (1) *Computational Diffie-Hellman (CDH) problem*: given $P, aP, bP \in G_1$ for some $a, b \in \mathbb{Z}_q^*$, the CDH problem is to compute $abP \in G_1$.
- (2) *Decisional bilinear Diffie-Hellman (DBDH) problem*: given $P, aP, bP, cP, dP \in G_1$ for some $a, b, c, d \in \mathbb{Z}_q^*$, the DBDH problem is to distinguish $(P, aP, bP, cP, dP, e(P, P)^{abc})$ from $(P, aP, bP, cP, dP, e(P, P)^d)$.

Definition 1 (CDH assumption). Given $P, aP, bP \in G_1$ for some $a, b \in \mathbb{Z}_q^*$, there does not exist a probabilistic polynomial-time algorithm A with a nonnegligible probability to compute $abP \in G_1$. The advantage of A within running time t is defined as $Adv_{CDH}(t) = \Pr[A(P, aP, bP) = abP \mid P, aP, bP \in G_1]$.

Definition 2 (DBDH assumption). Given $P, aP, bP, cP, dP \in G_1$ for some $a, b, c, d \in \mathbb{Z}_q^*$, there does not exist a probabilistic polynomial-time algorithm A with nonnegligible probability to distinguish $(P, aP, bP, cP, dP, e(P, P)^{abc})$ from $(P, aP, bP, cP, dP, e(P, P)^d)$. The advantage of A within running time t is defined as $Adv_{DBDH}(t) = \Pr[A(e(P, P)^{abc}, e(P, P)^d) = 1 \mid P, aP, bP, cP, dP \in G_1]$.

3. Model and Notions

In this section, we define the model and notions for RID-AGKE protocol. Note that some of the following definitions and notations are referred to in [4, 6, 22–24].

Initialization. The initialization of RID-AGKE protocol has three algorithms.

- (1) *Setup Algorithm.* This algorithm is a probabilistic algorithm which takes as input a security parameter k and a total

number z of periods. It returns a system private key s and public parameters $param$. Note that the whole life time of the system is divide into distinct periods $1, 2, \dots, z$. Here, $param$ is made public.

(2) *Initial Key Extract Algorithm*. This algorithm is a deterministic algorithm which takes as input the system private key s and a participant's identity ID . It returns the participant's initial private key DID .

(3) *Key Update Algorithm*. This algorithm is a deterministic algorithm which takes as input the system private key s , a participant's identity ID , and a period index j , where $1 \leq j \leq z$. It returns the participant's update key TID_j .

Here, note that the participant's private key for period j is defined by $DID_j = DID + TID_j$.

Related Notions. For simplicity, there is a fixed set $G = \{U_1, U_2, \dots, U_n\}$ with polynomial size of potential participants. Assume that each participant U_i has a unique identity $ID_i \in \{0, 1\}^*$. Any subset of G may run a RID-AGKE protocol many times (possibly concurrently) in some period index j to establish a group session key, where $1 \leq j \leq z$ and z is a total number of periods. Note that the set of participants' identities, $\mathbf{ID} = \{ID_1, ID_2, \dots, ID_n\}$ is known by all participants (including adversary).

An instance t of participant U in period j is denoted by $\Pi_U^{j(t)}$, where t is a positive integer. Each instance $\Pi_U^{j(t)}$ has associated with seven variables as follows.

- (i) $state_U^{j(t)}$: it presents the current state of instance $\Pi_U^{j(t)}$.
- (ii) $acc_U^{j(t)}$ and $term_U^{j(t)}$: they take Boolean values to demonstrate whether $\Pi_U^{j(t)}$ has accepted or terminated. Informally, we say that an instance has accepted meaning that it does not detect any incorrect behavior. An instance is called terminating if it has sent and received messages. Note that a terminated instance may also possibly accept.
- (iii) $used_U^{j(t)}$: it indicates whether $\Pi_U^{j(t)}$ is used in a RID-AGKE protocol.
- (iv) $pid_U^{j(t)}$: the partner ID of instance $\Pi_U^{j(t)}$ is a set which contains the identities of participants in the group with whom $\Pi_U^{j(t)}$ wants to establish a group session key (including U itself).
- (v) $sid_U^{j(t)}$: the session ID of instance $\Pi_U^{j(t)}$ is a concatenation of all messages sent and received by the instance in a given execution of RID-AGKE protocol.
- (vi) $sk_U^{j(t)}$: a group session key which is accepted by instance $\Pi_U^{j(t)}$.

In the following definitions, we will only focus on the three variables $pid_U^{j(t)}$, $sid_U^{j(t)}$, and $sk_U^{j(t)}$. The remaining variables will be left implicit. We say that two instances $\Pi_U^{j(t)}$ and $\Pi_{U'}^{j(v)}$ are *partnered* if (1) they have accepted the same group session key; (2) $sid_U^{j(t)} = sid_{U'}^{j(v)}$; and (3) $pid_U^{j(t)} = pid_{U'}^{j(v)}$.

Adversarial Model. An adversary A can be viewed as a probabilistic polynomial-time algorithm. Here, we assume that A can potentially control all communications in a RID-AGKE protocol. The interaction between A and instances of participants in the protocol is modeled by the following oracles.

- (i) *Execute* (V, j): when A makes *Execute query* on (V, j), it executes the RID-AGKE protocol between the unused instances of participants in V for period index j and then returns a transcript of the execution, where V is a subset of G . Here, *Execute query* is used to model passive attacks.
- (ii) *Inextract* (ID_U): when A makes *Initial key extract query* on identity ID_U , it generates an initial private key DID_U corresponding to ID_U and returns it to A , where $ID_U \notin \mathbf{ID}$.
- (iii) *Kupdate* (ID_U, j): when A makes *Key update query* on (ID_U, j), it generates an update key $TID_{U,j}$ corresponding to (ID_U, j) and returns it to A , where $ID_U \notin \mathbf{ID}$ and j is a period index.
- (iv) *Send* (U, j, t, M): when A makes *Send query* on (U, j, t, M), it sends message M to instance $\Pi_U^{j(t)}$ and then returns the reply generated by this instance according to procedures of RID-AGKE protocol.
- (v) *Reveal* (U, j, t): when A makes *Reveal query* on (U, j, t), it returns a group session key $sk_U^{j(t)}$ for a terminated instance $\Pi_U^{j(t)}$. Here, *Reveal query* is used to model known session key attacks.
- (vi) *Corrupt* (ID_U, j): when A makes *Corrupt query* on (ID_U, j), it returns a private key $DID_{U,j}$ of ID_U in period j . Note that *Corrupt query* models the corruption of this participant at a time in which it is not currently executing the protocol. We say that a participant U is honest if and only if no *Corrupt query* has been made by A .
- (vii) *Test* (U, j, t, \cdot): at any time, the adversary A makes *Test query* only once to this oracle during A 's execution. In this moment, a random coin $b \in \{0, 1\}$ is selected. If $b = 1$, a group session key $sk_U^{j(t)}$ is returned. Otherwise, a random value is returned. Here, *Test query* is used to model the semantic security of group session key.

According to the above adversarial model, we define two types of adversaries. A *passive adversary* is allowed to make *Execute*, *Reveal*, *Corrupt*, and *Test* queries. An *active adversary* is allowed to make the above all queries. In order to get more precise analysis, we still use *Execute query* though it can be substituted by making *Send query* repeatedly.

Remark 3. According to the adversarial model above, the adversary A can compute the participant U 's private key $DID_{U,j} = DID_U + TID_{U,j}$ for period index j while A makes both *Initial key extract query* on ID_U and *Key update query* on (ID_U, j) simultaneously. Hence, we disallow A to make both queries in the same time.

Correctness. A RID-AGKE protocol is called *correct* if the following three conditions hold.

- (1) All participants are honest and all messages are delivered honestly.
- (2) $acc_U^{j(t)} = acc_{U'}^{j(v)}$ = "True" and $sk_U^{j(t)} = sk_{U'}^{j(v)}$.
- (3) $sid_U^{j(t)} = sid_{U'}^{j(v)}$ and $pid_U^{j(t)} = pid_{U'}^{j(v)}$ for all participants $U, U' \in V \subseteq G$ with instances $\Pi_U^{j(t)}$ and $\Pi_{U'}^{j(v)}$.

Freshness. We say that an instance $\Pi_U^{j(t)}$ is called *fresh* (or called holding a *fresh* group session key $sk_U^{j(t)}$) if the following three conditions hold.

- (1) $\Pi_U^{j(t)}$ has accepted a group session key $sk_U^{j(t)}$.
- (2) Neither $\Pi_U^{j(t)}$ nor its partners have been made *Reveal query*.
- (3) No *Corrupt query* has been made on $ID_V \in pid_U^{j(t)}$ before *Send query* to $\Pi_U^{j(t)}$ or *Send query* to $\Pi_{U'}^{j(v)}$, where $ID_{U'} \in pid_{U'}^{j(v)}$.

Here, we assume all instances are *fresh*. Note that the notion of *freshness* is defined appropriately for the purpose of forward secrecy.

Secure RID-AGKE. A secure RID-AGKE protocol contains the following four parts.

- (1) *Freshness.*
- (2) *Security of RID-AGKE Protocol.* The security of RID-AGKE protocol is defined in the following game played between an active adversary A and a set of instances:
 - (a) *initialization:* the system private key, public parameters, and participants' private keys are generated in this phase;
 - (b) *query:* A may make different types of queries to oracles and gets back the answers corresponding to the RID-AGKE protocol;
 - (c) *guess:* finally, the adversary A outputs its guess for the coin b in *Test query* and terminates.

In this game, the goal of A is to distinguish a group session key from a random value. Let *Succ* be the event that A correctly guesses the coin b in *Test query*. The advantage of A in attacking a RID-AGKE protocol Ψ is defined by $Adv_{A,\Psi}(k) = |2 \cdot \Pr[\text{Succ}] - 1|$. We say that the protocol Ψ is secure, if the advantage $Adv_{A,\Psi}(k)$ is negligible.

(3) *Forward Secrecy.* We say that a RID-AGKE protocol Ψ provides *forward secrecy*. It means that though an adversary A obtains participants' private keys in Ψ , the previous establishing group session keys is preserved. The advantage of A in attacking the protocol Ψ within running time t is defined by $Adv_{\Psi}^{RIDAGKE-fs}(t, q_{ex}, q_s)$, where q_{ex} and q_s are the maximum numbers of making *Execute* and *Send queries*, respectively.

(4) *Authentication.* We say that a RID-AGKE protocol Ψ provides *implicit key authentication* if all participants in Ψ are guaranteed that nobody other than their partners can learn the session key. In other words, any adversary should not learn the key. Note that this security property does not guarantee that the partners have computed the key.

Malicious Participant. A participant U_m is called *malicious* in a RID-AGKE protocol Ψ if he is a legal participant but is fully controlled by adversary. The goal of malicious participant is to disturb the group key establishing in Ψ .

4. Concrete Protocol

In this section, we propose a concrete RID-AGKE protocol with identifying malicious participants. Our protocol fuses the Tseng-Tsai R-IDPKS [14] and a noninteractive confirmed computation technique [6]. In the initialization phase, given a security parameter k and a total number z of periods, a private key generator (PKG) executes *Setup algorithm* to generate the system private key s and the public parameters $param = \{G_1, G_2, e, q, P, P_{pub}, H_1, H_2, H_3, H_4\}$ defined in Notations section at the end of the paper.

When a participant U with identity $ID_U \in \{0, 1\}^*$ wants to obtain her/his initial private key DID_U , the PKG runs *Initial key extract algorithm* to compute $DID_U = s \cdot H_1(ID_U) = s \cdot QID_U$ and returns it to U via a secure channel. For a nonrevoked participant U with identity ID_U in time period j , the PKG runs *Key update algorithm* to compute her/his update key $TID_{U,j} = s \cdot H_2(ID_U, j) = s \cdot RID_{U,j}$ and returns it to U via a public channel, where $1 \leq j \leq z$. Hence, any nonrevoked participant U can update her/his private key $DID_{U,j} = DID_U + TID_{U,j}$ by itself in period j .

Let $G = \{U_1, U_2, \dots, U_n\}$ be a set of participants who want to establish a group session key SK_j in period j . We assume that each U_i has a unique identity $ID_i \in \{0, 1\}^*$ as public key and U_i 's private key is $DID_{i,j} = DID_i + TID_{i,j}$ for period j . Note that the indices are subject to modulo n ; that is, U_{n+1} and U_0 denote U_1 and U_n , respectively. Finally, $m \in \{0, 1\}^*$ is a preknown common message by all participants. The details of proposed RID-AGKE protocol are described as follows.

Round 1. Each participant U_i randomly selects a secret value $a_i \in Z_q^*$ and computes $P_i = a_i \cdot P, h_i = H_3(ID_i, PID_j, j, m, P_i)$, and $V_i = DID_{i,j} + a_i \cdot h_i \cdot P_{pub}$, where PID_j denotes the concatenation of all participants' identities in period j ; that is, $PID_j = ID_1 \parallel ID_2 \parallel \dots \parallel ID_n$. Finally, each U_i broadcasts (ID_i, j, P_i, V_i) to other participants.

Round 2. Upon receiving $(ID_{i-1}, j, P_{i-1}, V_{i-1})$ and $(ID_{i+1}, j, P_{i+1}, V_{i+1})$, each U_i first verifies them by checking

$$\begin{aligned}
 & e \left(P, \sum_{k \in \{-1, 1\}} V_{i+k} \right) \\
 & \stackrel{?}{=} e \left(P_{pub}, \sum_{k \in \{-1, 1\}} H_1(ID_{i+k}) + H_2(ID_{i+k}, j) \right. \\
 & \quad \left. + h_{i+k} \cdot P_{i+k} \right),
 \end{aligned} \tag{1}$$

where $h_{i+k} = H_3(ID_{i+k}, PID_j, j, m, P_{i+k})$. If the verification is true, each U_i uses her/his secret value a_i to compute $D_i = e(a_i \cdot (P_{i+1} - P_{i-1}), P_{pub})$. Then, U_i randomly selects a value $r_i \in Z_q^*$ and computes a tuple $(ID_i, j, D_i, \alpha_i, \beta_i, \gamma_i)$, where $\alpha_i = r_i \cdot P$, $\beta_i = r_i \cdot (P_{i+1} - P_{i-1})$, $\gamma_i = r_i \cdot P_i + w_i \cdot a_i \cdot P_{pub}$, $w_i = H_4(ID_i \parallel PID_j \parallel j \parallel D_i \parallel S_j, P_{i+1} - P_{i-1}, \alpha_i, \beta_i)$, and $S_j = P_1 \parallel P_2 \parallel \dots \parallel P_n$. Finally, U_i sends this tuple to all other participants.

Group Key Computation. Upon receiving all $(ID_k, j, D_k, \alpha_k, \beta_k, \gamma_k)$ for $k = 1, 2, \dots, n$ except i , each U_i verifies them by checking

$$e\left(P, \sum_{k=1, k \neq i}^n \gamma_k\right) = \prod_{k=1, k \neq i}^n e(P_k, \alpha_k + w_k \cdot P_{pub}), \quad (2)$$

$$e(P_{k+1} - P_{k-1}, \gamma_k) \stackrel{?}{=} e(\beta_k, P_k) \cdot D_k^{w_k}.$$

If the two verifications hold, U_i can confirm that each D_k is computed by U_k using her/his secret a_k honestly for $k = 1, 2, \dots, n$ except i . Finally, in period j , each participant U_i can compute the group session key $SK_j = e(a_i \cdot P_{i-1}, P_{pub})^n \cdot D_i^{n-1} \cdot D_{i+1}^{n-2} \cdot \dots \cdot D_{i-2}$.

Identifying Malicious Participant. When a malicious participant U_m tries to send a wrong tuple $(ID_m, D_m, j, \alpha_m, \beta_m, \gamma_m)$ to disrupt the establishment of group session key, he will be identified as a malicious participant by using the following two verifying equations: $e(P, \gamma_k) \stackrel{?}{=} e(P_k, \alpha_k + w_k \cdot P_{pub})$ and $e(P_{k+1} - P_{k-1}, \gamma_k) \stackrel{?}{=} e(\beta_k, P_k) \cdot D_k^{w_k}$. Later on, U_m will be deleted from the participant set G and other honest participants may rerun the protocol.

5. Security Analysis

In this section, we prove the security of the proposed RID-AGKE protocol in the random oracle model [19] and under the CDH and DBDH assumptions.

ID and Forgery Attacks

Theorem 4. *The proposed RID-AGKE protocol is secure against ID and forgery attacks.*

Proof. Note that we adopt a revocable ID-based signature (RIDS) scheme [16] in Round 1 and a pairing-based signature scheme [6] in Round 2, respectively. The two signature schemes had been proven secure against ID and forgery attacks for single signature and multiple signatures with batch verification. Therefore, the proposed RID-AGKE protocol Ψ is secure against ID and forgery attacks.

Secure RID-AGKE Providing Forward Secrecy. Now, we demonstrate that the proposed RID-AGKE protocol Ψ is a secure RID-AGKE providing forward secrecy. Note that we use a similar technique in [3, 4, 6] to prove Theorem 5. \square

Theorem 5. *Assume that four hash functions H_1, H_2, H_3 , and H_4 are random oracles. Then, the proposed RID-AGKE protocol Ψ is a secure RID-AGKE providing forward secrecy under the decisional bilinear Diffie-Hellman (DBDH) and the computational Diffie-Hellman (CDH) assumptions. Concretely,*

$$Adv_{\Psi}^{RIDAGKE-fs}(t, q_{ex}, q_s) \leq 2nq_{ex} \cdot Adv_{DBDH}(t) + Adv_{\Psi}^{forge}(t), \quad (3)$$

where q_{ex} and q_s are total numbers of making Execute and Send queries, respectively. Note that $Adv_{\Psi}^{forge}(t)$ denotes the advantage of any forgers successfully attacking the protocol Ψ .

Proof. Assume that A is an active adversary in attacking the proposed RID-AGKE protocol Ψ with a nonnegligible advantage. Now, we consider the two possible cases. The first case is that A with the advantage can impersonate a participant (i.e., forging authentication transcripts). Another case is that A with the advantage can break the protocol Ψ without modifying any transcripts.

Case 1. We assume that the adversary A with an adaptive impersonation ability can break the RID-AGKE protocol Ψ . Using A , we would like to construct a forger F which can return valid signature tuples (ID, j, aP, V) and $(ID, j, D, \alpha, \beta, \gamma)$ with respect to the proposed protocol Ψ as follows. The forger F first generates all needed system parameters and keys. Then, F simulates the oracle queries made by A . This simulation is called perfect indistinguishable from A 's oracle queries except that A makes *Corrupt query* on (ID, j) , where j is a period index. If it occurs, F fails and stops. Otherwise, when A generates two signature tuples (ID, j, aP, V) and $(ID, j, D, \alpha, \beta, \gamma)$, F returns the tuples (ID, j, aP, V) and $(ID, j, D, \alpha, \beta, \gamma)$. Let *Forge* be the event that the adversary A successfully generates two valid signature tuples. Then, the probability that F successfully returns two valid signature tuples is bounded by $\Pr_A[\text{Forge}] \leq Adv_{F, \Psi}^{forge}(t) \leq Adv_{\Psi}^{forge}(t)$.

Case 2. We assume that the adversary A can break the proposed RID-AGKE protocol without modifying any transcripts. We first focus on the case that A makes *Execute query* once on $(ID_1, ID_2, \dots, ID_n, j)$ and then extends this to the case that A makes multiple *Execute queries*, where the number of participants n and period j are selected by A . The real execution of Ψ is given by

$$param = \left\{ \begin{array}{l} (G_1, G_2, e) \leftarrow \text{PKG}; P \leftarrow G_1; s \leftarrow Z_q^*; P_{pub} = s \cdot P; \\ QID_1, \dots, QID_n, RID_{1,j}, \dots, RID_{n,j} \leftarrow G_1; \\ DID_{1,j} = (QID_1 + RID_{1,j}) \cdot s, \dots, DID_{n,j} = (QID_n + RID_{n,j}) \cdot s; \\ (G_1, G_2, e, P, P_{pub}, PID) \end{array} \right\},$$

$$Real = \left\{ \begin{array}{l} a_1, \dots, a_n, h_1, \dots, h_n, r_1, \dots, r_n, w_1, \dots, w_n \leftarrow Z_q^*; \\ P_1 = a_1P, \dots, P_n = a_nP; V_1 = DID_{1,j} + a_1h_1P_{pub}, \dots, V_n = DID_{n,j} + a_nh_nP_{pub}; \\ D_1 = e(P_2 - P_n, P_{pub})^{a_1}, \dots, D_n = e(P_1 - P_{n-1}, P_{pub})^{a_n}; \\ \alpha_1 = r_1P, \dots, \alpha_n = r_nP; \beta_1 = r_1(P_2 - P_n), \dots, \beta_n = r_n(P_1 - P_{n-1}); \\ \gamma_1 = r_1P_1 + w_1a_1P_{pub}, \dots, \gamma_n = r_nP_n + w_na_nP_{pub}; \\ T = (P_1, \dots, P_n, V_1, \dots, V_n, D_1, \dots, D_n, \alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n, \gamma_1, \dots, \gamma_n); \\ SK_j = e(a_1P_n, P_{pub})^n \cdot D_1^{n-1} \cdot D_2^{n-2} \dots D_{n-1} : (j, T, SK_j) \end{array} \right\}, \quad (4)$$

where T denotes the transcript and SK_j is the group session key for period j .

In *Real*, each $D_i = e(P_{i+1} - P_{i-1}, P_{pub})^{a_i} = e(a_iP_{i+1}, P_{pub})/e(a_iP_{i-1}, P_{pub}) = e(a_i a_{i+1}P, P_{pub})/e(a_{i-1}a_iP, P_{pub})$. by

the bilinear pairing operations. We can use a random value $d_{1,2} \in Z_q^*$ to substitute $a_1 \cdot a_2$. Thus, a new distribution *Fake₁* is obtained as follows:

$$Fake_1 = \left\{ \begin{array}{l} d_{1,2}, a_1, \dots, a_n, h_1, \dots, h_n, r_1, \dots, r_n, w_1, \dots, w_n \leftarrow Z_q^*; \\ P_1 = a_1P, \dots, P_n = a_nP; \\ V_1 = DID_{1,j} + a_1h_1P_{pub}, \dots, V_n = DID_{n,j} + a_nh_nP_{pub}; \\ D_1 = \frac{e(d_{1,2}P, P_{pub})}{e(a_n a_1P, P_{pub})}, D_2 = \frac{e(a_2 a_3P, P_{pub})}{e(d_{1,2}P, P_{pub})}, \dots, D_n = \frac{e(a_n a_1P, P_{pub})}{e(a_{n-1} a_nP, P_{pub})}; \\ \alpha_1 = r_1P, \dots, \alpha_n = r_nP; \beta_1 = r_1(P_2 - P_n), \dots, \beta_n = r_n(P_1 - P_{n-1}); \\ \gamma_1 = r_1P_1 + w_1a_1P_{pub}, \dots, \gamma_n = r_nP_n + w_na_nP_{pub}; \\ T = (P_1, \dots, P_n, V_1, \dots, V_n, D_1, \dots, D_n, \alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n, \gamma_1, \dots, \gamma_n); \\ SK_j = e(a_1P_n, P_{pub})^n \cdot D_1^{n-1} \cdot D_2^{n-2} \dots D_{n-1} : (j, T, SK_j) \end{array} \right\}. \quad (5)$$

Note that A can obtain all private keys $DID_{i,j}$ and hash values h_i by making *Corrupt* and *Hash queries*. It means that A can compute all $a_i \cdot P_{pub} = h_i^{-1} \cdot (V_i - DID_{i,j})$ for $i = 1, 2, \dots, n$. Since the discrete logarithm assumption in G_1 is intractable, A cannot obtain some information about a_i from $a_i \cdot P_{pub}$ for $i = 1, 2, \dots, n$.

In the following claim, we want to show that to distinguish two distributions *Real* from *Fake₁* can be reduced to solve the decisional bilinear Diffie-Hellman (DBDH) problem. Let $\varepsilon(t) = Adv_{DBDH}(t)$.

Claim. For any algorithm A with running time t , we have

$$\begin{aligned} & \left| \Pr \left[(j, T, SK_j) \leftarrow Real \mid A(j, T, SK_j) = 1 \right] \right. \\ & \quad \left. - \Pr \left[(j, T, SK_j) \leftarrow Fake_1 \mid A(j, T, SK_j) = 1 \right] \right| \\ & \leq \varepsilon(t). \end{aligned} \quad (6)$$

Proof. As mentioned above, each $D_i = e(a_i a_{i+1}P, P_{pub})/e(a_{i-1}a_iP, P_{pub}) = e(P, P_{pub})^{a_i a_{i+1}} e(P, P_{pub})^{a_{i-1} a_i}$. Here, we use $\Gamma_{i,i+1}$ to substitute $e(P, P_{pub})^{a_i a_{i+1}}$ and then each D_i can be written into $\Gamma_{i,i+1}/\Gamma_{i-1,i}$ for $i = 1, 2, \dots, n$. Hence, the group session key SK_j also can be written into $(\Gamma_{n,1})^n \cdot D_1^{n-1} \cdot D_2^{n-2} \dots D_{n-1}$, where $(\Gamma_{n,1})^n = e(P, P_{pub})^{na_n a_1} = e(a_1 P_n, P_{pub})^n$.

To solve the DBDH problem, we use a technique to dispose the related parameter. Considering the following algorithm D which inputs $P_a = aP, P_b = bP$, and $P_c = cP \in G_1$ for some $a, b, c \in_R Z_q^*$. D first generates (j, T, SK_j) according to the distribution $Dist^1$. Then, D runs $A(j, T, SK_j)$ and outputs whatever A outputs. The distribution $Dist^1$ is defined as follows:

$$Dist^1 = \left\{ \begin{array}{l} a_1, \dots, a_n, h_1, \dots, h_n, u_1, \dots, u_{n-2}, r_1, \dots, r_n, w_1, \dots, w_n \leftarrow Z_q^*; \\ P_1 = a_1P, \dots, P_n = a_nP; V_1 = DID_{1,j} + a_1h_1P_{pub}, \dots, V_n = DID_{n,j} + a_nh_nP_{pub}; \\ \Gamma_{1,2} = g_{sab} \in G_2, \Gamma_{2,3} = e(P_b, P_{pub})^{u_1}, \Gamma_{i,i+1} = e(P, P_{pub})^{u_{i-2}u_{i-1}} \text{ for } i = 3 \text{ to } n-1 \\ \Gamma_{n,1} = e(P_a, P_{pub})^{u_{n-2}}; D_1 = \frac{\Gamma_{1,2}}{\Gamma_{n,1}}, \dots, D_n = \frac{\Gamma_{n,1}}{\Gamma_{n-1,n}}; \\ \alpha_1 = r_1P, \dots, \alpha_n = r_nP; \beta_1 = r_1(P_2 - P_n), \dots, \beta_n = r_n(P_1 - P_{n-1}); \\ \gamma_1 = r_1P_1 + w_1a_1P_{pub}, \dots, \gamma_n = r_nP_n + w_na_nP_{pub}; \\ T = (P_1, \dots, P_n, V_1, \dots, V_n, D_1, \dots, D_n, \alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n, \gamma_1, \dots, \gamma_n); \\ SK_j = (\Gamma_{n,1})^n \cdot D_1^{n-1} \cdot D_2^{n-2} \dots D_{n-1} : (j, T, SK_j) \end{array} \right\}. \quad (7)$$

Note that this distribution depends on P_a , P_b , and P_c .

By the above distribution $Dist^1$, let $\Gamma_{1,2} = e(P, P_{pub})^{ab} = e(P, P)^{sab}$. Then, we can obtain another distribution called $Dist^1_{DBDH}$. Obviously, $Dist^1_{DBDH}$ is identical to $Real$ because

$$\begin{aligned} SK_j &= (\Gamma_{n,1})(\Gamma_{1,2})(\Gamma_{2,3}) \cdots (\Gamma_{n-2,n-1})(\Gamma_{n-1,n}) \\ &= e(P_a, P_{pub})^{u_{n-2}} \cdot e(P, P)^{sab} \\ &\quad \cdot e(P_b, P_{pub})^{u_1} \cdots e(P, P_{pub})^{u_{n-4}u_{n-3}} \\ &\quad \cdot e(P, P_{pub})^{u_{n-3}u_{n-2}} \\ &= e(P, P)^{su_{n-2}a+sab+sbu_1+\cdots+su_{n-4}u_{n-3}+su_{n-3}u_{n-2}}. \end{aligned} \tag{8}$$

Similarly, let $\Gamma_{1,2} = e(P_c, P_{pub}) = e(P, P)^{sc}$ for some $c \neq ab \in Z_q^*$. Then, we can obtain another distribution called $Dist^1_{Random}$. Obviously, $Dist^1_{Random}$ is identical to $Fake_1$ because

$$\begin{aligned} SK_j &= (\Gamma_{n,1})(\Gamma_{1,2})(\Gamma_{2,3}) \cdots (\Gamma_{n-2,n-1})(\Gamma_{n-1,n}) \\ &= e(P_a, P_{pub})^{u_{n-2}} \cdot e(P, P)^{sc} \\ &\quad \cdot e(P_b, P_{pub})^{u_1} \cdots e(P, P_{pub})^{u_{n-4}u_{n-3}} \\ &\quad \cdot e(P, P_{pub})^{u_{n-3}u_{n-2}} \\ &= e(P, P)^{su_{n-2}a+sc+sbu_1+\cdots+su_{n-4}u_{n-3}+su_{n-3}u_{n-2}}. \end{aligned} \tag{9}$$

Therefore, we have

$$\begin{aligned} & \left| \Pr \left[(j, T, SK_j) \leftarrow Real \mid A(j, T, SK_j) = 1 \right] \right. \\ & \quad \left. - \Pr \left[(j, T, SK_j) \leftarrow Fake_1 \mid A(j, T, SK_j) = 1 \right] \right| \\ & \leq \varepsilon(t). \end{aligned} \tag{10}$$

This completes the proof of claim.

Using the same process in $Fake_1$, we can define other distributions $Fake_i$ for $i = 2, 3, \dots, n$. By a similar approach in claim, we can obtain the following $n-1$ equations in (11) for any adversary A with running time t

$$\begin{aligned} & \left| \Pr \left[(j, T, SK_j) \leftarrow Fake_1 \mid A(j, T, SK_j) = 1 \right] \right. \\ & \quad \left. - \Pr \left[(j, T, SK_j) \leftarrow Fake_2 \mid A(j, T, SK_j) = 1 \right] \right| \\ & \leq \varepsilon(t), \\ & \quad \vdots \\ & \left| \Pr \left[(j, T, SK_j) \leftarrow Fake_{n-1} \mid A(j, T, SK_j) = 1 \right] \right. \\ & \quad \left. - \Pr \left[(j, T, SK_j) \leftarrow Fake_n \mid A(j, T, SK_j) = 1 \right] \right| \\ & \leq \varepsilon(t). \end{aligned} \tag{11}$$

This implies

$$\begin{aligned} & \left| \Pr \left[(j, T, SK_j) \leftarrow Real \mid A(j, T, SK_j) = 1 \right] \right. \\ & \quad \left. - \Pr \left[(j, T, SK_j) \leftarrow Fake_n \mid A(j, T, SK_j) = 1 \right] \right| \\ & \leq n \cdot \varepsilon(t). \end{aligned} \tag{12}$$

In $Fake_n$, the values $d_{1,2}, d_{2,3}, \dots, d_{n-1,n}, d_{n,1}$ are constrained by T according to the following n equations:

$$\begin{aligned} \log_g D_1 &= s \cdot (d_{1,2} - d_{n,1}), \\ \log_g D_2 &= s \cdot (d_{2,3} - d_{1,2}), \dots, \\ \log_g D_n &= s \cdot (d_{n,1} - d_{n-1,n}), \end{aligned} \tag{13}$$

where $g = e(P, P)$. Since SK_j can be expressed as $e(P, P)^{sd_{1,2}+sd_{2,3}+\cdots+sd_{n,1}}$, we can obtain $\log_g SK_j = sd_{1,2} + sd_{2,3} + \cdots + sd_{n,1}$. Because $sd_{1,2} + sd_{2,3} + \cdots + sd_{n,1}$ is linear and independent from the set $\{\log_g D_i = s \cdot (d_{i,i+1} - d_{i-1,i}) \mid i = 1, 2, \dots, n\}$, it implies that SK_j is independent for the transcript T . In other words, for any adversary A

$$\begin{aligned} & \Pr \left[(j, T, SK_{j,0}) \leftarrow Fake_n, SK_{j,1} \leftarrow G_2 \mid A(j, T, SK_{j,b}) \right. \\ & \quad \left. = 1, b \leftarrow \{0, 1\} \right] = \frac{1}{2}. \end{aligned} \tag{14}$$

Therefore, the advantage of A on the event $\neg Forge$ is bounded by $2n \cdot Adv_{DBDH}(t)$. Combining the two cases, the advantage of A is bounded by

$$Adv_{\Psi}^{RIDAGKE-fs}(t, 1, q_s) \leq 2n \cdot Adv_{DBDH}(t) + Adv_{\Psi}^{forge}(t). \tag{15}$$

Finally, a standard hybrid argument immediately demonstrates that

$$\begin{aligned} Adv_{\Psi}^{RIDAGKE-fs}(t, q_{ex}, q_s) &\leq 2nq_{ex} \cdot Adv_{DBDH}(t) \\ &\quad + Adv_{\Psi}^{forge}(t) \text{ for } q_{ex} > 1. \end{aligned} \tag{16}$$

□

Under the decisional bilinear Diffie-Hellman (DBDH) assumption, the advantage $Adv_{DBDH}(t)$ is negligible. By Theorem 4, the advantage $Adv_{\Psi}^{forge}(t)$ is also negligible. Hence, we can obtain that the advantage $Adv_{\Psi}^{RIDAGKE-fs}(t, q_{ex}, q_s)$ is negligible according to the result in Theorem 5. It implies that the proposed RID-AGKE protocol Ψ is a secure RID-AGKE providing forward secrecy.

Identifying Malicious Participant

Theorem 6. *The proposed RID-AGKE protocol can identify malicious participants.*

TABLE 1: Comparisons between our protocol and the previously proposed AGKE protocols.

	Tseng's AGKE [25]	Choi et al.'s ID-AGKE [26]	Wu et al.'s ID-AGKE [6]	Wu et al.'s RID-AGKE [4]	Our protocol
Public key setting	ElGmal	IDPKS	IDPKS	R-IDPKS	R-IDPKS
Certificate management	Required	Not required	Not required	Not required	Not required
Rounds	2	2	2	3	2
Computational cost for each participant	$(8n-2)T_{exp} + (n+1)T_{inv}$	$6TG_e + (n+11)TG_{mul} + (n+3)TG_H$	$(3n+3)TG_e + (n+10)TG_{mul} + 3TG_H + (n-1)T_{exp}$	$8TG_e + (2n+8)TG_{mul} + 4nTG_H$	$(3n+2)TG_e + (n+9)TG_{mul} + 4TG_H + (n-1)T_{exp}$
Security	Provably secure	Existing attack [27]	Provably secure	Provably secure	Provably secure
Revocation functionality	Using CRL [28]	No	No	Yes	Yes
Resistant to malicious participants	Yes (confirmed computation)	No	Yes (confirmed computation)	Yes (explicit key confirmation)	Yes (confirmed computation)
Identifying malicious participants	Yes	No	Yes	No	Yes

Proof. Note that in Round 2 a noninteractive confirmed computation technique is involved in adopted pairing-based signature scheme. The security of confirmed computation had been proven in [6]. Concretely, each participant U_i can confirm the broadcasted value D_k is computed by U_k using her/his secret a_k after passing two verifying equations for $k = 1, 2, \dots, n$ except i . Hence, if there is a participant U_m who broadcasts a wrong D_m to disturb the group session key establishing, he will be identified as a malicious participant. In other words, the proposed RID-AGKE protocol can identify malicious participants by using the confirmed computation technique. \square

6. Performance Analysis and Comparisons

For convenience to evaluate the computational cost, we focus on the time-consuming pairing-based operations as follows:

- (i) TG_e : the time of executing a bilinear map operation $e : G_1 \times G_1 \rightarrow G_2$;
- (ii) TG_{mul} : the time of executing a point scalar multiplication operation in G_1 ;
- (iii) TG_H : the time of executing a map-to-point hash function $H_1, H_2 : \{0, 1\}^* \rightarrow G_1$;
- (iv) T_{exp} : the time of executing a modular exponentiation operation over a finite field F_p , where p is a large prime number;
- (v) T_{inv} : the time of executing a modular multiplicative inverse operation over a finite field F_p , where p is a large prime number.

Here, we first analyze the computational cost of our protocol. In Round 1, $2TG_{mul}$ is required to compute (P_i, V_i) . In Round 2, each participant requires $3TG_e + 7TG_{mul} + 4TG_H$ to verify $(ID_{i+k}, j, P_{i+k}, V_{i+k})$ for $k \in \{-1, 1\}$ and to generate $(D_i, \alpha_i, \beta_i, \gamma_i)$. In the group key computation phase, $(3n-1)TG_e + nTG_{mul} + (n-1)T_{exp}$ is required to verify all $(ID_i, j, D_i, \alpha_i, \beta_i, \gamma_i)$ and to compute a group key SK_j .

Note that to evaluate $SK_j = e(a_i \cdot P_{i-1}, P_{pub})^n \cdot D_i^{n-1} \cdot D_{i+1}^{n-2} \cdot \dots \cdot D_{i-2}$ is required $TG_e + TG_{mul}$ since $SK_j = A_{i-1} \cdot A_i \cdot A_{i+1} \cdot \dots \cdot A_{i-2}$, where $A_{i-1} = e(a_i \cdot P_{i-1}, P_{pub})$, $A_i = A_{i-1} \cdot D_i$, $A_{i+1} = A_i \cdot D_{i+1}$, \dots , and $A_{i-2} = A_{i-3} \cdot D_{i-2}$. As a result, each participant requires $(3n+2)TG_e + (n+9)TG_{mul} + 4TG_H + (n-1)T_{exp}$ in our protocol.

In Table 1, we compare our RID-AGKE protocol with four previously proposed AGKE protocols which include Tseng's AGKE protocol [25], Choi et al.'s ID-AGKE protocol [26], Wu et al.'s ID-AGKE protocol [6], and Wu et al.'s RID-AGKE protocol [4] in terms of the public key setting, number of rounds, computational cost, and security properties. One recent non-ID-based and non-RID-based AGKE protocol with identifying malicious participants was proposed by Tseng [25]. Since Tseng's protocol is based on the ElGmal system [29], each participant must verify the other participants' certificates for participant authentication. It will increase the required computational costs for verifying certificates, besides $(8n-2)T_{exp} + (n+1)T_{inv}$. On the contrary, Choi et al.'s ID-AGKE [26], Wu et al.'s ID-AGKE [6], Wu et al.'s RID-AGKE [4], and our protocol rely on the IDPKS system [8] or the R-IDPKS system [14]. Thus, they need not manage and verify the participants' certificates. However, Choi et al.'s ID-AGKE [26] suffered from an insider colluding attack demonstrated by Wu and Tseng [27].

For Wu et al.'s ID-AGKE [6], Wu et al.'s RID-AGKE [4], and our protocol, they are provably secure and are able to resist malicious participants. It is easy to see that our protocol is more efficient than Wu et al.'s ID-AGKE [6] even though both protocols can identify malicious participants via confirmed computation approach. More importantly, Wu et al.'s ID-AGKE protocol [6] does not provide a solution to revoke the compromised or misbehaving user in the group. It is very serious because these revoked participants should not be allowed to establish a common key with other legal (nonrevoked) participants. In another aspect, Wu et al.'s RID-AGKE [4] is a three-round protocol and adopts explicit key confirmation approach to resist malicious participants.

Though their protocol can detect the existence of malicious participants, it cannot still identify malicious participant. Our proposed RID-AGKE is a two-round protocol and provides an active detection mechanism to identify malicious participants. According to Table 1, the advantage of our protocol is demonstrated.

7. Conclusions

In this paper, we have fused the Tseng-Tsai R-IDPKS system and a noninteractive confirmed computation technique to propose the first RID-AGKE protocol with identifying malicious participants. The framework and security notions for RID-AGKE protocols have been defined to formalize the possible threats and attacks. When compared with the recently proposed ID/RID-AGKE protocols resistant to malicious participants, our protocol has better performance and provides an active detection way to identify malicious participants. In the random oracle model and under two mathematical assumptions (CDH and DBDH), we have proven that the proposed protocol is a secure RID-AGKE protocol with forward secrecy and identifying malicious participants.

Notations

- e : A bilinear map, $e : G_1 \times G_1 \rightarrow G_2$, defined in Section 2.1
 s : The system private key, $s \in Z_q^*$
 P : A generator of group G_1
 P_{pub} : The system public key, $P_{pub} = s \cdot P$
 ID_i : The identity of participant U_i
 DID_i : The participant U_i 's initial private key
 $TID_{i,j}$: The participant U_i 's update key for period j
 $DID_{i,j}$: The participant U_i 's private key for period j ,
 $DID_{i,j} = DID_i + TID_{i,j}$
 H_1 : A map-to-point hash function,
 $H_1 : \{0, 1\}^* \rightarrow G_1$
 H_2 : A map-to-point hash function,
 $H_2 : \{0, 1\}^* \rightarrow G_1$
 H_3 : A hash function, $H_3 : \{0, 1\}^* \times G_1 \rightarrow Z_q$
 H_4 : A hash function, $H_4 : \{0, 1\}^* \times G_1^3 \rightarrow Z_q$

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors thank the referees for their valuable comments and constructive suggestions. This research was partially supported by Shenzhen Peacock Project of China (no. KQC201109020055A) and Shenzhen Strategic Emerging Industries Program of China (no. ZDSY20120613125016389).

References

- [1] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," in *Advances in Cryptology—EUROCRYPT '94*, vol. 950 of *Lecture Notes in Computer Science*, pp. 275–286, 1995.
- [2] J. Katz and J. S. Shin, "Modeling insider attacks on group key-exchange protocols," in *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS '05)*, pp. 180–189, November 2005.
- [3] T.-Y. Wu, Y.-M. Tseng, and C.-W. Yu, "A secure ID-Based authenticated group key exchange protocol resistant to insider attacks," *Journal of Information Science and Engineering*, vol. 27, no. 3, pp. 915–932, 2011.
- [4] T.-Y. Wu, Y.-M. Tseng, and T.-T. Tsai, "A revocable ID-based authenticated group key exchange protocol with resistant to malicious participants," *Computer Networks*, vol. 56, no. 12, pp. 2994–3006, 2012.
- [5] Y.-M. Tseng, "A robust multi-party key agreement protocol resistant to malicious participants," *The Computer Journal*, vol. 48, no. 4, pp. 480–487, 2005.
- [6] T.-Y. Wu and Y.-M. Tseng, "Towards ID-based authenticated group key exchange protocol with identifying malicious participants," *Informatica*, vol. 23, no. 2, pp. 315–334, 2012.
- [7] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Identity-Based Cryptosystems and Signature Schemes*, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, 1985.
- [8] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003, Preliminary version: in *Advances in Cryptology—CRYPTO '01*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, 2001.
- [9] A. Boldyreva, V. Goyal, and V. Kumart, "Identity-based encryption with efficient revocation," in *Proceedings of the 15th ACM conference on Computer and Communications Security (CCS '08)*, pp. 417–426, October 2008.
- [10] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," *Journal of Cryptology*, vol. 20, no. 3, pp. 265–294, 2007, Preliminary version: in *Advances in Cryptology—EUROCRYPT 2003*, vol. 2656 of *Lecture Notes in Computer Science*, pp. 255–271, 2003.
- [11] B. Libert and D. Vergnaud, "Adaptive-ID secure revocable identity-based encryption," in *Topics in Cryptology—CT-RSA 2009*, vol. 5473 of *Lecture Notes in Computer Science*, pp. 1–15, 2009.
- [12] J. H. Seo and K. Emura, "Revocable identity-based encryption revisited: security model and construction," in *Public-Key Cryptography—PKC 2013*, vol. 7778 of *Lecture Notes in Computer Science*, pp. 216–234, 2013.
- [13] J. H. Seo and K. Emura, "Efficient delegation of key generation and revocation functionalities in identity-based encryption," in *Topics in Cryptology—CT-RSA 2013*, vol. 7779 of *Lecture Notes in Computer Science*, pp. 343–358, 2013.
- [14] Y.-M. Tseng and T.-T. Tsai, "Efficient revocable ID-based encryption with a public channel," *The Computer Journal*, vol. 55, no. 4, pp. 475–486, 2012.
- [15] T.-T. Tsai, Y.-M. Tseng, and T.-Y. Wu, "A fully secure revocable ID-based encryption in the standard model," *Informatica*, vol. 23, no. 3, pp. 487–505, 2012.
- [16] T.-Y. Wu, T.-T. Tsai, and Y.-M. Tseng, "Revocable ID-based signature scheme with batch verifications," in *Proceedings of the*

8th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '12), pp. 49–54, July 2012.

- [17] T. T. Tsai, Y. M. Tseng, and T. Y. Wu, “Provably secure revocable ID-based signature in the standard model,” *Security and Communication Networks*, vol. 6, no. 10, pp. 1250–1260, 2013.
- [18] T.-Y. Wu, T.-T. Tsai, and Y.-M. Tseng, “A revocable ID-based signcryption scheme,” *Journal of Information Hiding and Multimedia Signal Processing*, vol. 3, no. 3, pp. 240–251, 2012.
- [19] M. Bellare and P. Rogaway, “Random oracles are practical: a paradigm for designing efficient protocols,” in *Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS '93)*, pp. 62–73, November 1993.
- [20] L. Chen, Z. Cheng, and N. P. Smart, “Identity-based key agreement protocols from pairings,” *International Journal of Information Security*, vol. 6, no. 4, pp. 213–241, 2007.
- [21] T.-Y. Wu and Y.-M. Tseng, “An ID-based mutual authentication and key exchange protocol for low-power mobile devices,” *The Computer Journal*, vol. 53, no. 7, pp. 1062–1070, 2010.
- [22] K. Y. Choi, J. Y. Hwang, and D. H. Lee, “Efficient ID-based group key agreement with bilinear maps,” in *Public Key Cryptography—PKC 2004*, vol. 2947 of *Lecture Notes in Computer Science*, pp. 130–144, 2004.
- [23] J. Katz and J. S. Shin, “Modeling insider attacks on group key-exchange protocols,” in *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS '05)*, pp. 180–189, November 2005.
- [24] R. Steinwandt and A. S. Corona, “Attribute-based group key establishment,” *Advances in Mathematics of Communications*, vol. 4, no. 3, pp. 381–398, 2010.
- [25] Y.-M. Tseng, “A communication-efficient and fault-tolerant conference-key agreement protocol with forward secrecy,” *Journal of Systems and Software*, vol. 80, no. 7, pp. 1091–1101, 2007.
- [26] K. Y. Choi, J. Y. Hwang, and D. H. Lee, “ID-based authenticated group key agreement secure against insider attacks,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E91-A, no. 7, pp. 1828–1830, 2008.
- [27] T. Y. Wu and Y. M. Tseng, “Comments on an ID-based authenticated group key agreement protocol with withstanding insider attacks,” *IEICE Transactions on Fundamentals*, vol. E92-A, no. 10, pp. 2638–2640, 2009.
- [28] R. Housley, W. Polk, W. Ford, and D. Solo, “Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile,” RFC 3280, IETF, Anaheim, Calif, USA, 2002.
- [29] T. ElGamal, “A public-key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

Research Article

Efficient and Provable Secure Pairing-Free Security-Mediated Identity-Based Identification Schemes

Ji-Jian Chin,¹ Syh-Yuan Tan,² Swee-Huay Heng,² and Raphael C.-W. Phan¹

¹ Faculty of Engineering, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia

² Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia

Correspondence should be addressed to Ji-Jian Chin; jjchin@mmu.edu.my

Received 13 March 2014; Accepted 17 April 2014; Published 26 May 2014

Academic Editor: Mirjana Ivanovic

Copyright © 2014 Ji-Jian Chin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Security-mediated cryptography was first introduced by Boneh et al. in 2001. The main motivation behind security-mediated cryptography was the capability to allow instant revocation of a user's secret key by necessitating the cooperation of a security mediator in any given transaction. Subsequently in 2003, Boneh et al. showed how to convert a RSA-based security-mediated encryption scheme from a traditional public key setting to an identity-based one, where certificates would no longer be required. Following these two pioneering papers, other cryptographic primitives that utilize a security-mediated approach began to surface. However, the security-mediated identity-based identification scheme (SM-IBI) was not introduced until Chin et al. in 2013 with a scheme built on bilinear pairings. In this paper, we improve on the efficiency results for SM-IBI schemes by proposing two schemes that are pairing-free and are based on well-studied complexity assumptions: the RSA and discrete logarithm assumptions.

1. Introduction

1.1. Background. Identification schemes allow one party, the prover, to prove itself to another party, the verifier, that it knows its secret key without revealing anything else about itself in the process. The main utilization of the identification primitive is to facilitate one-sided entity authentication and is conventionally deployed in access control mechanisms to facilitate resource control and distribution.

In traditional public key cryptography, certificates are used to ensure that a user's public key is legitimately bound to a particular user and cannot be replaced. This certificate is usually issued by a certificate authority. However, certificate management can become an issue when the number of users in a cryptosystem grows larger. One of the methods of mitigating this potentially costly problem is through the deployment of the cryptographic primitive in an identity-based setting introduced by Shamir [1].

Another issue that traditional public key cryptography faces is the revocation of user secret keys. This would necessarily involve the revocation of a user's certificate along with his public/secret key pair and is a costly operation.

Timeliness is also a factor, as the procedure to revoke a user's keys may be a (relatively) long one and creates additional load on the certificate authority, turning the revocation procedure into a potentially costly process exercise. All these compound the certificate management issue mentioned earlier.

In the identity-based setting, where certificates are not used, key revocation is conventionally done by tagging validity periods onto the user's identity-string as an extension. This also creates an issue of timeliness since revocation is only possible at the end of those validity dates. Without checking certificates, it is also difficult to check if a user is still valid or if his user secret key has been revoked already.

In [2], Boneh et al. proposed the initial groundwork for instant revocation of user keys and privileges, including the public key and certificates by introducing the concept of security-mediated cryptography. The idea of security-mediated cryptography is to necessitate the cooperation of a security mediator, a trusted third party, in any form of transaction that a user needs to use. For example, in an encryption scheme, a security mediator needs to lend his cooperation to the user in order for the user to decrypt a particular ciphertext. And for signatures, a security mediator

has to agree to cooperate with a signer in order to produce a valid signature.

Specifically, this is done by separating the user's secret key into two portions during the key generation. One portion of the key is given to the mediator while the other is given to the user. Therefore, the security mediator cooperates in the form of providing his portion of the secret key to be combined with the user's portion to create the full secret key for the transaction.

1.2. Related Work. Identification schemes were first introduced by Fiat and Shamir [3], while their identity-based counterparts, namely, identity-based identification schemes, were first formalized by Bellare et al. [4] and Kurosawa and Heng [5] independently. In recent years, there have been various advances in the area of identity-based identification, such as the introduction of identity-based identification schemes in the standard model [6, 7], hierarchical model [8–10], and certificateless model [11].

Following Boneh et al.'s initial work, Ding and Tsudik expanded on security-mediated cryptography to cover identity-based encryption using the RSA assumption [12]. Shortly thereafter, Libert and Quisquater proposed the first pairing-based security-mediated identity-based encryption schemes [13]. On the signature front, Cheng et al. proposed the first security-mediated identity-based signature [14].

Chow et al. [15] and Yap et al. [16] both extended security-mediated cryptography into the certificateless setting, where the issue of key escrow was addressed. In certificateless cryptography, first proposed by Al-Riyami and Paterson [17], the key generation center only produces half of the user secret key. The user then produces the other half of the user secret key to be combined into the full user secret key, thus securing their key from even the key generation center. However, the cryptographic primitives in the certificateless setting introduce more operational cost to the scheme and are known to be difficult to be proven secure [18].

1.3. Motivations and Contribution. In 2013, Chin et al. first combined the notion of security-mediated cryptography with identifications in the identity-based setting to propose the first security-mediated identity-based identification (SM-IBI) scheme [19]. In the paper, the authors provided the first formal definitions for SM-IBIs and also proposed a concrete construction. The motivation of the authors was to allow fast revocation of keys for identification schemes in the identity-based setting.

For SM-IBI schemes, a security mediator is required to participate in the identification protocol in order for a user to authenticate himself to a verifier. The security mediator can then hold a revocation list to identify which users' secret keys have been revoked and refuse participation for those users.

However the first concrete scheme that was proposed by the authors was built based on bilinear pairings. The pairing operation is widely known by cryptographers to be a costly operation; thus, it would be beneficial to construct pairing-free alternatives to facilitate more efficient running of the cryptographic primitive.

In this paper, we propose two pairing-free SM-IBI constructs as faster alternatives to the pairing-based scheme proposed by Chin et al. Our two schemes are constructed based on the RSA assumption and the discrete logarithm assumption, respectively. The RSA-based scheme, which we name as the GQ-SM-IBI, is constructed based on the Guillou Quisquater identification scheme constructed by Bellare and Palacio [20]. On the other hand, the discrete logarithm-based scheme, which we name as the BNN-SM-IBI, is constructed based on the BNN identity-based identification scheme proposed by Bellare et al. [4]. We provide security analysis for both schemes, proving them secure against impersonation under passive attacks if the RSA assumption and the discrete logarithm assumption hold and secure against impersonation under active and concurrent attacks if the one-more RSA inversion assumption and the one-more discrete logarithm assumption hold. Lastly, we provide an efficiency analysis, both theoretically and practically, and show that both schemes are significantly faster than Chin et al.'s pairing-based SM-IBI scheme.

The rest of the paper is organized as follows. We begin with some preliminaries and review the formal definitions and security model of SM-IBI schemes in Section 2. Then we show the construction and security analysis for the GQ-SM-IBI scheme in Section 3. This is followed by the construction and security analysis for the BNN-SM-IBI scheme in Section 4. In Section 5 we show the operational costs of both schemes as well as presenting our implementation results. Finally we conclude in Section 6.

2. Preliminaries

2.1. Discrete Logarithm Assumption. Let G be a cyclic group with prime order q and let g be a generator of G . The DL problem (DL) is defined as given a number $A = g^a$ in group G , output a .

Definition 1. The discrete logarithm assumption states that there exists no polynomial-time algorithm M that is able to (t_{DL}, ϵ_{DL}) -solve the discrete logarithm problem with nonnegligible probability such that

$$\Pr [M(G, q, g, A) = a] \geq \epsilon_{DL}. \quad (1)$$

2.2. The One-More Discrete Logarithm Assumption. The one-more discrete logarithm (OMDL) problem was first introduced by Bellare and Palacio [20] in their proof against impersonation under active and concurrent attacks for the standard Schnorr identification scheme. Later work that involves proving security of identification schemes based on discrete logarithms to be secure against active and concurrent attacks for discrete logarithm also makes use of this assumption such as [4, 21].

Let G be a finite cyclic group of order q and let g be a generator of G . Define an experiment E_{DL} where an adversary M is given a challenge oracle $CHALL$ that produces a random group element $W_i \in G$ when queried and a discrete log oracle $DLOG$, which provides the discrete log $w_i \in \mathbb{Z}_q$ corresponding to the query W_i where $g^{w_i} = W_i$. M wins

if after making i queries to $CHALL_{DL}$, M is able to output solutions to all i challenges with only $i - 1$ queries to $DLOG$, meaning M has to solve at least one instance of the discrete logarithm problem without relying on the discrete log oracle. E_{DL} returns 1 if M is successful and 0 otherwise.

Definition 2. The *OMDL* assumption states that there exists no polynomial-time algorithm M that is able to $(t_{OMDL}, q_{OMDL}, \epsilon_{OMDL})$ -solve the *OMDL* problem with non-negligible probability where

$$\Pr [E_{DL} (M (1^k, G, g, CHALL_{DL}, DLOG) = 1) \geq \epsilon_{OMDL}. \tag{2}$$

2.3. RSA Inversion (RSAI) Assumption. Given $(N, e, X) \xleftarrow{\$} k_{RSA}(1^k)$ compute Y such that $Y = X^d \pmod N$ where $ed = 1 \pmod{\phi(N)}$.

Definition 3. The *RSIAI* assumption states that there exists no polynomial-time algorithm M that is able to $(t_{RSIAI}, \epsilon_{RSIAI})$ -solve the *RSIAI* problem with nonnegligible probability such that

$$\Pr [M (1^k, N, e, X) = Y : Y = X^d \text{ and } ed = 1 \pmod{\phi(N)} \geq \epsilon_{RSIAI}. \tag{3}$$

2.4. One-More RSA Inversion (OMRSAI) Assumption. This is the interactive variant of the *RSIAI* problem first proposed by Bellare and Palacio [20] to prove security of the GQ identification scheme and is analogous to the *OMDL* assumption. This assumption is applied in the proof of security against active and concurrent attacks for RSA-based schemes.

Define an experiment E_{RSIAI} where an adversary M is given $(N, e) \xleftarrow{\$} k_{RSA}(1^k)$ as input and access to two oracles $CHALL$ and RSA . $CHALL$ on any input returns a random point W_i , while RSA on any input h will return h^d where $ed = 1 \pmod N$. M is required to compute the *RSIAI* solutions to all the target points W_0, \dots, W_n while using strictly less queries to the RSA oracle. In other words, M is required to find W_0^d, \dots, W_n^d while using the RSA oracle only $i < n$ times. E_{RSIAI} returns 1 if M is successful and 0 otherwise.

Definition 4. The *OMRSAI* assumption states that there exists no polynomial-time algorithm M that is able to $(t_{OMRSAI}, q_{OMRSAI}, \epsilon_{OMRSAI})$ -win the *OMRSAI* problem with nonnegligible probability where

$$\Pr [E_{RSIAI} (M (1^k, N, e, CHALL_{RSA}, RSA)) = 1] \geq \epsilon_{RSIAI}. \tag{4}$$

2.5. Definition of Security-Mediated IBI Schemes. In this section, we review the definition of SM-IBI schemes as defined by [19]. The definition follows closely to that of conventional IBI schemes, with the difference being that the prover segment is extended to encompass obtaining tokens from the security mediator. The SM-IBI scheme is defined as five probabilistic polynomial-time algorithms.

(i) *Setup.* It takes in the security parameter 1^k as input and outputs the system parameters $params$ along with the master secret key MSK .

(ii) *Extract.* Upon receiving a user's request for a key, it takes in $params$, MSK and a user's identity ID . Once the secret key is created, the PKG separates the key into two portions, one for the user, USK_{user} , and one for the security mediator, USK_{sem} , and returns the portions to the respective parties.

(iii) *Identification Protocol.* The identification protocol is an interactive protocol run by the 3 algorithms: *User-Prover* and *SEM-prover* on the prover side trying to authenticate himself to the *Verifier*. Both provers are used cooperatively in the interactive three-step canonical honest verifier zero knowledge proof of knowledge protocol with the verifier as follows.

(1) *User-Prover* initiates by sending his identity to ID to the *SEM-prover*. *SEM-prover* checks whether ID 's keys have been revoked and stops with an error code if true. If ID is legitimate then it generates and sends *SEM-COMMIT* to *User-Prover* who then combines *SEM-COMMIT* with his own *USER-COMMIT* to form *FULL-COMMIT* to send to *Verifier*.

(2) *Verifier* selects a random *CHALLENGE* and sends it to the *User-Prover*.

(3) *User-Prover* relays *CHALLENGE* to *SEM-Prover* and receives *SEM-RESPONSE* from *SEM-Prover* which it then combines with his own *USER-RESPONSE* to form *FULL-RESPONSE* and sends to the *Verifier*. The *Verifier* will choose to either accept or reject it.

2.6. Security Model for Security-Mediated IBI. Adversaries of SM-IBI follow the description of standard identification schemes: passive and active/concurrent attackers. However, the adversary for SM-IBI is able to query additionally partial conversation components, specifically the user's prover and the security mediator's prover besides the usual full prover query.

The security of SM-IBI schemes is modelled as a game played by an adversary I against a challenger C as follows.

(i) *Setup.* C runs *Setup*, creates the system parameters $params$ and passes them to I while keeping the master secret key MSK to itself.

(ii) *Phase 1.* This is the training phase. I is allowed to adaptively make the following queries to C .

(a) *User-Extract (ID).* C will run *Extract* but returns only the user's portion of the secret key to I .

(b) *SEM-Extract (ID).* C will run *Extract* but returns only the security mediator's portion of the secret key to I .

- (c) *Full-Extract (ID)*. C will run *Extract* and returns both the user's portion of the secret key and the security mediator's portion of the secret key to I .
- (d) *Identification Queries (ID)*. For passive adversaries, C will generate transcripts of valid conversations for I . For active/concurrent adversaries, C will act as the cheating prover, engaging I as the cheating verifier in conversations. I is able to issue any one of the following identification queries.
- (1) *SEM-Identification (ID)*. C runs the security mediator's half of the prover session.
 - (2) *User-Identification (ID)*. C runs the user's half of the prover session.
 - (3) *Full-Identification (ID)*. C combines both security mediator's and user's session to generate a full and valid conversation.

(iii) *Phase 2*. I will eventually output ID^* on which it wants to be challenged on and begins its role as the cheating prover for both security mediator and user prover sessions. C on the other hand assumes the role of the verifier. I wins the game if it manages to convince C to accept with nonnegligible probability.

We say a security-mediated IBI scheme Π is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -secure under passive or active/concurrent attacks if for any passive or active/concurrent Type-1 impersonator I who runs in time t_{SMIBI} , $\Pr[I \text{ can impersonate}] < \epsilon_{SMIBI}$, where I can make at most q_{SMIBI} full extract queries.

It is interesting to point out that extracting the security mediator's half of the secret key gives no information about the full user key and that neither security mediator's prover sessions nor user's prover sessions done alone will provide a valid conversation, but only the combined session will. This models the security requirement that any user cannot legitimately prove himself to a verifier without the security mediator's help.

3. GQ-SMIBI: RSA-Based Security-Mediated IBI Scheme

The GQ-SM-IBI scheme is derived from the GQ identification scheme proposed in [20] and is provably secure against passive attackers assuming the *RSAI* assumption and against active/concurrent attackers assuming the *OMRSAI* assumption.

The GQ-SM-IBI scheme is constructed as follows.

- (1) *Setup* (1^k). It takes in the security parameter 1^k , runs the key generation algorithm for RSA, and obtains an RSA instance, $(N, e, d) \xleftarrow{\$} k_{RSA}(1^k)$. It chooses a hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_N^*$ and publishes the system parameters $\text{mpk} = \langle N, e, H \rangle$. The master secret key $\text{msk} = d$ is kept secret.
- (2) *Extract* ($\text{mpk}, \text{msk}, \text{ID}$). It takes in $\text{mpk}, \text{msk} = s$, and ID as input. It calculates $H(\text{ID})$ and sets

$X_{FULL} = H(\text{ID})^d$. It further chooses $d_{SEM} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$, sets $d_{USER} = d - d_{SEM}$, and calculates $X_{USER} = H(\text{ID})^{d_{USER}}$ and $X_{SEM} = H(\text{ID})^{d_{SEM}}$. It gives X_{USER} to the user as its partial secret key and X_{SEM} to the security mediator as its partial private key and keeps X_{FULL} secret.

- (3) *Identification Protocol*. It is run by the *User-Prover* and *SEM-Prover* with *Verifier* as such.

(a) *User-Prover* sends its ID to *Sem-Prover* and chooses $y_{USER} \xleftarrow{\$} \mathbb{Z}_N^*$ and sets $Y_{USER} = y_{USER}^e$. *SEM-Prover* upon receiving ID checks if it is still valid and returns error if it is revoked. Otherwise, *SEM-Prover* chooses $y_{SEM} \xleftarrow{\$} \mathbb{Z}_N^*$ and sends $Y_{SEM} = y_{SEM}^e$ to *User-Prover*. *User-Prover* combines $Y_{FULL} = Y_{SEM} \times Y_{USER}$ and sends Y_{FULL} to *Verifier*.

(b) *Verifier* chooses a random challenge $c \xleftarrow{\$} \mathbb{Z}_{2^{l(k)}}$ where $l(\cdot)$ is a super-logarithmic challenge length such that $2^{l(k)} < e$. *Verifier* then sends it to *User-Prover*. *User-Prover* relays c to *SEM-Prover*.

(c) *SEM-Prover* calculates its half of the response $z_{SEM} = y_{SEM} X_{SEM}^c$ and sends it to *User-Prover*. *User-Prover* combines it with his response $z_{USER} = y_{USER} X_{USER}^c$ to create $z_{FULL} = z_{SEM} \times z_{USER}$ and sends it as a response to *Verifier*.

Verifier checks if $z_{FULL}^e = Y_{FULL} H(\text{ID})^c$ and accepts if yes; otherwise it outputs reject.

To check for completeness,

$$\begin{aligned} Z_{FULL}^e &= [(y_{SEM} X_{SEM}^c)(y_{USER} X_{USER}^c)]^e \\ &= [y_{SEM} y_{USER} (X_{SEM} X_{USER})^c]^e \\ &= y_{SEM}^e y_{USER}^e (H(\text{ID})^{(d_{SEM} + d_{USER})ec}) \\ &= Y_{SEM} Y_{USER} (H(\text{ID})^{dec}) \\ &= Y_{FULL} (H(\text{ID})^c). \end{aligned} \quad (5)$$

3.1. Security Analysis: Impersonation under Passive Attack

Theorem 5. *The GQ-SM-IBI scheme is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -secure against impersonation under passive attacks in the random oracle if the RSA Inversion Problem is $(t_{RSAI}, \epsilon_{RSAI})$ -hard where*

$$\epsilon_{SMIBI} \leq \sqrt{\epsilon_{RSAI} e (q_{SMIBI} + 1)} + \frac{1}{2^{l(k)}}. \quad (6)$$

Proof. Assume the GQ-SM-IBI scheme is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -breakable; then a simulator M that $(t_{RSAI}, \epsilon_{RSAI})$ -breaks the *RSAI* problem can be shown. M takes in input (N, e, y) and runs the impersonator I as a subroutine.

Without loss of generality, it can be assumed that any *SEM-Extract*, *User-Extract*, *SEM-Identification*, *User-Identification*, and *Full-Identification* queries are preceded by a *Create-User* query. To avoid collision and consistently respond to these queries, M maintains a list $L_H = \langle \text{ID}_i, Q_{\text{ID}_i}, f_{\text{ID}_i}, X_{\text{FULL}_{\text{ID}_i}}, X_{\text{SEM}_{\text{ID}_i}}, X_{\text{USER}_{\text{ID}_i}}, d_{\text{SEM}_{\text{ID}_i}}, d_{\text{USER}_{\text{ID}_i}} \rangle$ which is initially empty. The following shows how M simulates the environment and oracle queries for I .

- (1) *Setup*. M selects a hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_N^*$ and sets the system parameters as $\text{mpk} = \langle N, e, H \rangle$. It sends mpk to I .
- (2) *Create-User* (ID_i). M chooses $l \in \{1, \dots, q_H\}$ randomly and lets $\text{ID}_l = \text{ID}^*$ at this point. Whenever I makes a H query on ID_i , consider the following.

- (a) If $i = l$, M will choose $f_{\text{ID}^*} \xleftarrow{\$} \mathbb{Z}_N^*$ sets $Q_{\text{ID}^*} = \gamma f_{\text{ID}^*}^e$ and will return Q_{ID^*} to I . It adds $\langle \text{ID}_l, Q_{\text{ID}^*}, f_{\text{ID}^*}, X_{\text{FULL}_{\text{ID}^*}} = \perp, X_{\text{USER}_{\text{ID}^*}} = \perp, X_{\text{SEM}_{\text{ID}^*}} = \perp, d_{\text{SEM}_{\text{ID}^*}} = \perp, d_{\text{USER}_{\text{ID}^*}} = \text{bot} \rangle$ to L_H .
- (b) Otherwise, for $i \neq l$, M chooses $f_{\text{ID}_i} \xleftarrow{\$} \mathbb{Z}_q^*$, sets $Q_{\text{ID}_i} = f_{\text{ID}_i}^e$, and returns Q_{ID_i} to I . M also sets $X_{\text{FULL}_{\text{ID}_i}} = f_{\text{ID}_i}$, chooses $d_{\text{USER}_{\text{ID}^*}} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$, and sets $X_{\text{SEM}_{\text{ID}_i}} = Q_{\text{ID}_i}^{d_{\text{SEM}_{\text{ID}_i}}}$ as the SEM portion of the user private key and $X_{\text{USER}_{\text{ID}_i}} = f_{\text{ID}_i} Q_{\text{ID}_i}^{-d_{\text{USER}_{\text{ID}_i}}}$ as the user portion of the user private key. It can be seen that

$$\begin{aligned} X_{\text{SEM}_{\text{ID}_i}} \times X_{\text{USER}_{\text{ID}_i}} &= Q_{\text{ID}_i}^{d_{\text{SEM}_{\text{ID}_i}}} \times f_{\text{ID}_i} Q_{\text{ID}_i}^{-d_{\text{USER}_{\text{ID}_i}}} \\ &= f_{\text{ID}_i} \\ &= X_{\text{FULL}_{\text{ID}_i}}. \end{aligned} \quad (7)$$

M adds $\langle \text{ID}_i, Q_{\text{ID}_i}, f_{\text{ID}_i}, X_{\text{FULL}_{\text{ID}_i}}, X_{\text{USER}_{\text{ID}_i}}, X_{\text{SEM}_{\text{ID}_i}}, d_{\text{SEM}_{\text{ID}_i}}, d_{\text{USER}_{\text{ID}_i}} \rangle$ to L_H . M returns Q_{ID_i} to I .

- (3) *SEM-Extract* (ID_i). If $\text{ID}_i = \text{ID}^*$ and *USER-Extract* (ID^*) has been queried before, M aborts. If $\text{ID}_i = \text{ID}^*$, but *USER-Extract* (ID^*) has not been queried before, M chooses $d_{\text{SEM}_{\text{ID}^*}} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$ and sets $X_{\text{SEM}_{\text{ID}^*}} = Q_{\text{ID}^*}^{d_{\text{SEM}_{\text{ID}^*}}}$ as the SEM portion of the user private key, saves it in L_H , and returns it to I . Otherwise for all other $\text{ID}_i \neq \text{ID}^*$, M just retrieves $X_{\text{SEM}_{\text{ID}_i}}$ in L_H and returns it to I .
- (4) *USER-Extract* (ID_i). If $\text{ID}_i = \text{ID}^*$ and *USER-Extract* (ID^*) has been queried before, M aborts. If $\text{ID}_i = \text{ID}^*$, but *SEM-Extract* (ID^*) has not been queried before, M chooses $d_{\text{USER}_{\text{ID}^*}} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$ and sets $X_{\text{USER}_{\text{ID}^*}} = Q_{\text{ID}^*}^{d_{\text{USER}_{\text{ID}^*}}}$ as the USER portion of the user

private key, saves it in L_H , and returns it to I . Otherwise for all other $\text{ID}_i \neq \text{ID}^*$, M just retrieves $X_{\text{USER}_{\text{ID}_i}}$ in L_H and returns it to I .

- (5) *Identification* (ID_j , *session*). I will act as the cheating verifier to learn information from valid conversation transcripts from M . If $\text{ID}_i \neq \text{ID}^*$ M just retrieves ID_i 's entry in L_H and runs the identification protocol for either the SEM's interactions, the user's interactions, or both combined as in the protocol.

Otherwise, for $\text{ID}_j = \text{ID}^*$, M then creates a full valid transcript for each m th query by picking $z_{m,\text{USER}_{\text{ID}^*}}, z_{m,\text{SEM}_{\text{ID}^*}} \xleftarrow{\$} \mathbb{Z}_q^*$, and $c_{m,\text{ID}^*} \xleftarrow{\$} \mathbb{Z}_q^*$ and sets $Z_{m,\text{FULL}_{\text{ID}^*}} = z_{m,\text{USER}_{\text{ID}^*}} \times z_{m,\text{SEM}_{\text{ID}^*}}$. M also sets $Y_{m,\text{USER}_{\text{ID}^*}} = z_{m,\text{USER}_{\text{ID}^*}}^e Q_{\text{ID}^*}^{-(c_{m,\text{ID}^*}/2)}$, $Y_{m,\text{SEM}_{\text{ID}^*}} = z_{m,\text{SEM}_{\text{ID}^*}}^e Q_{\text{ID}^*}^{-(c_{m,\text{ID}^*}/2)}$, and $Y_{m,\text{FULL}_{\text{ID}^*}} = z_{m,\text{FULL}_{\text{ID}^*}}^e Q_{\text{ID}^*}^{-c_{m,\text{ID}^*}}$. M tags the session with *session* = m . If I queries

- (i) *SEM-Identification* (ID^* , m), M will return $\langle Y_{m,\text{SEM}_{\text{ID}^*}}, c_{m,\text{ID}^*}, z_{m,\text{SEM}_{\text{ID}^*}} \rangle$;
- (ii) *User-Identification* (ID^* , m), M will return $\langle Y_{m,\text{USER}_{\text{ID}^*}}, c_{m,\text{ID}^*}, z_{m,\text{USER}_{\text{ID}^*}} \rangle$;
- (iii) *Full-Identification* (ID^* , m), M will return $\langle Y_{m,\text{FULL}_{\text{ID}^*}}, c_{m,\text{ID}^*}, z_{m,\text{FULL}_{\text{ID}^*}} \rangle$.

If no session is specified for the query, M just returns the next session in sequence. One can see that *SEM-Identification* (ID^* , m) and *User-Identification* (ID^* , m) can be combined to create a full and valid transcript on session m :

$$\begin{aligned} Y_{m,\text{FULL}_{\text{ID}^*}} H(\text{ID}^*)^{c_{m,\text{ID}^*}} &= Y_{m,\text{SEM}_{\text{ID}^*}} Y_{m,\text{USER}_{\text{ID}^*}} Q_{\text{ID}^*}^{c_{m,\text{ID}^*}} \\ &= z_{m,\text{SEM}_{\text{ID}^*}}^e Q_{\text{ID}^*}^{-(c_{m,\text{ID}^*}/2)} z_{m,\text{USER}_{\text{ID}^*}}^e Q_{\text{ID}^*}^{-(c_{m,\text{ID}^*}/2)} Q_{\text{ID}^*}^{c_{m,\text{ID}^*}} \\ &= [z_{m,\text{SEM}_{\text{ID}^*}} z_{m,\text{USER}_{\text{ID}^*}}]^e Q_{\text{ID}^*}^{-c_{m,\text{ID}^*}} Q_{\text{ID}^*}^{c_{m,\text{ID}^*}} \\ &= z_{m,\text{SEM}_{\text{ID}^*}}^e z_{m,\text{USER}_{\text{ID}^*}}^e \\ &= z_{m,\text{FULL}_{\text{ID}^*}}^e. \end{aligned} \quad (8)$$

Eventually, I stops Phase 1 and outputs the challenge ID , ID^* on which it wishes to be challenged on. M checks if $\text{ID}^* = \text{ID}_l$ from L_H and aborts if not. Otherwise, M runs I now as a cheating prover on ID^* by obtaining its commitment, $Y_{\text{FULL}_{\text{ID}^*}}$, selecting a challenge $c_1 \xleftarrow{\$} \mathbb{Z}_q^*$ and obtaining the response $z_{\text{FULL}_{\text{ID}^*},1}$ from I . M then resets I to the step whereby I just sent $Y_{\text{FULL}_{\text{ID}^*}}$, selects a second challenge $c_2 \xleftarrow{\$} \mathbb{Z}_q^*$, and receives $z_{\text{FULL}_{\text{ID}^*},2}$ as response. It should hold that $z_{\text{FULL}_{\text{ID}^*},1}^e = Y_{\text{FULL}_{\text{ID}^*}} H(\text{ID})^{c_1}$ and $z_{\text{FULL}_{\text{ID}^*},2}^e = Y_{\text{FULL}_{\text{ID}^*}} H(\text{ID})^{c_2}$. One can then obtain $(z_{\text{FULL}_{\text{ID}^*},1}/z_{\text{FULL}_{\text{ID}^*},2})^e = H(\text{ID})^{c_1-c_2}$. Since e is a prime and

$-e < c_1 - c_2 < e$, $GCD(e, c_1 - c_2) = 1$. Use the extended Euclidean algorithm to obtain integers S and T such that $eS + (c_1 - c_2)T = 1$. It follows that

$$\begin{aligned}
 H(\text{ID}^*) &= Q_{\text{ID}^*}^{eS+(c_1-c_2)T} \pmod N \\
 &= (yf_{\text{ID}^*}^e)^{eS+(c_1-c_2)T} \pmod N \\
 &= (yf_{\text{ID}^*}^{e^2S})(yf_{\text{ID}^*}^{e(c_1-c_2)T}) \pmod N \\
 &= (yf_{\text{ID}^*}^{e^2S}) \left(\frac{z_{\text{FULL}_{\text{ID}^*,1}}}{z_{\text{FULL}_{\text{ID}^*,2}}} \right)^{eT} \pmod N \\
 &= \left[(yf_{\text{ID}^*}^{eS}) \left(\frac{z_{\text{FULL}_{\text{ID}^*,1}}}{z_{\text{FULL}_{\text{ID}^*,2}}} \right)^T \right]^e \pmod N.
 \end{aligned} \tag{9}$$

M then calculates the solution to the RSAI problem as follows:

$$\begin{aligned}
 yf_{\text{ID}^*}^e &= \left[(yf_{\text{ID}^*}^{eS}) \left(\frac{z_{\text{FULL}_{\text{ID}^*,1}}}{z_{\text{FULL}_{\text{ID}^*,2}}} \right)^T \right]^e \pmod N, \\
 y^d f_{\text{ID}^*} &= (yf_{\text{ID}^*}^{eS}) \left(\frac{z_{\text{FULL}_{\text{ID}^*,1}}}{z_{\text{FULL}_{\text{ID}^*,2}}} \right)^T \pmod N, \\
 y^d &= \frac{(yf_{\text{ID}^*}^{eS}) (z_{\text{FULL}_{\text{ID}^*,1}}/z_{\text{FULL}_{\text{ID}^*,2}})^T}{f_{\text{ID}^*}} \pmod N, \\
 y^d &= (yf_{\text{ID}^*}^{eS-1}) \left(\frac{z_{\text{FULL}_{\text{ID}^*,1}}}{z_{\text{FULL}_{\text{ID}^*,2}}} \right)^T \pmod N.
 \end{aligned} \tag{10}$$

It remains to calculate the probability of M solving the RSAI problem and winning the game. The probability of M successfully extracting two valid conversation transcripts from I is bounded by $(\epsilon_{\text{IBI}} - (1/2^{l(k)}))^2$ as given by the reset lemma [20]:

$$\begin{aligned}
 &\Pr [M \text{ wins RSAI}] \\
 &= \Pr [M \text{ computes } g^{ab} \wedge \neg \text{abort}] \\
 &= \Pr [M \text{ computes } g^{ab} \mid \neg \text{abort}] \Pr [\neg \text{abort}] \\
 &\epsilon_{\text{RSAI}} \geq \left(\epsilon_{\text{SMIBI}} - \frac{1}{2^{l(k)}} \right)^2 \Pr [\neg \text{abort}].
 \end{aligned} \tag{11}$$

Finally calculate $\Pr[\neg \text{abort}]$. Let δ be the probability that I issues both a *SEM-Extract* and a *USER-Extract* query on ID^* and that I makes a total of q_{SMIBI} of such queries. The probability of M answering all the extraction queries is δ_{SMIBI}^q . In Phase 2, the probability of M not aborting is if I outputs the challenge identity ID^* that was not queried before. This is given by the probability $1 - \delta$. Compiling them, the probability of M not aborting is $\delta_{\text{SMIBI}}^q (1 - \delta)$. This probability is maximised at $\delta_{\text{opt}} = 1 - (1/(q_{\text{SMIBI}} + 1))$. Using δ_{opt} , the probability that M does not abort is at least $1/(2^{l(k)}(q_{\text{SMIBI}} + 1))$ because the value $(1 - (1/(q_{\text{SMIBI}} + 1)))^{q_{\text{SMIBI}}}$

approaches $1/e$ for large q_{SMIBI} . Therefore, the advantage of M , ϵ_{RSAI} , and the bound of the simulation are given as follows:

$$\begin{aligned}
 \epsilon_{\text{RSAI}} &\geq \left(\epsilon_{\text{SMIBI}} - \frac{1}{2^{l(k)}} \right)^2 \frac{1}{(q_{\text{SMIBI}} + 1)} \\
 \epsilon_{\text{RSAI}} (q_{\text{SMIBI}} + 1) &\geq \left(\epsilon_{\text{SMIBI}} - \frac{1}{2^{l(k)}} \right)^2 \\
 \epsilon_{\text{SMIBI}} &\leq \sqrt{\epsilon_{\text{RSAI}} (q_{\text{SMIBI}} + 1)} + \frac{1}{2^{l(k)}}.
 \end{aligned} \tag{12}$$

□

3.2. Security Analysis: Impersonation under Active/Concurrent Attack

Theorem 6. *The GQ-SM-IBI scheme is $(t_{\text{SMIBI}}, q_{\text{SMIBI}}, \epsilon_{\text{SMIBI}})$ -secure against impersonation under active/concurrent attacks in the random oracle if the OMCDH Problem is $(t_{\text{OMRSAI}}, q_{\text{OMRSAI}}, \epsilon_{\text{OMRSAI}})$ -hard where*

$$\epsilon_{\text{SMIBI}} \leq \sqrt{\epsilon_{\text{OMRSAI}} e (q_{\text{SMIBI}} + 1)} + \frac{1}{2^{l(k)}}. \tag{13}$$

Proof. Assume the GQ-SM-IBI scheme is $(t_{\text{SMIBI}}, q_{\text{SMIBI}}, \epsilon_{\text{SMIBI}})$ -breakable; then a simulator M that $(t_{\text{OMRSAI}}, q_{\text{OMRSAI}}, \epsilon_{\text{OMRSAI}})$ -breaks the OMRSAI problem can be shown. M takes in input (N, e) , is given access to *CHALL* and *RSA* oracles, and runs the impersonator I as a subroutine. Any *SEM-Extract*, *User-Extract*, *SEM-Identification*, *User-Identification*, and *Full-Identification* queries are preceded by a *Create-User* query. M maintains a list $L_H = \langle \text{ID}_i, Q_{\text{ID}_i}, f_{\text{ID}_i}, X_{\text{FULL}_{\text{ID}_i}}, X_{\text{SEM}_{\text{ID}_i}}, X_{\text{USER}_{\text{ID}_i}}, d_{\text{SEM}_{\text{ID}_i}}, d_{\text{USER}_{\text{ID}_i}} \rangle$ which is initially empty. The following shows how M simulates the environment and oracle queries for I .

(1) *Create-User* (ID_i). M chooses $l \in \{1, \dots, q_H\}$ randomly and lets $\text{ID}_i = \text{ID}^*$ at this point. Whenever I makes a *H* query on ID_i , consider the following.

(i) If $i = l$, M will choose $f_{\text{ID}^*} \xleftarrow{\$} \mathbb{Z}_N^*$, queries *CHALL* for w_0 , and sets $Q_{\text{ID}^*} = w_0 f_{\text{ID}^*}^e$ and will return Q_{ID^*} to I . It adds $\langle \text{ID}_i, Q_{\text{ID}^*}, f_{\text{ID}^*}, X_{\text{FULL}_{\text{ID}^*}} = \perp, X_{\text{USER}_{\text{ID}^*}} = \perp, X_{\text{SEM}_{\text{ID}^*}} = \perp, d_{\text{SEM}_{\text{ID}^*}} = \perp, d_{\text{USER}_{\text{ID}^*}} = \perp \rangle$ to L_H .

(ii) Otherwise, for $i \neq l$, M chooses $f_{\text{ID}_i} \xleftarrow{\$} \mathbb{Z}_q^*$, sets $Q_{\text{ID}_i} = f_{\text{ID}_i}^e$, and returns Q_{ID_i} to I . M also sets $X_{\text{FULL}_{\text{ID}_i}} = f_{\text{ID}_i}$, chooses $d_{\text{USER}_{\text{ID}^*}} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$, and sets $X_{\text{SEM}_{\text{ID}_i}} = Q_{\text{ID}_i}^{d_{\text{SEM}_{\text{ID}_i}}}$ as the SEM portion of the user private key and $X_{\text{USER}_{\text{ID}_i}} = f_{\text{ID}_i}^{-d_{\text{USER}_{\text{ID}_i}}}$ as

the user portion of the user private key. It can be seen that

$$\begin{aligned} X_{SEM_{ID_i}} \times X_{USER_{ID_i}} &= Q_{ID_i}^{d_{SEM_{ID_i}}} \times f_{ID_i}^{-d_{SEM_{ID_i}}} \\ &= f_{ID_i} \\ &= X_{FULL_{ID_i}}. \end{aligned} \quad (14)$$

M adds $\langle ID_i, Q_{ID_i}, f_{ID_i}, X_{FULL_{ID_i}}, X_{USER_{ID_i}}, X_{SEM_{ID_i}}, d_{SEM_{ID_i}}, d_{USER_{ID_i}} \rangle$ to L_H . M returns Q_{ID_i} to I .

- (2) *SEM-Extract* (ID_i). If $ID_i = ID^*$ and *USER-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *USER-Extract* (ID^*) has not been queried before, M chooses $d_{SEM_{ID^*}} \xleftarrow{\$} Z_{\phi(N)}^*$ and sets $X_{SEM_{ID^*}} = Q_{ID^*}^{d_{SEM_{ID^*}}}$ as the SEM portion of the user private key, saves it in L_H , and returns it to I . Otherwise for all other $ID_i \neq ID^*$, M just retrieves $X_{SEM_{ID_i}}$ in L_H and returns it to I .
- (3) *USER-Extract* (ID_i). If $ID_i = ID^*$ and *SEM-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *SEM-Extract* (ID^*) has not been queried before, M chooses $d_{USER_{ID^*}} \xleftarrow{\$} Z_{\phi(N)}^*$ and sets $X_{USER_{ID^*}} = Q_{ID^*}^{d_{USER_{ID^*}}}$ as the USER portion of the user private key, saves it in L_H , and returns it to I . Otherwise for all other $ID_i \neq ID^*$, M just retrieves $X_{USER_{ID_i}}$ in L_H and returns it to I .
- (4) *Identification* (ID_i , *session*). I will act as the cheating verifier to learn information from valid conversation interactions from M . If $ID_i \neq ID^*$ M just retrieves ID_i 's entry in L_H and runs the identification protocol for either the SEM's interactions, the user's interactions, or both combined as in the protocol.

Otherwise, for $ID^* = ID^*$, M then creates a full valid conversation for each m th query by querying *CHALL* for W_m and setting $Y_{FULL_{ID^*}} = W_m$. Additionally,

M chooses $Y_{m,SEM_{ID^*}} \xleftarrow{\$} Z_N^*$ and sets $Y_{m,USER_{ID^*}} = W_m/Y_{m,SEM_{ID^*}}$. Upon receiving c_{m,ID^*} from I , M will query $RSA(W_m(W_0 f_{ID^*}^e)^{c_{m,ID^*}})$ to receive $V_m = (W_m(W_0 f_{ID^*}^e)^{c_{m,ID^*}})^d$ and sets $z_{FULL_{ID^*}} = V_m$. M then chooses $z_{m,SEM_{ID^*}} \xleftarrow{\$} Z_N^*$ and then sets $z_{m,USER_{ID^*}} = V_m/z_{m,SEM_{ID^*}}$. M tags the session with *session* = m .

- (i) If I queries *SEM-Identification* (ID^* , m), M will commit $Y_{m,SEM_{ID^*}}$ and respond with $z_{m,SEM_{ID^*}}$ upon receiving c_{m,ID^*} .
- (ii) If I queries *User-Identification* (ID^* , m), M will commit $Y_{m,USER_{ID^*}}$ and respond with $z_{m,USER_{ID^*}}$ upon receiving c_{m,ID^*} .

- (iii) If I queries *Full-Identification* (ID^* , m), M will commit $Y_{m,FULL_{ID^*}}$ and respond with $z_{m,FULL_{ID^*}}$ upon receiving c_{m,ID^*} .

If no session is specified for the query, M just returns the next session in sequence. One can see that *SEM-Identification* (ID^* , m) and *User-Identification* (ID^* , m) can be combined to create a full and valid conversation on session m :

$$\begin{aligned} Y_{m,FULL_{ID^*}} H(ID^*)^{c_{m,ID^*}} &= Y_{m,SEM_{ID^*}} Y_{m,USER_{ID^*}} Q_{ID^*}^{c_{m,ID^*}} \\ &= Y_{m,SEM_{ID^*}} \frac{W_m}{Y_{m,SEM_{ID^*}}} (W_0 f_{ID^*}^e)^{c_{m,ID^*}} \\ &= W_m (W_0 f_{ID^*}^e)^{c_{m,ID^*}} \\ &= [W_m (W_0 f_{ID^*}^e)^{c_{m,ID^*}}]^{de} \\ &= \left[[W_m (W_0 f_{ID^*}^e)^{c_{m,ID^*}}]^d \right]^e \\ &= [V_m]^e \\ &= \left(\frac{V_m}{z_{m,SEM_{ID^*}}} \right)^e (z_{m,SEM_{ID^*}})^e \\ &= z_{m,SEM_{ID^*}}^e z_{m,USER_{ID^*}}^e \\ &= z_{m,FULL_{ID^*}}^e. \end{aligned} \quad (15)$$

Eventually, I stops Phase 1 and outputs the challenge ID , ID^* on which it wishes to be challenged on. M checks if $ID^* = ID_l$ from L_H and aborts if not. Otherwise, M runs I now as a cheating prover on ID^* . M runs by obtaining its commitment, $Y_{FULL_{ID^*}}$, selects a challenge $c_1 \xleftarrow{\$} Z_q^*$, and obtains the response $z_{FULL_{ID^*},1}$ from I . M then resets I to the step whereby I just sent $Y_{FULL_{ID^*}}$, selects a second challenge $c_2 \xleftarrow{\$} Z_q^*$, and receives $z_{FULL_{ID^*},2}$ as response. It should hold that $z_{FULL_{ID^*},1}^e = Y_{FULL_{ID^*}} H(ID)^{c_1}$ and $z_{FULL_{ID^*},2}^e = Y_{FULL_{ID^*}} H(ID)^{c_2}$. One can then obtain $(z_{FULL_{ID^*},1}/z_{FULL_{ID^*},2})^e = H(ID)^{c_1-c_2}$. Since e is a prime and $-e < c_1 - c_2 < e$, $GCD(e, c_1 - c_2) = 1$. Use the extended Euclidean algorithm to obtain integers S and T such that $eS + (c_1 - c_2)T = 1$. It follows that

$$\begin{aligned} H(ID^*) &= Q_{ID^*}^{eS+(c_1-c_2)T} \text{ mod } N \\ &= (w_0 f_{ID^*}^e)^{eS+(c_1-c_2)T} \text{ mod } N \\ &= \left(w_0 f_{ID^*}^{e^2 S} \right) \left(w_0 f_{ID^*}^{e(c_1-c_2)T} \right) \text{ mod } N \end{aligned}$$

$$\begin{aligned}
&= \left(w_0 f_{ID^*}^{e^2 S} \right) \left(\frac{z_{FULL_{ID^*},1}}{z_{FULL_{ID^*},2}} \right)^{eT} \text{ mod } N \\
&= \left[\left(w_0 f_{ID^*}^{eS} \right) \left(\frac{z_{FULL_{ID^*},1}}{z_{FULL_{ID^*},2}} \right)^T \right]^e \text{ mod } N.
\end{aligned} \tag{16}$$

M then calculates the solution to the RSAI problem as follows:

$$\begin{aligned}
w_0 f_{ID^*}^e &= \left[\left(w_0 f_{ID^*}^{eS} \right) \left(\frac{z_{FULL_{ID^*},1}}{z_{FULL_{ID^*},2}} \right)^T \right]^e \text{ mod } N, \\
w_0^d f_{ID^*} &= \left(w_0 f_{ID^*}^{eS} \right) \left(\frac{z_{FULL_{ID^*},1}}{z_{FULL_{ID^*},2}} \right)^T \text{ mod } N, \\
w_0^d &= \frac{\left(w_0 f_{ID^*}^{eS} \right) \left(z_{FULL_{ID^*},1} / z_{FULL_{ID^*},2} \right)^T}{f_{ID^*}} \text{ mod } N, \\
w_0^d &= \left(w_0 f_{ID^*}^{eS-1} \right) \left(\frac{z_{FULL_{ID^*},1}}{z_{FULL_{ID^*},2}} \right)^T \text{ mod } N.
\end{aligned} \tag{17}$$

M proceeds to calculate the solutions to the other challenges as follows:

$$w_j = V_j(w_0 f_{ID^*})^{-cm_{ID^*}}. \tag{18}$$

The probability study for the simulation above is similar to that of the impersonation under passive attack game and is therefore omitted. \square

4. BNN-SM-IBI: An DL-Based Security-Mediated IBI Scheme

The BNN-SM-IBI scheme is derived from the BNN-IBI scheme proposed in the work of [4] and is provably secure against passive attackers assuming the DL assumption and against active/concurrent attackers assuming the OMDL assumption.

The BNN-SM-IBI scheme is constructed as follows.

- (1) *Setup* (1^k). It takes in the security parameter 1^k . It randomly selects $x \xleftarrow{\$} \mathbb{Z}_q$, a generator $g \xleftarrow{\$} G$ and computes $X = g^x$. Setup also chooses $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q$ and publishes the system parameters $\text{mpk} = \langle G, q, g, X, H \rangle$. The master private key $\text{msk} = x$ is kept secret.
- (2) *Extract* ($\text{mpk}, \text{msk}, \text{ID}$). It takes in $\text{mpk}, \text{msk} = s$, and ID as input. It calculates $H(\text{ID})$, picks $r_{FULL}, r_{SEM}, s_{SEM} \xleftarrow{\$} \mathbb{Z}_q$, calculates $s_{FULL} = r_{FULL} + xH(\text{ID})$ and $R_{FULL} = g^{r_{FULL}}$, and sets the full user private key as $\text{usk}_{FULL} = \langle s_{FULL}, R_{FULL} \rangle$. Additionally, it sets $s_{USER} = s_{FULL} - s_{SEM}$, $r_{USER} = r_{FULL} - r_{SEM}$, $R_{USER} = g^{r_{USER}}$, $R_{SEM} = g^{r_{SEM}}$. It then sets $\text{usk}_{USER} = \langle s_{USER}, R_{USER} \rangle$ and $\text{usk}_{SEM} = \langle s_{SEM}, R_{SEM} \rangle$. It gives

usk_{USER} to the user as its partial private key and usk_{SEM} to the security mediator as its partial private key and keeps usk_{FULL} secret.

- (3) *Identification Protocol*. It is run by the *User-Prover* and *SEM-Prover* with *Verifier* as such.
 - (a) *User-Prover* sends its ID to *Sem-Prover*, chooses $y_{USER} \xleftarrow{\$} \mathbb{Z}_q$, and sets $Y_{USER} = g^{y_{USER}}$ and $S_{User} = g^{s_{USER}}$. *SEM-Prover* upon receiving ID checks if it is still valid and returns error if it is revoked. Otherwise, *SEM-Prover* chooses $y_{SEM} \xleftarrow{\$} \mathbb{Z}_q$ and sends $Y_{SEM} = g^{y_{SEM}}$, $S_{SEM} = g^{s_{SEM}}$, and R_{SEM} to *User-Prover*. *User-Prover* combines $Y_{FULL} = Y_{SEM} \times Y_{USER}$, $S_{FULL} = S_{SEM} \times S_{USER}$, $R_{FULL} = R_{SEM} \times R_{USER}$ and sends Y_{FULL}, S_{FULL} , and R_{FULL} to *Verifier*.
 - (b) *Verifier* chooses a random challenge $c \xleftarrow{\$} \mathbb{Z}_q$ and sends it to *User-Prover*. *User-Prover* relays c to *SEM-Prover*.
 - (c) *SEM-Prover* calculates its half of the response $z_{SEM} = y_{SEM} + cs_{SEM}$ and sends it to *User-Prover*. *User-Prover* combines it with his response $z_{USER} = y_{USER} + cs_{USER}$ to create $z_{FULL} = z_{SEM} + z_{USER}$ and sends it as a response to *Verifier*.

Verifier checks if $g^{z_{FULL}} = Y_{FULL} R_{FULL}^c X^{H(\text{ID})c}$ and accepts if yes; otherwise it outputs reject.

To check for completeness,

$$\begin{aligned}
g^{z_{FULL}} &= g^{z_{SEM} + z_{USER}} \\
&= g^{y_{SEM} + cs_{SEM} + y_{USER} + cs_{USER}} \\
&= g^{y_{SEM} + y_{USER} + c(s_{SEM} + s_{USER})} \\
&= g^{y_{FULL} + c(s_{FULL})} \\
&= g^{y_{FULL} + c(r_{FULL} + xH(\text{ID}))} \\
&= g^{y_{FULL}} g^{r_{FULL}c} g^{xH(\text{ID})c} \\
&= Y_{FULL} R_{FULL}^c X^{H(\text{ID})c}.
\end{aligned} \tag{19}$$

4.1. Security Analysis: Impersonation under Passive Attack

Theorem 7. *The BNN-SM-IBI scheme is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -secure against impersonation under passive attacks in the random oracle if the DL problem is (t_{DL}, ϵ_{DL}) -hard where*

$$\epsilon_{SMIBI} \leq \sqrt{\epsilon_{DL} e (q_{SMIBI} + 1)} + \frac{1}{q}. \tag{20}$$

Proof. Assume the BNN-SM-IBI scheme is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -breakable; then a simulator M that (t_{DL}, ϵ_{DL}) -breaks the DL problem can be shown. M takes in input

($g, A = g^a$) and runs the impersonator I as a subroutine. Without loss of generality, it can be assumed that any *SEM-Extract*, *User-Extract*, *SEM-Identification*, *User-Identification*, and *Full-Identification* queries are preceded by a *Create-User* query. To avoid collision and consistently respond to these queries, M maintains a list $L_H = \langle ID_i, Q_{ID_i}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, r_{FULL_{ID_i}}, r_{SEM_{ID_i}}, r_{USER_{ID_i}}, R_{FULL_{ID_i}}, R_{SEM_{ID_i}}, R_{USER_{ID_i}} \rangle$ which is initially empty. The following shows how M simulates the environment and oracle queries for I .

- (1) *Setup*. M sets the system parameters as $mpk = \langle G, q, g, X = g^x, H \rangle$ and keeps master private key x secret. It sends mpk to I .
- (2) *Create-User* (ID_i). M chooses $l \in \{1, \dots, q_H\}$ randomly and lets $ID_i = ID^*$ at this point. Whenever I makes a H query on ID_i , consider the following.

(i) If $i = l$, M chooses $Q_{ID^*} \xleftarrow{\$} \mathbb{Z}_q$ and sets $S_{FULL_{ID^*}} = A$ and $R_{FULL_{ID^*}} = AX^{Q_{ID^*}}$ and returns Q_{ID^*} to I . M adds $\langle ID_i, Q_{ID^*}, S_{FULL_{ID^*}}, S_{SEM_{ID^*}} = \perp, S_{USER_{ID^*}}, S_{FULL_{ID^*}} = \perp, S_{SEM_{ID^*}}, S_{USER_{ID^*}} = \perp, r_{FULL_{ID^*}} = \perp, r_{SEM_{ID^*}} = \perp, r_{USER_{ID^*}} = \perp, R_{FULL_{ID^*}}, R_{SEM_{ID^*}} = \perp, R_{USER_{ID^*}} = \perp \rangle$ to L_H .

(ii) Otherwise, for $i \neq l$, M chooses $Q_{ID_i} \xleftarrow{\$} \mathbb{Z}_q^*$ and returns Q_{ID_i} . It then chooses $r_{FULL_{ID_i}}, r_{SEM_{ID_i}}, s_{SEM_{ID_i}} \xleftarrow{\$} \mathbb{Z}_q^*$, calculates $s_{FULL_{ID_i}} = r_{FULL_{ID_i}} + xQ$, and sets $r_{USER_{ID_i}} = r_{FULL_{ID_i}} - r_{SEM_{ID_i}}$ and $s_{USER_{ID_i}} = s_{FULL_{ID_i}} - s_{SEM_{ID_i}}$. M adds $\langle ID_i, Q_{ID_i}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, r_{FULL_{ID_i}}, r_{SEM_{ID_i}}, r_{USER_{ID_i}}, R_{FULL_{ID_i}}, R_{SEM_{ID_i}}, R_{USER_{ID_i}} \rangle$ to L_H .

- (3) *SEM-Extract* (ID_i). If $ID_i = ID^*$ and *USER-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *USER-Extract* (ID^*) has not been queried before, M will choose $r_{SEM_{ID^*}}, s_{SEM_{ID^*}} \xleftarrow{\$} \mathbb{Z}_q$ and sets $R_{SEM_{ID^*}} = g^{r_{SEM_{ID^*}}}$ and $S_{SEM_{ID^*}} = g^{s_{SEM_{ID^*}}}$. M also sets $S_{USER_{ID^*}} = S_{FULL_{ID^*}} / S_{SEM_{ID^*}}$ and $R_{USER_{ID^*}} = R_{FULL_{ID^*}} / R_{SEM_{ID^*}}$, saves these values in L_H , and returns them to I . These values will be used in the event of an *Identification* query later on. Otherwise for all other $ID_i \neq ID^*$, M just retrieves $\langle s_{SEM_{ID_i}}, R_{SEM_{ID_i}} \rangle$ in L_H and returns it to I .
- (4) *USER-Extract* (ID_i). If $ID_i = ID^*$ and *SEM-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *SEM-Extract* (ID^*) has not been queried before, M will choose $r_{USER_{ID^*}}, s_{USER_{ID^*}} \xleftarrow{\$} \mathbb{Z}_q$ and sets $R_{USER_{ID^*}} = g^{r_{USER_{ID^*}}}$ and $S_{USER_{ID^*}} = g^{s_{USER_{ID^*}}}$. M also sets $S_{SEM_{ID^*}} = S_{FULL_{ID^*}} / S_{USER_{ID^*}}$ and $R_{SEM_{ID^*}} = R_{FULL_{ID^*}} / R_{USER_{ID^*}}$, saves these values in L_H , and returns them to I . These values will be

used in the event of an *Identification* query later on. Otherwise for all other $ID_i \neq ID^*$, M just retrieves $\langle s_{USER_{ID_i}}, R_{USER_{ID_i}} \rangle$ in L_H and returns it to I .

- (5) *Identification* (ID_j , *session*). I will act as the cheating verifier to learn information from valid conversation transcripts from M . If $ID_j \neq ID^*$, M just retrieves ID_j 's entry in L_H and runs the identification protocol for either the *SEM*'s interactions, the user's interactions, or both combined as in the protocol.

Otherwise, for $ID_j = ID^*$, M then creates a full valid transcript for each m th query by picking $z_{m,FULL_{ID^*}}, z_{m,SEM_{ID^*}}, c_{m,ID^*} \xleftarrow{\$} \mathbb{Z}_q$. M retrieves $S_{FULL_{ID^*}}, R_{FULL_{ID^*}}$ and sets $Y_{m,FULL_{ID^*}} = g^{z_{m,FULL_{ID^*}}}$. Additionally, M chooses $z_{m,SEM_{ID^*}} \xleftarrow{\$} \mathbb{Z}_q$, calculates $z_{m,USER_{ID^*}} = z_{m,FULL_{ID^*}} - z_{m,SEM_{ID^*}}$, and retrieves $S_{SEM_{ID^*}}, R_{SEM_{ID^*}}$ and $S_{USER_{ID^*}}, R_{USER_{ID^*}}$ from L_H as well. M then sets $Y_{m,SEM_{ID^*}} = g^{z_{m,SEM_{ID^*}}} S_{SEM_{ID^*}}^{c_{m,ID^*}}$ and $Y_{m,USER_{ID^*}} = g^{z_{m,USER_{ID^*}}} S_{USER_{ID^*}}^{c_{m,ID^*}}$. M tags the session with *session* = m . If I queries

- (i) *SEM-Identification* (ID^* , m), M will return $\langle Y_{m,SEM_{ID^*}}, S_{SEM_{ID^*}}, R_{SEM_{ID^*}}, c_{m,ID^*}, z_{m,SEM_{ID^*}} \rangle$;
- (ii) *User-Identification* (ID^* , m), M will return $\langle Y_{m,USER_{ID^*}}, S_{USER_{ID^*}}, R_{USER_{ID^*}}, c_{m,ID^*}, z_{m,USER_{ID^*}} \rangle$;
- (iii) *Full-Identification* (ID^* , m), M will return $\langle Y_{m,FULL_{ID^*}}, S_{FULL_{ID^*}}, R_{FULL_{ID^*}}, c_{m,ID^*}, z_{m,FULL_{ID^*}} \rangle$.

If no session is specified for the query, M just returns the next session in sequence. One can see that *SEM-Identification* (ID^* , m) and *User-Identification* (ID^* , m) can be combined to create a full and valid transcript on session m :

$$\begin{aligned}
& Y_{m,FULL_{ID^*}} R_{FULL_{ID^*}}^{c_{m,ID^*}} X^{H(ID^*)c_{m,ID^*}} \\
&= Y_{m,USER_{ID^*}} Y_{m,SEM_{ID^*}} R_{USER_{ID^*}}^{c_{m,ID^*}} R_{SEM_{ID^*}}^{c_{m,ID^*}} \\
&\quad \times X^{H(ID^*)c_{m,ID^*}} \\
&= \left(g^{z_{m,USER_{ID^*}}} S_{USER_{ID^*}}^{-c_{m,ID^*}} g^{z_{m,SEM_{ID^*}}} \right) S_{SEM_{ID^*}}^{-c_{m,ID^*}} \\
&\quad \times \left(R_{USER_{ID^*}} R_{SEM_{ID^*}} \right)^{c_{m,ID^*}} X^{H(ID^*)c_{m,ID^*}} \\
&= g^{z_{m,USER_{ID^*}}} g^{z_{m,SEM_{ID^*}}} \left(S_{USER_{ID^*}} S_{SEM_{ID^*}} \right)^{-c_{m,ID^*}} \\
&\quad \times \left(R_{USER_{ID^*}} R_{SEM_{ID^*}} \right)^{c_{m,ID^*}} X^{H(ID^*)c_{m,ID^*}} \\
&= g^{z_{m,USER_{ID^*}}} g^{z_{m,SEM_{ID^*}}} \left(\frac{S_{FULL_{ID^*}}}{S_{SEM_{ID^*}}} S_{SEM_{ID^*}} \right)^{-c_{m,ID^*}} \\
&\quad \times \left(\frac{R_{FULL_{ID^*}}}{R_{SEM_{ID^*}}} R_{SEM_{ID^*}} \right)^{c_{m,ID^*}} X^{H(ID^*)c_{m,ID^*}}
\end{aligned}$$

$$\begin{aligned}
&= g^{z_{m,FULL_{ID^*}} A^{-c_{m,ID^*}} A^{c_{m,ID^*}} X^{-H(ID^*)c_{m,ID^*}}} \\
&\quad \times X^{H(ID^*)c_{m,ID^*}} \\
&= g^{z_{m,FULL_{ID^*}}}.
\end{aligned} \tag{21}$$

Eventually, I stops Phase 1 and outputs the challenge ID , ID^* on which it wishes to be challenged on. M checks if $ID = ID_l$ from L_H and aborts if not. Otherwise, M runs I now as a cheating prover on ID^* by obtaining its commitment, $Y_{FULL_{ID^*}}, S_{FULL_{ID^*}}, R_{FULL_{ID^*}}$ selecting a challenge $c_1 \xleftarrow{\$} \mathbb{Z}_q^*$ and obtaining the response $z_{FULL_{ID^*},1}$ from I . M then resets I to the step whereby I just sent $Y_{FULL_{ID^*}}, S_{FULL_{ID^*}}, R_{FULL_{ID^*}}$, selects a second challenge $c_2 \xleftarrow{\$} \mathbb{Z}_q^*$, and receives $z_{FULL_{ID^*},2}$ as response. M is then able to extract the full user private key as follows:

$$\begin{aligned}
&\frac{z_{FULL_{ID^*},1} - z_{FULL_{ID^*},2}}{c_1 - c_2} \\
&= \frac{y_{FULL_{ID^*}} + c_1 a - y_{FULL_{ID^*}} + c_2 a}{c_1 - c_2} \\
&= \frac{(c_1 - c_2) a}{c_1 - c_2} \\
&= a.
\end{aligned} \tag{22}$$

M outputs a as the discrete log solution.

It remains to calculate the probability of M solving the CDH problem and winning the game. The probability of M successfully extracting two valid conversation transcripts from I is bounded by $(\epsilon_{SMIBI} - (1/q))^2$ as given by the reset lemma [20]:

$$\begin{aligned}
&\Pr [M \text{ wins DL}] \\
&= \Pr [M \text{ computes } a \wedge \neg \text{abort}] \\
&= \Pr [M \text{ computes } a \mid \neg \text{abort}] \Pr [\neg \text{abort}] \\
&\epsilon_{DL} \geq \left(\epsilon_{SMIBI} - \frac{1}{q} \right)^2 \Pr [\neg \text{abort}].
\end{aligned} \tag{23}$$

Finally, let δ be the probability that I issues a *SEM-Extract* and a *USER-Extract* query on ID^* and that I makes a total of q_{SMIBI} of such queries. The probability of M answering all the extraction queries is δ_l^q . In Phase 2, the probability of M not aborting is if I outputs the challenge identity ID^* that was not queried before. This is given by the probability $1 - \delta$. Compiling them, the probability of M not aborting is $\delta^{q_{SMIBI}} (1 - \delta)$. This value is maximised at $\delta_{opt} = 1 - (1/(q_{SMIBI} + 1))$. Using δ_{opt} , The probability that M does not abort is at least $1/e(q_{SMIBI} + 1)$ because the value $(1 - (1/(q_{SMIBI} + 1)))^{q_{SMIBI}}$

approaches $1/e$ for large q_{SMIBI} . Therefore, the advantage of M , ϵ_{DL} , and the bound of the simulation is given as follows:

$$\begin{aligned}
\epsilon_{DL} &\geq \left(\epsilon_{SMIBI} - \frac{1}{q} \right)^2 \frac{1}{e(q_{SMIBI} + 1)} \\
\epsilon_{DL} e(q_{SMIBI} + 1) &\geq \left(\epsilon_{SMIBI} - \frac{1}{q} \right)^2 \\
\epsilon_{SMIBI} &\leq \sqrt{\epsilon_{DL} e(q_{SMIBI} + 1)} + \frac{1}{q}.
\end{aligned} \tag{24}$$

□

4.2. Security Analysis: Impersonation under Active/Concurrent Attack

Theorem 8. *The BNN-SM-IBI scheme is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -secure against impersonation under active/concurrent attacks in the random oracle if the OMCDH Problem is $(t_{OMCDH}, q_{OMCDH}, \epsilon_{OMCDH})$ -hard where*

$$\epsilon_{SMIBI} \leq \sqrt{\epsilon_{OMCDH} e(q_{SMIBI} + 1)} + \frac{1}{q}. \tag{25}$$

Proof. Assume the BNN-SM-IBI scheme is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -breakable; then a simulator M that $(t_{OMCDH}, q_{OMCDH}, \epsilon_{OMCDH})$ -breaks the OMCDH Problem can be shown. M takes in input (g, g^a) , is given access to *CHALL* and *DLOG* oracles, and runs the impersonator I as a subroutine. Without loss of generality, it can be assumed that any *SEM-Extract*, *User-Extract*, *SEM-Identification*, *User-Identification*, and *Full-Identification* queries are preceded by a *Create-User* query. To avoid collision and consistently respond to these queries, M maintains a list $L_H = \langle ID_i, Q_{ID_i}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, r_{FULL_{ID_i}}, r_{SEM_{ID_i}}, r_{USER_{ID_i}}, R_{FULL_{ID_i}}, R_{SEM_{ID_i}}, R_{USER_{ID_i}} \rangle$ which is initially empty. The following shows how M simulates the environment and oracle queries for I .

- (1) *Setup.* M sets the system parameters as $\text{mpk} = \langle G, q, g, X = g^x, H \rangle$ and keeps master private key x secret. It sends mpk to I .
- (2) *Create-User* (ID_i). M chooses $l \in \{1, \dots, q_H\}$ randomly and lets $ID_l = ID^*$ at this point. Whenever I makes a H query on ID_i , consider the following.

(i) If $i = l$, M chooses $Q_{ID^*} \xleftarrow{\$} \mathbb{Z}_q$, queries *CHALL* for W_0 , and sets $S_{FULL_{ID^*}} = W_0$ and $R_{FULL_{ID^*}} = W_0 X^{Q_{ID^*}}$. M returns Q_{ID^*} to I . M adds $L_H = \langle ID_l, Q_{ID^*}, S_{FULL_{ID^*}}, S_{SEM_{ID^*}} = \perp, S_{USER_{ID^*}} = \perp, S_{FULL_{ID^*}} = \perp, S_{SEM_{ID^*}} = \perp, S_{USER_{ID^*}} = \perp, r_{FULL_{ID^*}} = \perp, r_{SEM_{ID^*}} = \perp, r_{USER_{ID^*}} = \perp, R_{FULL_{ID^*}}, R_{SEM_{ID^*}} = \perp, R_{USER_{ID^*}} = \perp \rangle$ to L_H .

(ii) Otherwise, for $i \neq l$, M chooses $Q_{ID_i} \xleftarrow{\$} \mathbb{Z}_q^*$ and returns Q_{ID_i} . It then chooses $r_{FULL_{ID_i}}, r_{SEM_{ID_i}}, S_{SEM_{ID_i}} \xleftarrow{\$} \mathbb{Z}_q^*$, calculates

$s_{FULL_{ID_i}} = r_{FULL_{ID_i}} + xQ$, and sets $r_{USER_{ID_i}} = r_{FULL_{ID_i}} - r_{SEM_{ID_i}}$ and $s_{USER_{ID_i}} = s_{FULL_{ID_i}} - s_{SEM_{ID_i}}$. M adds $L_H = \langle ID_i, Q_{ID_i}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, s_{FULL_{ID_i}}, s_{SEM_{ID_i}}, s_{USER_{ID_i}}, r_{FULL_{ID_i}}, r_{SEM_{ID_i}}, r_{USER_{ID_i}}, R_{FULL_{ID_i}}, R_{SEM_{ID_i}}, R_{USER_{ID_i}} \rangle$ to L_H .

(3) *SEM-Extract* (ID_i). If $ID_i = ID^*$ and *USER-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *USER-Extract* (ID^*) has not been queried before, M will choose $r_{SEM_{ID^*}}, s_{SEM_{ID^*}} \xleftarrow{\$} \mathbb{Z}_q$ and sets $R_{SEM_{ID^*}} = g^{r_{SEM_{ID^*}}}$ and $S_{SEM_{ID^*}} = g^{s_{SEM_{ID^*}}}$. M also sets $S_{USER_{ID^*}} = S_{FULL_{ID^*}}/S_{SEM_{ID^*}}$ and $R_{USER_{ID^*}} = R_{FULL_{ID^*}}/R_{SEM_{ID^*}}$, saves these values in L_H , and returns them to I . These values will be used in the event of an *Identification* query later on. Otherwise for all other $ID_i \neq ID^*$, M just retrieves $\langle s_{SEM_{ID_i}}, R_{SEM_{ID_i}} \rangle$ in L_H and returns it to I .

(4) *USER-Extract* (ID_i). If $ID_i = ID^*$ and *SEM-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *SEM-Extract* (ID^*) has not been queried before, M will choose $r_{USER_{ID^*}}, s_{USER_{ID^*}} \xleftarrow{\$} \mathbb{Z}_q$ and sets $R_{USER_{ID^*}} = g^{r_{USER_{ID^*}}}$ and $S_{USER_{ID^*}} = g^{s_{USER_{ID^*}}}$. M also sets $S_{SEM_{ID^*}} = S_{FULL_{ID^*}}/S_{USER_{ID^*}}$ and $R_{SEM_{ID^*}} = R_{FULL_{ID^*}}/R_{USER_{ID^*}}$, saves these values in L_H , and returns them to I . These values will be used in the event of an *Identification* query later on. Otherwise for all other $ID_i \neq ID^*$, M just retrieves $\langle s_{USER_{ID_i}}, R_{USER_{ID_i}} \rangle$ in L_H and returns it to I .

(5) *Identification* ($ID_j, session$). I will act as the cheating verifier to learn information from valid conversations from M . If $ID_j \neq ID^*$, M just retrieves ID_j 's entry in L_H and runs the identification protocol for either the SEM's interactions, the user's interactions, or both combined as in the protocol.

Otherwise, for $ID_i = ID^*$, M then creates a full valid conversation for each m th query by querying *CHALL* for W_m and setting $Y_{m,FULL_{ID^*}} = W_m$. M also retrieves. Additionally, M chooses $Y_{m,SEM_{ID_j}} \xleftarrow{\$} G$ and sets $Y_{m,USER_{ID_j}} = W_m/Y_{m,SEM_{ID_j}}$. M also retrieves $S_{m,FULL_{ID_j}}, S_{m,SEM_{ID_j}}, S_{m,USER_{ID_j}}, R_{m,FULL_{ID_j}}, R_{m,SEM_{ID_j}}$, and $R_{m,USER_{ID_j}}$ from L_H . Upon receiving c_{m,ID_j} from I , M will query $z_{m,FULL_{ID_j}} = DLOG(W_m W_0^{c_{m,ID_j}})$, selects $z_{m,SEM_{ID_j}} \xleftarrow{\$} \mathbb{Z}_q$, and sets $z_{m,USER_{ID_j}} = z_{m,FULL_{ID_j}} - z_{m,SEM_{ID_j}}$. M tags the session with $session = m$.

(a) If I queries *SEM-Identification* (ID_j, m), M will commit $Y_{m,SEM_{ID_j}}, S_{SEM_{ID_j}}, R_{SEM_{ID_j}}$ and respond with $z_{m,SEM_{ID_j}}$ upon receiving c_{m,ID_j} .

(b) If I queries *User-Identification* (ID_j, m), M will commit $Y_{m,USER_{ID_j}}, S_{USER_{ID_j}}, R_{USER_{ID_j}}$ and respond with $z_{m,USER_{ID_j}}$ upon receiving c_{m,ID_j} .

(c) If I queries *Full-Identification* (ID_j, m), M will commit $Y_{m,FULL_{ID_j}}, S_{FULL_{ID_j}}, R_{FULL_{ID_j}}$ and respond with $z_{m,FULL_{ID_j}}$ upon receiving c_{m,ID_j} .

If no session is specified for the query, M just returns the next session in sequence. One can see that *SEM-Identification* (ID_j, m) and *User-Identification* (ID^*, m) can be combined to create a full and valid conversation on session m :

$$\begin{aligned}
& Y_{m,FULL_{ID_j}} R_{FULL_{ID_j}}^{c_{m,ID_j}} X^{H(ID_j)c_{m,ID_j}} \\
&= Y_{m,USER_{ID_j}} Y_{m,SEM_{ID_j}} R_{USER_{ID_j}}^{c_{m,ID_j}} R_{SEM_{ID_j}}^{c_{m,ID_j}} X^{H(ID_j)c_{m,ID_j}} \\
&= \frac{W_m}{Y_{m,SEM_{ID_j}}} Y_{m,SEM_{ID_j}} \left(\frac{W_0 X^{-H(ID)}}{R_{SEM_{ID_j}}} \right)^{c_{m,ID_j}} \\
&\quad \times R_{SEM_{ID_j}}^{c_{m,ID_j}} X^{H(ID_j)c_{m,ID_j}} \tag{26} \\
&= W_m \left(W_0^{c_{m,ID_j}} \right) X^{-H(ID)c_{m,ID_j}} X^{H(ID_j)c_{m,ID_j}} \\
&= W_m \left(W_0^{c_{m,ID_j}} \right) \\
&= g^{DLOG(W_m W_0^{c_{m,ID_j}})} \\
&= g^{z_{m,FULL_{ID_j}}}.
\end{aligned}$$

Eventually, I stops Phase 1 and outputs the challenge ID, ID^* on which it wishes to be challenged on. M checks if $ID^* = ID_l$ from L_H and aborts if not. Otherwise, M runs I now as a cheating prover on ID^* . M runs by obtaining its commitment, $Y_{FULL_{ID^*}}, S_{FULL_{ID^*}}, R_{FULL_{ID^*}}$, selects a challenge $c_1 \xleftarrow{\$} \mathbb{Z}_q^*$, and obtains the response $z_{FULL_{ID^*},1}$ from I . M then resets I to the step whereby I just sent $Y_{FULL_{ID^*}}, S_{FULL_{ID^*}}, R_{FULL_{ID^*}}$, selects a second challenge $c_2 \xleftarrow{\$} \mathbb{Z}_q^*$, and receives $z_{FULL_{ID^*},2}$ as response. M is then able to extract the full user private key as follows:

$$\begin{aligned}
& \frac{z_{FULL_{ID^*},1} - z_{FULL_{ID^*},2}}{c_1 - c_2} \\
&= \frac{y_{FULL_{ID^*}} + c_1 w_0 - y_{FULL_{ID^*}} + c_2 w_0}{c_1 - c_2} \tag{27} \\
&= \frac{(c_1 - c_2) w_0}{c_1 - c_2} \\
&= w_0.
\end{aligned}$$

M outputs w_0 as the initial discrete log challenge solution.

TABLE 1: Operation costs for the GQ-SM-IBI scheme.

Algorithm	A	M1	M2	E
Setup	0	0	0	0
Extract	1	0	0	3
SEM-Prove	0	0	1	2
User-Prove	0	0	3	2
Verify	0	0	1	2

A: addition in $\mathbb{Z}_{\phi(N)}$, M1: multiplication in $\mathbb{Z}_{\phi(N)}$, M2: multiplication in \mathbb{Z}_N^* , and E: exponentiation in \mathbb{Z}_N^* .

TABLE 2: Communication costs between algorithms for the GQ-SM-IBI scheme.

Algorithms	Bitstring ID	Elements of $ e - 1 $	Elements in \mathbb{Z}_N^*
TA-User (Extract)	1	0	1
TA-SEM (Extract)	1	0	1
SEM-User (Identification)	1	1	2
User-Verifier (Identification)	1	1	2

M proceeds to calculate the solutions to the other challenges as follows:

$$\begin{aligned}
& z_{m,FULL_{ID_j}} - w_0 c_{m,ID_j} \\
&= DLOG \left(W_m W_0^{c_{m,ID_j}} \right) - w_0 c_{m,ID_j} \\
&= \left[w_m + w_0 c_{m,ID_j} \right] - w_0 c_{m,ID_j} \\
&= w_m.
\end{aligned} \tag{28}$$

The probability study for the simulation above is similar to that of the impersonation under passive attack game and is therefore omitted. \square

5. Efficiency Analysis

In this section we provide the breakdown of operation costs for both the GQ-SM-IBI scheme and the BNN-SM-IBI scheme.

We measure the operation costs of the GQ-SM-IBI scheme in terms of addition operations in $\mathbb{Z}_{\phi(N)}$, multiplication operations in $\mathbb{Z}_{\phi(N)}$ and \mathbb{Z}_N^* , and exponentiation operations in \mathbb{Z}_N^* . Overall, the operational costs of the GQ-SM-IBI scheme are given in Table 1.

We also provide the communication costs of the GQ-SM-IBI scheme in Table 2.

As for the BNN-SM-IBI scheme, we measure the operational costs in terms of group operational costs of addition modulo q , multiplication modulo q , multiplication in group G , and exponentiations modulo q . Overall, the operational costs of the BNN-SM-IBI scheme are given in Table 3.

We also provide the communication costs of the BNN-SM-IBI scheme in Table 4.

TABLE 3: Operation costs for the BNN-SM-IBI scheme.

Algorithm	A	M	GM	E
Setup	0	0	0	1
Extract	3	1	0	3
SEM-Prover	1	1	0	2
User-Prover	2	1	3	2
Verifier	0	1	2	3

H: hash operation, A: addition mod q , M: multiplication mod q , GM: group multiplication, and E: exponentiation mod q .

TABLE 4: Communication costs between algorithms for the BNN-SM-IBI scheme.

Algorithms	Bitstring	Elements in G_1	Elements in \mathbb{Z}_q
TA-User (Extract)	1	1	1
TA-SEM (Extract)	1	1	1
SEM-User (Identification)	1	3	2
User-Verifier (Identification)	1	3	2

TABLE 5: Comparison of average simulated runtimes for SM-IBI schemes.

	Identification (ns)
GQ-SM-IBI (1024-bits)	2,158,747
BNN-SM-IBI (1024-bits)	14,064,652
BLS-SM-IBI (512-bits)	104,664,826
GQ-SM-IBI (2048-bits)	7,706,801
BNN-SM-IBI (2048-bits)	82,456,452
BLS-SM-IBI (1024-bits)	487,682,172
GQ-SM-IBI (3072-bits)	17,027,899
BNN-SM-IBI (3072-bits)	182,153,886
BLS-SM-IBI (1536-bits)	1,188,023,987

However, it is difficult to compare all these operational costs since the schemes are using operations defined in different fields and groups. Therefore, we built a simulator to generate running time results using a common platform.

We compare the running time of only the identification protocol, since the protocol is invoked every time a prover wishes to perform an interaction with a verifier. We compare the running times of the GQ-SM-IBI scheme and BNN-SM-IBI scheme as well as the original pairing-based scheme by [19]. We name the scheme from [19] as BLS-SM-IBI since its design follows the BLS signature scheme by Boneh et al. from [22]. For GQ-SM-IBI and BNN-SM-IBI the level of security used was 1024-bits, 2048-bits, and 3072-bits. We also compared the results with the equivalent level of security for BLS-SM-IBI pairing-based scheme of 512-bits, 1024-bits, and 1536-bits, respectively.

The simulation was run on a i7-2630QM platform with 4 GB RAM running 64-bit Windows 7. The library used was Java Cryptography Extension. We ran the simulation 100

iterations for each algorithm and took the average running time, measured in nanoseconds as presented in Table 5.

From the results, one can see that the GQ-SM-IBI scheme has the fastest identification protocol running time with the BNN-SM-IBI scheme trailing behind. However, both pairing-free schemes vastly outperform the BLS-SM-IBI that is pairing-based with at least an approximate factor of 6 to 7 for all three security levels. Therefore we obtain faster SM-IBI schemes from pairing-free alternatives.

6. Conclusion

We proposed two SM-IBI schemes that have an instant revocation feature and are very efficient. Our schemes outperform the only pairing-based SM-IBI currently known and are provably secure in the random oracle model against both passive and active/concurrent attackers.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to acknowledge the Ministry of Education, Malaysia, for financially aiding this research through the Fundamental Research Grant Scheme FRGS/2/2013/ICT07/MMU/03/5.

References

- [1] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, G. R. Blakley and D. Chaum, Eds., vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, Springer, Berlin, Germany, 1984.
- [2] D. Boneh, X. Ding, G. Tsudik, and C. M. Wong, "A method for fast revocation of public key certificates and security capabilities," in *Proceedings of the 10th Conference on USENIX Security Symposium (SSYM '01)*, D. S. Wallach, Ed., vol. 10, p. 22, Berkeley, Calif, USA, 2001.
- [3] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," in *Advances in Cryptology-CRYPTO '86*, A. M. Odlyzko, Ed., vol. 263 of *Lecture Notes in Computer Science*, pp. 186–194, Springer, Berlin, Germany, 1986.
- [4] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," in *Advances in Cryptology-EUROCRYPT 2004*, C. Cachin and J. Camenisch, Eds., vol. 3027 of *Lecture Notes in Computer Science*, pp. 268–286, Springer, Berlin, Germany, 2004.
- [5] K. Kurosawa and S. H. Heng, "From digital signature to ID-based identification/signature," in *Public Key Cryptography-PKC 2004*, F. Bao, R. H. Deng, and J. Zhou, Eds., vol. 2947 of *Lecture Notes in Computer Science*, pp. 248–261, Springer, Berlin, Germany, 2004.
- [6] J. J. Chin, S. H. Heng, and B. M. Goi, "An efficient and provable secure identity-based identification scheme in the standard model," in *Public Key Infrastructure*, S. F. Mjølne, S. Mauw, and S. K. Katsikas, Eds., vol. 5057 of *Lecture Notes in Computer Science*, pp. 60–73, Springer, Berlin, Germany, 2008.
- [7] J. J. Chin and S. H. Heng, "An adaptive-secure k -resilient identity-based identification scheme in the standard model," in *Proceedings of the FTRA-ACSA Summer*, Vancouver, Canada, June 2012.
- [8] J. J. Chin, S. H. Heng, and B. M. Goi, "Hierarchical identity-based identification schemes," in *Security Technology*, D. Slezak, T. H. Kim, W. C. Fang, and K. P. Arnett, Eds., vol. 58 of *Communications in Computer and Information Science*, pp. 93–99, Springer, Berlin, Germany, 2009.
- [9] A. Fujioka, T. Saito, and K. Xagawa, "Secure hierarchical identity-based identification without random oracles," in *Information Security*, D. Gollmann and F. C. Freiling, Eds., vol. 7483 of *Lecture Notes in Computer Science*, pp. 258–273, Springer, Berlin, Germany, 2012.
- [10] A. Fujioka, T. Saito, and K. Xagawa, "Applicability of OR-proof techniques to hierarchical identity-based identification," in *Cryptology and Network Security*, J. Pieprzyk, A. R. Sadeghi, and M. Manulis, Eds., vol. 7712 of *Lecture Notes in Computer Science*, pp. 169–184, Springer, Berlin, Germany, 2012.
- [11] J. J. Chin, R. C. W. Phan, R. Behnia, and S. H. Heng, "An efficient and provably secure certificateless identification scheme," in *SECRYPT 2013*, P. Samarati, Ed., pp. 371–378, SciTePress, 2013.
- [12] X. Ding and G. Tsudik, "Simple identity-based cryptography with mediated RSA," in *Topics in Cryptology-CT-RSA 2003*, M. Joye, Ed., vol. 2612 of *Lecture Notes in Computer Science*, pp. 193–210, Springer, Berlin, Germany, 2003.
- [13] B. Libert and J. J. Quisquater, "Efficient revocation and threshold pairing based cryptosystems," in *Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing (PODC '03)*, E. Borowsky and S. Rajsbaum, Eds., pp. 163–171, ACM, Boston, Mass, USA, July 2003.
- [14] X. Cheng, L. Guo, and X. Wang, "An identity-based mediated signature scheme from bilinear pairing," *International Journal of Network Security*, vol. 2, no. 1, pp. 29–33, 2006.
- [15] S. S. M. Chow, C. Boyd, and J. M. G. Nieto, "Security-mediated certificateless cryptography," in *Public Key Cryptography-PKC 2006*, M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, Eds., vol. 3958 of *Lecture Notes in Computer Science*, pp. 508–524, Springer, Berlin, Germany, 2006.
- [16] W. S. Yap, S. S. M. Chow, S. H. Heng, and B. M. Goi, "Security mediated certificateless signatures," in *Applied Cryptography and Network Security*, J. Katz and M. Yung, Eds., vol. 4521 of *Lecture Notes in Computer Science*, pp. 459–477, Springer, Berlin, Germany, 2007.
- [17] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology-ASIACRYPT 2003*, C.-S. Laih, Ed., vol. 2894 of *Lecture Notes in Computer Science*, pp. 452–473, Springer, Berlin, Germany, 2003.
- [18] J. J. Chin, R. Behnia, S. H. Heng, and R. C. W. Phan, "Cryptanalysis of a certificateless identification scheme," *Security and Communication Networks*, 2014.
- [19] J. J. Chin, R. Behnia, S. H. Heng, and R. C. Phan, "An efficient and provable secure security-mediated identity-based identification scheme," in *Proceedings of the 8th Asia Joint Conference on Information Security (Asia JCIS '13)*, pp. 27–32, IEEE, Seoul, South Korea, 2013.
- [20] M. Bellare and A. Palacio, "GQ and Schnorr identification schemes: proofs of security against impersonation under active and concurrent attacks," in *Advances in Cryptology-CRYPTO*

2002, M. Yung, Ed., vol. 2442 of *Lecture Notes in Computer Science*, pp. 162–177, Springer, Berlin, Germany, 2002.

- [21] S. Y. Tan, S. H. Heng, R. C. W. Phan, and B. M. Goi, “A variant of Schnorr identity-based identification scheme with tight reduction,” in *Future Generation Information Technology*, T. H. Kim, H. Adeli, D. Slezak et al., Eds., vol. 7105 of *Lecture Notes in Computer Science*, pp. 361–370, Springer, Berlin, Germany, 2011.
- [22] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the weil pairing,” *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

Research Article

Date Attachable Offline Electronic Cash Scheme

Chun-I Fan, Wei-Zhe Sun, and Hoi-Tung Hau

Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan

Correspondence should be addressed to Chun-I Fan; cifan@faculty.nsysu.edu.tw

Received 15 January 2014; Accepted 26 February 2014; Published 18 May 2014

Academic Editors: T. Cao, M. Ivanovic, and F. Yu

Copyright © 2014 Chun-I Fan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Electronic cash (e-cash) is definitely one of the most popular research topics in the e-commerce field. It is very important that e-cash be able to hold the anonymity and accuracy in order to preserve the privacy and rights of customers. There are two types of e-cash in general, which are online e-cash and offline e-cash. Both systems have their own pros and cons and they can be used to construct various applications. In this paper, we pioneer to propose a provably secure and efficient offline e-cash scheme with date attachability based on the blind signature technique, where expiration date and deposit date can be embedded in an e-cash simultaneously. With the help of expiration date, the bank can manage the huge database much more easily against unlimited growth, and the deposit date cannot be forged so that users are able to calculate the amount of interests they can receive in the future correctly. Furthermore, we offer security analysis and formal proofs for all essential properties of offline e-cash, which are anonymity control, unforgeability, conditional-traceability, and no-swindling.

1. Introduction

Due to the rapid growth of the Internet and communication developments, electronic commerce has become much more popular and widely used than ever [1–8]. The mobile telecommunications have been developed from 2 G to 3.5 G. Furthermore, LTE Advanced, 4 G, and 5 G are being implemented to the market in recent years. With the convenience of mobile network, people can do shopping or electronic payments by using any devices with network capability instead of leaving home. As a result, electronic commerce has been emphasized nowadays. Electronic cash (e-cash) is definitely one of the most popular research topics among electronic commerce. E-cash and the traditional cash notes are very much alike except e-cash is digitized and used on Internet transactions; therefore, it is very important that e-cash be able to hold the accuracy, privacy, and all other security concerns.

A typical e-cash system usually consists of payers (customers), payees (shops), and a bank. There are two types of e-cash in general which are online e-cash [9–13] and offline e-cash [14–27]. Online e-cash system involves participation of the bank during transactions (the payment stage). Banks are able to check whether customers have double-spent the e-cash(s) or not, and if yes, banks can terminate the transactions at once. Thus, the bank has to be online during every

transaction and it may lead to a bottleneck of the system. On the other hand, while banks do not participate in the payment stage of offline e-cash systems, double-spending check is only held during the deposit stage. Yet, the bank is set to be offline, but the system design is usually much more complicated than the online type and it may lead to a longer transaction time. Since both systems have their own pros and cons, they are used under different circumstances.

Extending online and offline e-cash systems, many e-cash schemes with other different features have been proposed over the years. For instance, e-cash can be stored compactly such that the space to store these e-cash is much reduced [15, 16], e-cash is generated by multiauthorities instead of one bank only [25], exact payments e-cash [13], recoverable e-cash which can be recovered when an e-cash is lost [26], and so on.

Based on the majority of the existing approaches, we summarize that a secure e-cash system should satisfy the following requirements.

- (i) *Anonymity*: no one, except the judge, can obtain any information of the e-cash owner's identity from the contents of e-cash.
- (ii) *Unlinkability*: no one, except the judge, can link any e-cash payment contents.

- (iii) *Unforgeability*: no one, except the bank, can generate a legal e-cash.
- (iv) *Double-Spending Control*: banks should have the ability to check if the e-cash is double-spent or not. No e-cash is allowed to be spent twice or more in an e-cash system.
- (v) *Conditional-Traceability*: the system should be able to trace and revoke the anonymity of users who violate any of the security rules so that they will receive penalties.
- (vi) *No-swindling*: no one, except the real owner, can spend a valid offline e-cash successfully.

In order to perform double-spending checks, banks have to store information of e-cash(s) in their database. Thus, the database of banks grows in direct proportion to the number of e-cash(s) withdrawn. Embedding an expiration date into each e-cash has been considered since it helps the banks to manage the database more easily. On the other hand, customers have to exchange their expired e-cash(s) with banks for new ones so as to keep the validity of the e-cash. Furthermore, customers will receive interest from banks after cash is deposited. In order to guarantee customers will receive the right amount of interest, it is necessary for customers to attach the deposit date to their e-cash(s) and the date cannot be modified by anyone else [11]. So far, there are a number of online e-cash schemes with an expiration date attachment [9, 11, 28]. However, there are very few offline approaches [21].

In this paper, we are going to propose an efficient date attachable offline e-cash scheme and provide formal proofs on essential properties to it in the random oracle model. Considering the practical needs, we pioneer to embed two kinds of date, which are expiration data and deposit date, to the offline e-cash. Moreover, we will offer an *E-cash renewal protocol* in our scheme (Section 3.2.5). Users can exchange their unused expired e-cash for a new one with another valid expiration date more efficiently. Compared with other similar works, our scheme is efficient from the aspect of considering computation cost.

The rest of this paper is organized as follows. In Section 2, we briefly review techniques employed throughout our scheme. Our proposed scheme is described in Section 3 in detail. Security proofs and analysis are covered in Section 4. Features and performance comparisons are made in Section 5, and the conclusion is given in Section 6.

2. Preliminaries

In this section, we briefly review techniques used in our date attachable offline e-cash scheme.

2.1. Chaum's Blind Signature Scheme. Blind signature was first introduced by Chaum [29]. It has been widely used in e-cash protocols since it has been proposed. A signer will not be able to view the content of the message while she/he is signing the message. Afterwards, a user can get a message with the signature of the signer by unblinding the signed message. The protocol is described as follows.

- (1) Initialization:

The signer randomly chooses two distinct large primes p and q , then computes $n = pq$ and $\phi(n) = (p-1)(q-1)$. Afterwards, the signer selects two integers e and d at random such that $ed \equiv 1 \pmod{\phi(n)}$. Finally, the signer publishes the public parameters (e, n) and a one-way hash function H .

- (2) User \rightarrow Signer: α

The user chooses a message m and a random integer r in \mathbb{Z}_n^* , then blinds the message by computing $\alpha = r^e H(m) \pmod{n}$ and sends it to the signer.

- (3) Signer \rightarrow User: t

After receiving α , the signer signs it with her/his private key d and sends it back to the user. The signed message will be $t = \alpha^d \pmod{n}$.

- (4) Unblinding:

After receiving t from the signer, the user unblinds it by computing $s = r^{-1}t \pmod{n}$. The signature-message pair is (s, m) .

- (5) Verification:

The (s, m) can be verified by checking if $s^e \equiv H(m) \pmod{n}$ is true or not.

2.2. Chameleon Hashing Based on Discrete Logarithm.

Chameleon hashing was proposed by Krawczyk and Rabin [30]. The chameleon hash function is associated with a one-time public-private key pair; it is a collision resistant function except for users who own a trapdoor for finding collision. Any user who knows the public key can compute the hashing, and for those who do not know the private key (trapdoor), it is impossible for them to find any two inputs which lead to the same hashing output. On the contrary, any user who knows the trapdoor can find the collision of given inputs. The construction of the chameleon hashing based on discrete logarithm is described as follows.

- (1) Setup:

(i) p, q : two large primes such that $p = kq + 1$,

(ii) g : an element order q in \mathbb{Z}_p^* ,

(iii) x : private key in \mathbb{Z}_q^* ,

(iv) y : public key, where $y = g^x \pmod{p}$.

- (2) *The function*: a message $m \in \mathbb{Z}_q^*$ is given and a random integer $r \in \mathbb{Z}_q^*$ is chosen. The hash is defined as $\text{CHAM-HASH}_y(m, r) = g^m y^r \pmod{p}$.

- (3) *Collision*: for a user who knows x , she/he is able to find the collision of the hash for any given m, m' such that $\text{CHAM-HASH}_y(m, r) = \text{CHAM-HASH}_y(m', r')$. The user derives r' in the equation $m + xr = m' + xr' \pmod{q}$.

3. The Proposed Date Attachable Offline Electronic Cash Scheme

In this section, we will introduce a new date attachable offline e-cash scheme. Considering the issues mentioned in Section 1, we propose a secure offline e-cash scheme with two specific kinds of date attached to the e-cash, which are expiration date and deposit date.

3.1. Outline of the Proposed Scheme. Here we are going to briefly describe the procedures of our scheme. The proposed scheme contains four protocols, *withdrawal protocol*, *payment protocol*, *deposit protocol*, and *e-cash renewal protocol*. A user withdraws an e-cash with an expiration date attached to it from the bank. A trusted computing platform (i.e., *judge device*) [31, 32], as stated in the proposed scheme, is installed in the bank to hold the identity information of all users and it will further help trace users when it is needed. It is impossible for anyone except the judge to obtain any information embedded in the device [33]. Nowadays, judge device can be implemented by the technique of Trusted Platform Module (TPM) [32, 34] in practice.

Before an e-cash is deposited, the depositor attaches the deposit date on the e-cash and sends it to the bank during the deposit stage. When the bank receives an e-cash, it will perform double-spending checking to verify whether the e-cash is doubly spent or not. The bank can derive secret parameters of the user who does double-spending and let the judge revoke the anonymity of the user. Besides, when an unused e-cash is expired, a user will be able to exchange it for a new one with a new expiration date. In our scheme, for the efficiency concerns, some of the unused parameters of users can remain unchanged while exchanging for a new valid e-cash. In the following sections, we will describe our scheme in detail.

3.2. The Proposed Scheme. Firstly, we define some notations as follows.

- (1) H_1, H_2, H_3 : three one-way hash functions, $H_1, H_2, H_3 : \{0, 1\}^* \rightarrow \{0, 1\}^n$.
- (2) H_4, H_5 : two one-way hash functions, $H_4, H_5 : \{0, 1\}^* \rightarrow \{0, 1\}^q$.
- (3) \tilde{E}_x, \tilde{D}_x : a secure symmetric cryptosystem. Plaintext is both encrypted and decrypted with a symmetric key x .
- (4) $\hat{E}_{pk}, \hat{D}_{sk}$: a secure asymmetric cryptosystem. Plaintext is encrypted with a public key pk and decrypted with the corresponding private key sk .
- (5) (pk_j, sk_j) : the public-private key pair of the judge.
- (6) (e_b, d_b) : the public-private key pair of bank.
- (7) *Date*: expiration date. It represents an effective spending date of a withdrawn e-cash. Any e-cash withdrawn in the same period will have the same expiration date, and vice versa.
- (8) ID_c : the identity of user C .

(9) l_k, l_r : the security parameters.

(10) A *judge device*: a tamper-resistant device which is issued by the judge. It is installed into the system of the bank. It is impossible to intercept or modify any information stored in the device.

3.2.1. Initialization. Initially, the bank randomly chooses two distinct large primes (p_b, q_b) and computes RSA parameters $n_b = p_b q_b$. It selects an integer e_b at random such that $\text{GCD}(\phi(n_b), e_b) = 1$, where $\phi(n_b) = (p_b - 1)(q_b - 1)$ and $1 < e_b < \phi(n_b)$. Then, it finds a d_b such that $e_b d_b \equiv 1 \pmod{\phi(n_b)}$. Secondly, it also chooses two other large primes p and q and two generators g_1 and g_2 of order q in \mathbb{Z}_p^* . Then, the bank publishes $(n_b, e_b, p, q, g_1, g_2, pk_j, H_1, H_2, H_3, H_4, H_5, \tilde{E}, \tilde{D}, \hat{E}, \hat{D})$. Meanwhile, the judge embeds $(n_b, e_b, p, q, g_1, g_2, pk_j, sk_j, H_1, H_2, H_3, H_4, H_5, \tilde{E}, \tilde{D}, \hat{E}, \hat{D})$ into a judge device and issues it to the bank.

3.2.2. Withdrawal Protocol. Users run the withdrawal protocol with banks to get an e-cash, as shown in Figure 1, yet banks have to obtain information of users' identity, such as ID_c or account numbers, before the withdrawal protocol is proceeded. Therefore, users should perform an authentication with banks beforehand. Users can execute the withdrawal protocol by any devices that have the ability to compute and connect to the network. For instance, users can use mobile phones or computers to perform the withdrawal protocol and store the withdrawn e-cash. The detailed steps of the protocol are as follows.

(1) Bank \rightarrow User: D

Firstly, the user prepares parameters for withdrawing an e-cash. The user chooses integers a, x_1, x_2, r_1, r_2 , and r_3 in random, where $a \in_R \mathbb{Z}_{n_b}^*$ and $x_1, x_2, r_1, r_2, r_3 \in_R \{0, 1, \dots, q - 1\}$ and selects a string $k \in_R \{0, 1\}^{l_k}$ randomly. The user then computes (y_1, w_1, y_2, w_2) , where $y_i = g_i^{x_i} \pmod p$ and $w_i = g_i^{r_i} \pmod p$ for $i = \{1, 2\}$. Secondly, the bank computes parameters for expiration date. It randomly chooses a r in \mathbb{Z}_n^* , prepares $D = \text{Date} \parallel r$ for some expiration date *Date*. The bank will send D to the user when she/he requests to withdraw an e-cash.

(2) User \rightarrow Bank: (α, ϵ)

After receiving D , the user prepares $\epsilon = \hat{E}_{pk_j}(k \parallel ID_c)$ and

$$\alpha = [a^{e_b} H_1^2(m \parallel D)]^{-1} \pmod{n_b}, \quad (1)$$

where $m = (y_1 \parallel w_1 \parallel y_2 \parallel w_2 \parallel r_3)$. Finally, the user sends (α, ϵ) to the bank.

(3) Bank \rightarrow Judge device: (ϵ, μ, D)

The bank sets $\mu = ID_c$, where ID_c is the identity of user C , and inputs it together with ϵ and D to the judge device.

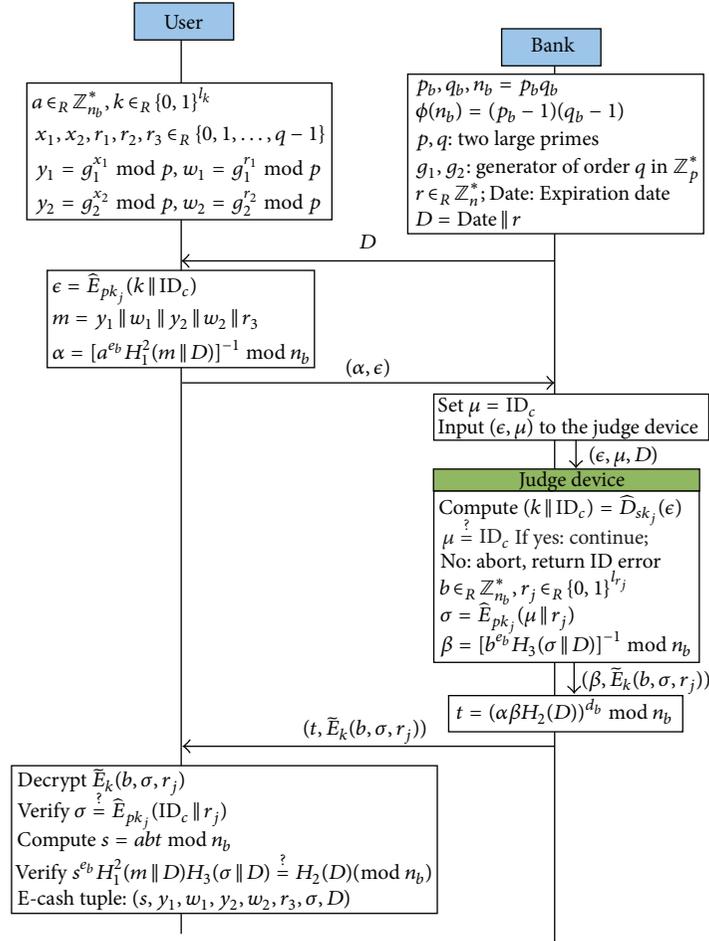


FIGURE 1: Withdrawal protocol.

(4) Judge device \rightarrow Bank: $(\beta, \tilde{E}_k(b, \sigma, r_j))$

The judge device decrypts ϵ and checks if $\mu = \text{ID}_c$. If not, it returns "ID error" to the bank; or else, it picks a random integer $b \in \mathbb{Z}_{n_b}^*$ and a string $r_j \in \{0, 1\}^{l_j}$ randomly. Then it computes $\sigma = \hat{E}_{pk_j}(\mu \parallel r_j)$ and

$$\beta = [b^{e_b} H_3(\sigma \parallel D)]^{-1} \bmod n_b. \quad (2)$$

Finally, it encrypts (b, σ, r_j) by using the symmetric key k and outputs it together with β to the bank.

(5) Bank \rightarrow User: $(t, \tilde{E}_k(b, \sigma, r_j))$

After receiving $(\beta, \tilde{E}_k(b, \sigma, r_j))$ from the judge device, it computes

$$t = (\alpha \beta H_2(D))^{d_b} \bmod n_b \quad (3)$$

and sends $(t, \tilde{E}_k(b, \sigma, r_j))$ to the user.

(6) Verifications

After receiving $(t, \tilde{E}_k(b, \sigma, r_j))$, the user firstly decrypts the ciphertext by using the symmetric key k

in order to obtain (b, σ, r_j) . Secondly, she/he checks if his/her ID is embedded correctly by computing if $\sigma = \hat{E}_{pk_j}(\text{ID}_c \parallel r_j)$ is true or not. Thirdly, she/he computes

$$s = abt \bmod n_b \quad (4)$$

and verifies s by checking if

$$s^{e_b} H_1^2(m \parallel D) H_3(\sigma \parallel D) = H_2(D) \pmod{n_b} \quad (5)$$

is true or not. Finally, when all verifications are done, the user gets the e-cash tuples (s, m, σ, D) and stores (x_1, x_2, r_1, r_2) for further payment usages.

3.2.3. Payment Protocol. When a user has to spend the e-cash, she/he performs the protocol as shown in Figure 2. The steps of the protocol are described as follows.

(1) User \rightarrow Shop: $(s, m, \sigma, D, x_2, r_2)$

The user sends $(s, m, \sigma, D, x_2, r_2)$ to the shop, where D contains the expiration date of the e-cash.

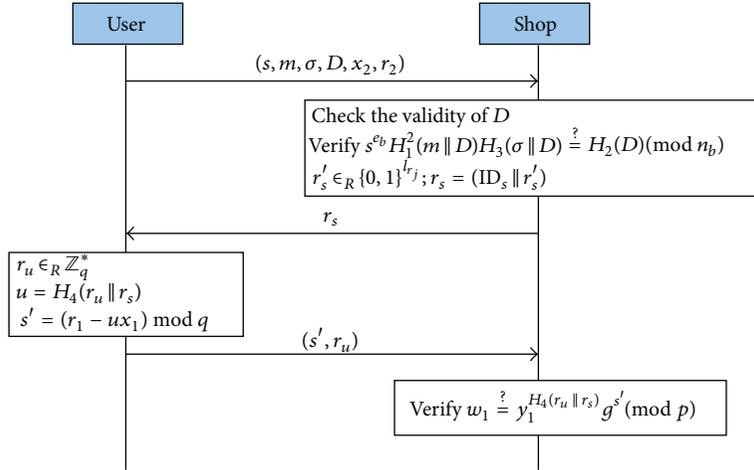


FIGURE 2: Payment protocol.

(2) Shop \rightarrow User: r_s

The shop first checks D to verify if the e-cash is still within the expiration date or not. If not, it terminates the transaction. Otherwise, it continues to verify $s^{e_b} H_1^2(m \parallel D) H_3(\sigma \parallel D) = H_2(D) \pmod{n_b}$. If it is not valid, the protocol is aborted; or else, it selects a string $r'_s \in_R \{0, 1\}^{l_j}$ and sets a challenge $r_s = (\text{ID}_s \parallel r'_s)$, where ID_s is the identity of the shop. Finally, it sends r_s to the user.

(3) User \rightarrow Shop: (s', r_u)

After receiving r_s from the shop, the user randomly selects a $r_u \in_R \mathbb{Z}_q^*$ and computes a response to the challenge

$$s' = (r_1 - u x_1) \pmod q, \quad (6)$$

where $u = H_4(r_u \parallel r_s)$. Then, the user sends (s', r_u) to the shop.

(4) Verifications

After receiving (s', r_u) from the user, the shop verifies if $w_1 = y_1^{H_4(r_u \parallel r_s)} g^{s'} \pmod p$ is true or not. If it is true, the shop will accept the e-cash. On the other hand, if it is not, the shop will reject it. Since it is an offline e-cash, the shop does not have to deposit it to the bank immediately. It can store the e-cash and deposit it later together with other received e-cash(s).

3.2.4. Deposit Protocol. As Figure 3 shows, shops attach the deposit date to their e-cash(s) and deposit them to banks in this protocol. Banks perform double-spending checks when they receive these e-cash(s). If any e-cash is double-spent, the bank will revoke the anonymity of the e-cash owner with the help of the judge. The steps are described in detail as follows.

(1) Shop \rightarrow Bank: $(s, m, \sigma, D, d, r_4, s', r_u, r_s)$

The shop computes $r_4 = r_2 - x_2 H_5(d)$, where d is the deposit date, and sends $(s, m, \sigma, D, d, r_4, s', r_u, r_s)$ to the bank.

(2) Verifications

Firstly, the bank checks the correctness of expiration date D and deposit date d , respectively, and also checks if

$$\begin{aligned} w_2 &= y_2^{H_5(d)} g_2^{r_4} \pmod p, \\ w_1 &= y_1^{H_4(r_u \parallel r_s)} g_2^{s'} \pmod p \end{aligned} \quad (7)$$

are true or not. Secondly, the bank verifies if $s^{e_b} H_1^2(m \parallel D) H_3(\sigma \parallel D) = H_2(D) \pmod{n_b}$ and checks the uniqueness of (s, m, σ, D) . Finally, if all of the above facts are verified successfully, the bank will accept and store the e-cash in its database and record $H_1(m \parallel D)$ in *exchange list*. Otherwise, it will reject this transaction and trace the owner of the e-cash.

3.2.5. E-Cash Renewal Protocol. In order to reduce the unlimited growth database problem of the bank, we have expiration date and renewal protocol in our scheme to achieve it, as shown in Figure 4. When an unused e-cash is expired, the user has to exchange it for another e-cash with a new expiration date from the bank.

(1) User \rightarrow Bank: (s, ρ, σ, D)

The user recalls $m = (y_1, w_1, y_2, w_2, x_2, r_3)$ and prepares

$$\rho = H_1(m \parallel D) \quad (8)$$

and sends it together with the unused (s, σ, D) to the bank.

(2) Verifications

Firstly, the bank checks the correctness of expiration date D and makes sure ρ does not exist in the *exchange list*. Secondly, the bank verifies if $s^{e_b} H_1(\rho) H_3(\sigma \parallel D) = H_2(D) \pmod{n_b}$. Finally, if all of the above facts are verified successfully, the bank will accept to

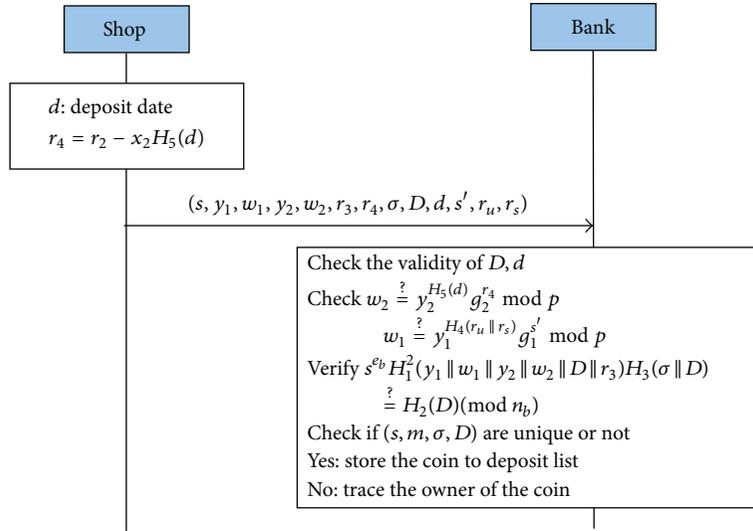


FIGURE 3: Deposit protocol.

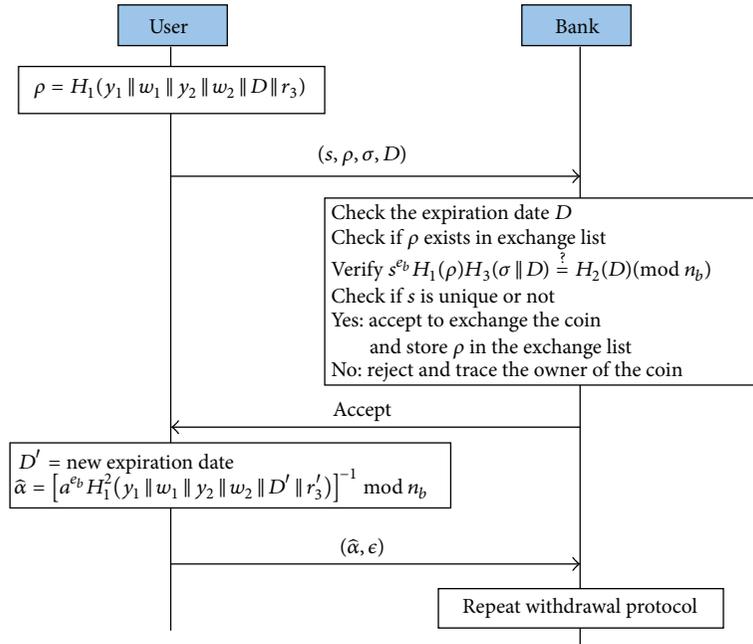


FIGURE 4: E-Cash renewal protocol.

exchange the e-cash. It will send a new expiration date D' and store ρ in the *exchange list*. Otherwise, it will reject the exchange request.

(3) User \rightarrow Bank: $(\tilde{\alpha}, \epsilon)$

The user computes

$$\tilde{\alpha} = [a^{e_b} H_1^2(m' \parallel D')]^{-1} \text{ mod } n_b, \quad (9)$$

where $m' = (y_1, w_1, y_2, w_2, x_2, r'_3)$, r'_3 is a random, and D' is the new expiration date issued by the bank. The user sends $(\tilde{\alpha}, \epsilon, ID_c)$ to the bank. Then the bank

repeats the withdrawal protocol in Section 3.2.2 from Step 2 with the user.

3.2.6. Double-Spending Checking and Anonymity Control.

In our scheme, the identity of the users is anonymous in general except when the users violate any security rules and, therefore, their identities will be revealed.

(1) Double-Spending Checking

When an e-cash is being doubly spent, there must be two e-cash(s) with the same record prefixed by $(s, y_1, w_1, y_2, w_2, r_3, \sigma, D)$ stored in the database of the

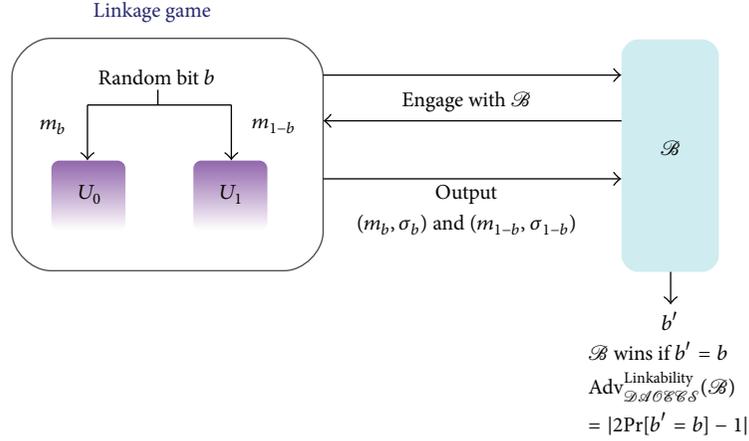


FIGURE 5: The game environment of linkage game.

bank. Therefore, the bank is able to detect any double-spent e-cash easily by checking the above parameters. For instance, the bank has received two e-cash(s),

$$\begin{aligned} & (s, y_1, w_1, y_2, w_2, x_2, r_3, r_4, \sigma, D, d, s', r_u, r_s), \\ & (s, y_1, w_1, y_2, w_2, x_2, r_3, \hat{r}_4, \sigma, D, \hat{d}, \hat{s}', \hat{r}_u, \hat{r}_s). \end{aligned} \quad (10)$$

Thus, the bank can obtain two equations as follows:

$$\begin{aligned} s' & \equiv r_1 - H_4(r_u \parallel r_s) x_1 \pmod{q}, \\ \hat{s}' & \equiv r_1 - H_4(\hat{r}_u \parallel \hat{r}_s) x_1 \pmod{q}. \end{aligned} \quad (11)$$

The bank can derive (x_1, r_1) from the above equations and send $(s, y_1, w_1, y_2, w_2, x_2, r_3, \sigma, D)$ and (x_1, r_1) to the judge to trace the owner of the e-cash.

(2) Revocation

The judge can trace any user who doubly spends e-cash(s) or violates any transaction regulations. When the judge receives $(s, y_1, w_1, y_2, w_2, x_2, r_3, \sigma, D)$ and (x_1, r_1) from the bank, it checks the following equations:

$$\begin{aligned} s^{e_b} H_1^2(m \parallel D) H_3(\sigma \parallel D) & \stackrel{?}{=} H_2(D) \pmod{n_b}, \\ y_1 & \stackrel{?}{=} g_1^{x_1} \pmod{p}, \\ w_1 & \stackrel{?}{=} g_1^{r_1} \pmod{p}. \end{aligned} \quad (12)$$

If all of the above equalities are true, the judge will decrypt σ and return the extracted ID_c to the bank.

4. Security Proofs

In this section, we provide security definitions and formal proofs of the following security features: unlinkability, unforgeability, traceability, and no-swindling for our

proposed date attachable offline electronic cash scheme (DAOEES).

4.1. E-Cash Unlinkability. Based on the definition of unlinkability introduced by Abe and Okamoto [35] and Juels et al. [36], we formally define the unlinkability property of DAOEES.

Definition 1 (The Linkage Game). Let U_0, U_1 , and \mathcal{J} be two honest users and the judge that follows DAOEES, respectively. Let \mathcal{B} be the bank that participates the following game with U_0, U_1 , and \mathcal{J} . The game environment is shown in Figure 5.

Step 1. According to DAOEES, \mathcal{B} generates the bank's public key (e_b, n_b) , the bank's private key (d_b, p_b, q_b) , system parameters (p, q, g_1, g_2) , the expiration date D , and the five public one-way hash functions H_1, H_2, H_3, H_4 , and H_5 . \mathcal{J} generates the judge's public-private key pair (pk_j, sk_j) .

Step 2. \mathcal{B} generates $x_{1i}, x_{2i}, r_{1i}, r_{2i}, r_{3i}$ in random, where $x_1, x_2, r_1, r_2, r_3 \in_R \{0, 1, \dots, q-1\}$, and computes (y_{ki}, w_{ki}) for $k = \{1, 2\}$ and $i = \{0, 1\}$, where $y_{ki} = g_k^{x_{ki}} \pmod{p}$ and $w_{ki} = g_k^{r_{ki}} \pmod{p}$.

Step 3. We choose a bit $\hat{b} \in \{0, 1\}$ randomly and place $(y_{1\hat{b}}, w_{1\hat{b}}, y_{2\hat{b}}, w_{2\hat{b}})$ and $(y_{11-\hat{b}}, w_{11-\hat{b}}, y_{21-\hat{b}}, w_{21-\hat{b}})$ on the private input tapes of U_0 and U_1 , respectively, where \hat{b} is not disclosed to \mathcal{B} .

Step 4. \mathcal{B} performs the withdrawal protocol of DAOEES with U_0 and U_1 , respectively.

Step 5. If U_0 and U_1 output two e-cash(s) $(s_{\hat{b}}, m_{\hat{b}}, \sigma_{\hat{b}}, D_{\hat{b}})$ and $(s_{1-\hat{b}}, m_{1-\hat{b}}, \sigma_{1-\hat{b}}, D_{1-\hat{b}})$, where $m_i = (y_{1i}, w_{1i}, y_{2i}, w_{2i}, r_{3i})$, on their private tapes, respectively, we give the two e-cash(s) in a random order to \mathcal{B} ; otherwise, \perp is given to \mathcal{B} .

Experiment $\text{Exp}_{\mathcal{A}}^{\text{FG-1}}(l_k)$
 $(pk_j, sk_j, g_1, g_2, e_b, d_b, p_b, q_b, n_b, H_1, H_2, H_3, H_4, H_5) \leftarrow \text{Setup}(l_k)$
 $\{(s_1, m_1, \sigma_1, D_1), \dots, (s_{\ell+1}, m_{\ell+1}, \sigma_{\ell+1}, D_{\ell+1})\} \leftarrow \mathcal{A}^{\mathcal{O}_S}(pk_j, g_1, g_2, e_b, n_b, H_1, H_2, H_3, H_4, H_5)$
 if the following checks are true, **return 1**;
 (i) $s_i^{e_b} H_1^2(m_i) H_3(\sigma_i \parallel D_i) \equiv H_2(D_i) \pmod{n_b}, \forall i \in \{1, \dots, \ell + 1\}$;
 (ii) $m_1, \dots, m_{\ell+1}$ are all distinct
 else **return 0**;

ALGORITHM 1: Experiment FG-1.

Step 6. \mathcal{B} outputs $\hat{b}' \in \{0, 1\}$ as the guess of \hat{b} . The bank \mathcal{B} wins the game if $\hat{b}' = \hat{b}$ and \mathcal{F} has not revoked the anonymity of $(s_{\hat{b}}, m_{\hat{b}}, \sigma_{\hat{b}}, D_{\hat{b}})$ and $(s_{1-\hat{b}}, m_{1-\hat{b}}, \sigma_{1-\hat{b}}, D_{1-\hat{b}})$ to \mathcal{B} . We define the advantage of \mathcal{B} as

$$\text{Adv}_{\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}}^{\text{Linkability}}(\mathcal{B}) = \left| 2\Pr[\hat{b}' = \hat{b}] - 1 \right|, \quad (13)$$

where $\Pr[\hat{b}' = \hat{b}]$ denotes the probability of $\hat{b}' = \hat{b}$.

Definition 2 (Unlinkability). A $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ satisfies the unlinkability property if and only if the advantage $\text{Adv}_{\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}}^{\text{Linkability}}(\mathcal{B})$ defined in Definition 1 is negligible.

Theorem 3. A $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ satisfies the unlinkability property of Definition 2 if the adopted cryptosystems are semantically secure.

Proof. If \mathcal{B} is given \perp in the Step 5 of the game, it will determine \hat{b} with probability 1/2, which is exactly the same as a random guess of \hat{b} .

Here, we assume that \mathcal{B} gets two e-cash $(s_0, m_0, \sigma_0, D_0)$ and $(s_1, m_1, \sigma_1, D_1)$. Let $(\alpha_i, \beta_i, t_i, \epsilon_i, \tilde{E}_{k_i}(b_i, \sigma_i, r_{j_i}))$, $i \in \{0, 1\}$, be the view of data exchanged between U_i and \mathcal{B} in the withdrawal protocol (Section 3.2.2) and let $(x_2, r_2, r_4, r_{u_i}, r_{s_i}, s'_i, d_i)$ be the view of data exchanged when \mathcal{B} performs the payment protocol (Section 3.2.3) and the deposit protocol (Section 3.2.4) by using $(s_i, m_i, \sigma_i, D_i)$, where $i \in \{0, 1\}$.

For $(s, m, \sigma, D, x_2, r_2, r_4, r_u, r_s, s', d) \in$

$$\left\{ (s_0, m_0, \sigma_0, D_0, x_2, r_2, r_4, r_{u_0}, r_{s_0}, s'_0, d_0), \right. \\ \left. (s_1, m_1, \sigma_1, D_1, x_2, r_2, r_4, r_{u_1}, r_{s_1}, s'_1, d_1) \right\} \quad (14)$$

and $(\alpha_i, \beta_i, t_i, \epsilon_i, \tilde{E}_{k_i}(b_i, \sigma_i, r_{j_i}))$, $i \in \{0, 1\}$, there always exists a pair (a'_i, b'_i) such that

$$a'_i = [\alpha_i H_1^2(m \parallel D)]^{-d_b} \pmod{n_b} \quad (\text{via (1)}), \quad (15)$$

$$b'_i = [\beta_i H_3(\sigma \parallel D)]^{-d_b} \pmod{n_b} \quad (\text{via (2)}).$$

And from (3), $t_i \equiv (\alpha_i \beta_i H_2(D))^{d_b} \pmod{n_b}$, (4) always holds as

$$s \equiv (a'_i b'_i t_i) \\ \equiv \left[(H_1^2(m \parallel D) H_3(\sigma \parallel D))^{-1} H_2(D) \right]^{d_b} \pmod{n_b}. \quad (16)$$

Besides, \hat{E}_{pk_j} and \tilde{E}_{k_i} are semantically secure encryption functions. \mathcal{B} cannot learn any information from ϵ_i and $\tilde{E}_{k_i}(b_i, \sigma_i, r_{j_i})$.

From the above, given any $(s, m, \sigma, D) \in \{(s_0, m_0, \sigma_0, D_0), (s_1, m_1, \sigma_1, D_1)\}$ and (α_i, β_i, t_i) , where $i \in \{0, 1\}$, there always exists a corresponding pair (a'_i, b'_i) such that (1), (2), (3), and (4) are satisfied.

Thus, go back to Step 6 of the game, the bank \mathcal{B} succeeds in determining \hat{b} with probability $(1/2) + \epsilon$, where ϵ is negligible since \hat{E} and \tilde{E} are semantically secure. Therefore, we have $\text{Adv}_{\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}}^{\text{Linkability}}(\mathcal{B}) = 2\epsilon$, which is negligible, so that $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ satisfies the unlinkability property. \square

4.2. E-Cash Unforgeability. In this section, we will formally prove that the proposed date attachable offline electronic cash scheme ($\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$) is secure against forgery attack. The forgery attack can be roughly divided into two types, one is the typical one-more forgery type (i.e., $(\ell, \ell + 1)$ -forgery) [37] and the other is the forgery on some specific expiration date of an e-cash after sufficient communications with the signing oracle (i.e., bank). The details of definitions and our formal proofs will be described as follows.

Definition 4 (Forgery Game 1 in $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ (FG-1)). Let $l_k \in \mathbb{N}$ be a security parameter and \mathcal{A} be an adversary in $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$. \mathcal{O}_S is an oracle which plays the role of the bank in $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ to be responsible for issuing e-cash(s) (i.e., (s, m, σ, D) , where $m = (w_1, y_1, w_2, y_2, r_3, D)$) according to the queries from \mathcal{A} . \mathcal{A} is allowed to query \mathcal{O}_S for ℓ times; consider the experiment $\text{Exp}_{\mathcal{A}}^{\text{FG-1}}(l_k)$ shown in Algorithm 1. \mathcal{A} wins the forgery game FG-1 if the probability $\Pr[\text{Exp}_{\mathcal{A}}^{\text{FG-1}}(l_k) = 1]$ of \mathcal{A} is nonnegligible.

Definition 5 (Forgery Game 2 in $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ (FG-2)). Let $l_k \in \mathbb{N}$ be a security parameter and \mathcal{A} be an adversary in $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$. \mathcal{O}_S is an oracle which plays the role of the bank in $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ to take charge of the following two events:

- (i) issue e-cash(s) (i.e., (s, m, σ, D) , where $m = (w_1, y_1, w_2, y_2, r_3, D)$) according to the queries from \mathcal{A} ,
- (ii) record the total number ℓ_{D_i} of each distinct expiration date D_i .

\mathcal{A} is allowed to query \mathcal{O}_S for ℓ times; consider the experiment $\text{Exp}_{\mathcal{A}}^{\text{FG-2}}(l_k)$ shown in Algorithm 2. \mathcal{A} wins the forgery game

Experiment Exp _{\mathcal{A}} ^{FG-2}(l_k)
 $(pk_j, sk_j, g_1, g_2, e_b, d_b, p_b, q_b, n_b, H_1, H_2, H_3, H_4, H_5) \leftarrow \text{Setup}(l_k)$
 $\{(s_i, m_i, \sigma_i, D^*) \mid 1 \leq i \leq \ell_{D^*} + 1\} \leftarrow \mathcal{A}^{\mathcal{O}_S}(pk_j, g_1, g_2, e_b, n_b, H_1, H_2, H_3, H_4, H_5)$
 if the following checks are true, **return 1**;
 (i) $s_i^{e_b} H_1^2(m_i) H_3(\sigma_i \parallel D^*) \equiv H_2(D^*) \pmod{n_b}, \forall i \in \{1, \dots, \ell_{D^*} + 1\}$;
 (ii) $m_1, \dots, m_{\ell_{D^*} + 1}$ are all distinct;
 else **return 0**;

ALGORITHM 2: Experiment FG-2.

Experiment Exp _{\mathcal{A}} ^{RSA-ACTI}(k)
 $(N, e, d) \xleftarrow{R} \text{KeyGen}(k)$.
 $(y_1, \dots, y_m) \leftarrow \mathcal{O}_i(N, e, k)$
 $\{\pi, (x_1, y_1), \dots, (x_n, y_n)\} \leftarrow \mathcal{A}^{\mathcal{O}_{\text{inv}}, \mathcal{O}_t}(N, e, k)$
 if the following checks are true, **return 1**;
 (i) $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, m\}$ is injective
 (ii) $x_i^e \equiv y_i \pmod{N}, \forall i \in \{1, \dots, n\}$
 (iii) $n > q_h$
 else **return 0**;

ALGORITHM 3

FG-2 if the probability $\Pr[\text{Exp}_{\mathcal{A}}^{\text{FG-2}}(k) = 1]$ of \mathcal{A} is nonnegligible.

Here we introduce the hard problems used in our proof models.

Definition 6 (Alternative Formulation of RSA Chosen-Target Inversion Problem (RSA-ACTI)). Let $k \in \mathbb{N}$ be a security parameter and \mathcal{A} be an adversary who is allowed to access the RSA-inversion oracle \mathcal{O}_{inv} and the target oracle \mathcal{O}_t . \mathcal{A} is allowed to query \mathcal{O}_t and \mathcal{O}_{inv} for m and q_h times, respectively. Consider Algorithm 3.

We say \mathcal{A} breaks the RSA-ACTI problem if the probability $\Pr[\text{Exp}_{\mathcal{A}}^{\text{RSA-ACTI}}(k) = 1]$ of \mathcal{A} is nonnegligible.

Definition 7 (The RSA Inversion Problem). Given (e, n) , where n is the product of two distinct large primes p and q with roughly the same length and e is a positive integer relatively-prime to $(p - 1)(q - 1)$, and a randomly-chosen positive integer y less than n , find an integer x such that $x^e \equiv y \pmod{n}$.

Definition 8 (E-Cash Unforgeability). If there exists no probabilistic polynomial-time adversary who can win FG-1 or FG-2, then $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ is secure against forgery attacks.

Theorem 9. For a polynomial-time adversary \mathcal{A} who can win FG-1 or FG-2 with nonnegligible probability, there exists another adversary \mathcal{S} who can break the RSA-ACTI problem or RSA inversion problem with nonnegligible probability.

Proof. \mathcal{S} simulates the environment and controls three hash oracles, \mathcal{O}_{H_1} , \mathcal{O}_{H_2} , \mathcal{O}_{H_3} and an e-cash producing oracle \mathcal{O}_S

of $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ scheme to respond to different queries from \mathcal{A} in the random oracle model and takes advantage of \mathcal{A} to solve RSA-ACTI problem or RSA inversion problem, simultaneously. Then, for consistency, \mathcal{S} maintains three lists \mathcal{L}_{H_1} , \mathcal{L}_{H_2} , and \mathcal{L}_{H_3} to record every response of \mathcal{O}_{H_1} , \mathcal{O}_{H_2} , and \mathcal{O}_{H_3} , respectively.

Here we will start to do the simulation for the two games (i.e., FG-1 and FG-2) to prove $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ is secure against forgery attacks. The details of simulation are set below and illustrated in Figures 6 and 7, respectively.

Simulation in FG-1. In this proof model, \mathcal{S} is allowed to query the oracles \mathcal{O}_{inv} (i.e., $(\cdot)^d$) and \mathcal{O}_t of RSA-ACTI problem defined in Definition 6 for helping \mathcal{S} to produce e-cash(s) and the corresponding verifying key is (e, n) .

(i) H_1 Query of \mathcal{O}_{H_1}

Initially, every blank record in \mathcal{L}_{H_1} can be represented as (\perp, \perp, \perp) . When \mathcal{A} sends m for querying the hash value $H_1(m)$, \mathcal{S} will check the list \mathcal{L}_{H_1} :

- (a) if $m = m_i$ for some i , then \mathcal{S} retrieves the corresponding $H_1(m_i)$ and returns it to \mathcal{A} ;
- (b) else if $m = H_1(m_i)$ and $H_1^2(m_i) \neq \perp$ for some i , then \mathcal{S} retrieves the corresponding $H_1^2(m_i)$ and returns it to \mathcal{A} ;
- (c) else if $m = H_1(m_i)$ and $H_1^2(m_i) = \perp$ for some i , then \mathcal{S} queries \mathcal{O}_t to get an instance y and returns it to \mathcal{A} , then fills the record $(m_i, H_1(m_i), \perp)$ as $(m_i, H_1(m_i), y)$ in \mathcal{L}_{H_1} ;
- (d) otherwise, \mathcal{S} selects a random $\rho \in \mathbb{Z}_n$, records (m, ρ, \perp) in \mathcal{L}_{H_1} , and returns ρ to \mathcal{A} .

(ii) H_2 Query of \mathcal{O}_{H_2}

When \mathcal{A} asks for H_2 query by sending D to \mathcal{S} , \mathcal{S} will look up the list \mathcal{L}_{H_2} :

- (a) if $D = D_i$ for some i , the corresponding τ will be retrieved and \mathcal{S} will send $(\tau^e \pmod{n})$ back to \mathcal{A} ;
- (b) otherwise, \mathcal{S} will select a random $\tau \in \mathbb{Z}_n$, record (D, τ) in \mathcal{L}_{H_2} , and return $(\tau^e \pmod{n})$ back to \mathcal{A} .

(iii) H_3 Query of \mathcal{O}_{H_3}

While \mathcal{A} sends (σ, D) to \mathcal{S} for $H_3(\sigma \parallel D)$, \mathcal{S} will look up the list \mathcal{L}_{H_3} :

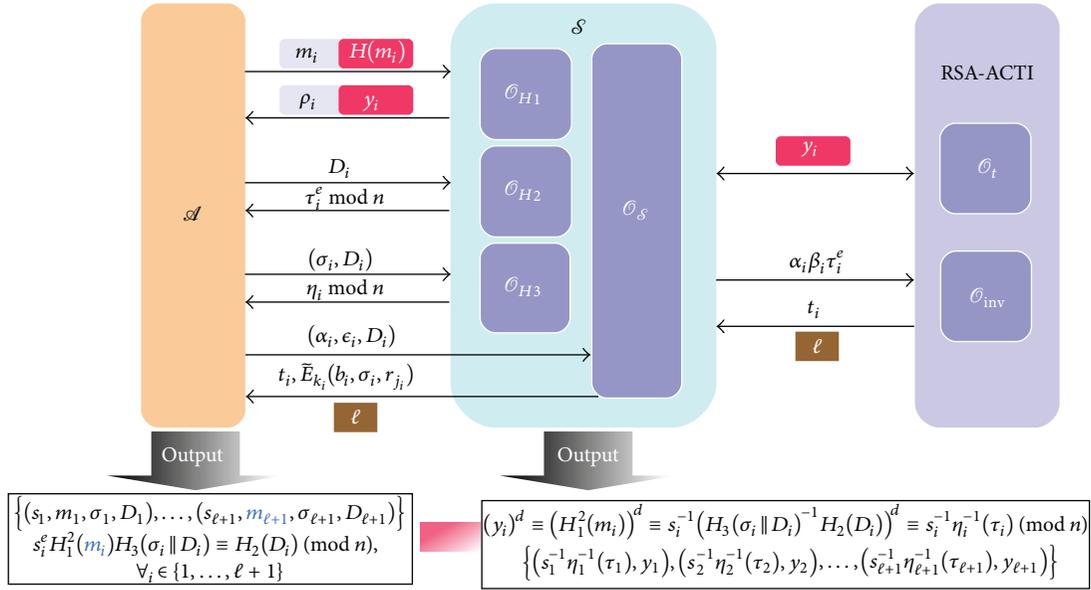


FIGURE 6: The proof model of FG-1.

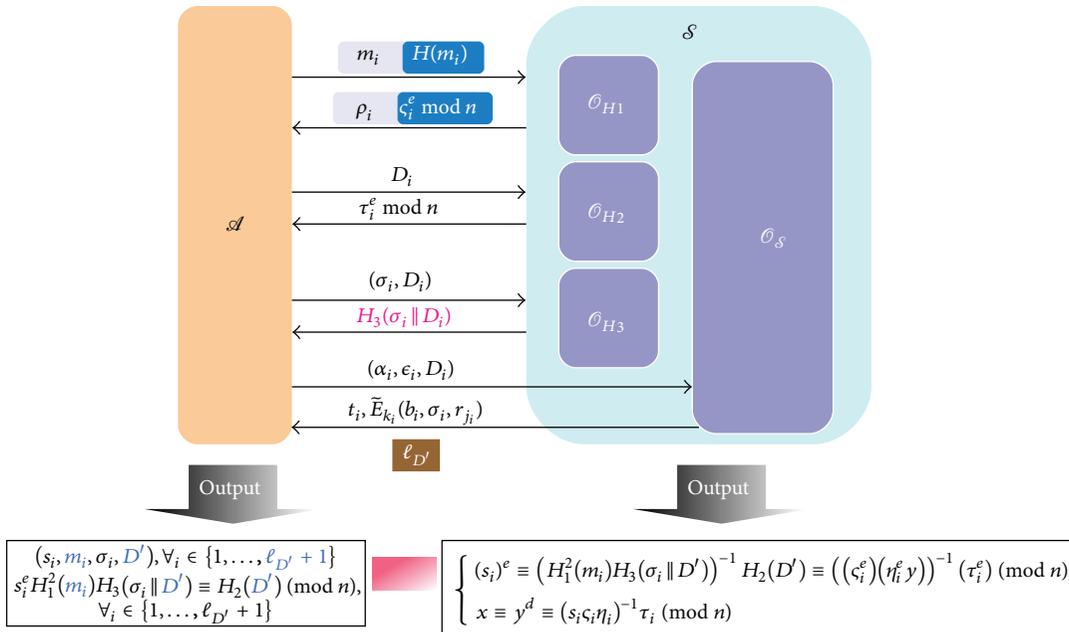


FIGURE 7: The proof model of FG-2.

- (a) if $(\sigma, D) = (\sigma_i, D_i)$ for some i , the corresponding η will be retrieved and $(\eta^e \bmod n)$ will be returned to \mathcal{A} ;
 - (b) otherwise, \mathcal{S} will select a random $\eta \in \mathbb{Z}_n$, record $((\sigma, D), \eta)$ in \mathcal{L}_{H_3} , and return $(\eta^e \bmod n)$ back to \mathcal{A} .
- (iv) E-Cash Producing Query of $\mathcal{O}_{\mathcal{S}}$
- When \mathcal{A} sends (α, ϵ, D) to \mathcal{S} , \mathcal{S} will do the following steps:
- (1) decrypt ϵ , obtain (k, ID) ;
 - (2) randomly select r_j and prepare $\sigma = \widehat{E}_{pk_j}(\text{ID} \parallel r_j)$;
 - (3) choose $\eta \in_R \mathbb{Z}_n$, set $H_3(\sigma \parallel D) = (\eta^e \bmod n)$, and store $((\sigma, D), \eta)$ in \mathcal{L}_{H_3} ;
 - (4) select $b \in_R \mathbb{Z}_n^*$ and compute $\beta = (b^e \eta^e)^{-1} \bmod n$;
 - (5) retrieve or assign τ such that $H_2(D) = (\tau^e)$ as the \mathcal{O}_{H_2} query described above;
 - (6) send $(\alpha \beta \tau^e)$ to oracle \mathcal{O}_{inv} to get $t = (\alpha \beta \tau^e)^d \bmod n$;
 - (7) return $(t, \widehat{E}_k(b, \sigma, r_j))$ back to \mathcal{A} .

Eventually, assume that \mathcal{A} can successfully output $\ell + 1$ e-cash tuples

$$\{(s_1, m_1, \sigma_1, D_1) \cdots (s_{\ell+1}, m_{\ell+1}, \sigma_{\ell+1}, D_{\ell+1})\}, \quad (17)$$

where m_i are all distinct, $\forall i, 1 \leq i \leq \ell + 1$, such that $s_i^e H_1^2(m) H_3(\sigma_i \parallel D_i) = H_2(D_i) \pmod{n}$ after ℓ times to query $\mathcal{O}_{\mathcal{S}}$ with nonnegligible probability $\epsilon_{\mathcal{A}}$.

According to \mathcal{L}_{H_1} , \mathcal{L}_{H_2} , and \mathcal{L}_{H_3} , \mathcal{S} can compute and retrieve RSA-inversion instances ($\forall i, 1 \leq i \leq \ell + 1$)

$$\begin{aligned} (y_i)^d &\equiv (H_1^2(m_i))^d \equiv s_i^{-1} (H_3(\sigma_i \parallel D_i)^{-1} H_2(D_i))^d \\ &\equiv s_i^{-1} \eta_i^{-1} (\tau_i) \pmod{n}. \end{aligned} \quad (18)$$

Via \mathcal{A} querying the signing oracle $\mathcal{O}_{\mathcal{S}}$ for ℓ times (i.e., query \mathcal{O}_{inv} for ℓ times by \mathcal{S}), \mathcal{S} can output $\ell + 1$ RSA-inversion instances

$$\begin{aligned} &\{(s_1^{-1} \eta_1^{-1} (\tau_1), y_1), (s_2^{-1} \eta_2^{-1} (\tau_2), y_2), \dots, \\ &(s_{\ell+1}^{-1} \eta_{\ell+1}^{-1} (\tau_{\ell+1}), y_{\ell+1})\} \end{aligned} \quad (19)$$

and break the RSA-ACTI problem with nonnegligible probability at least $\epsilon_{\mathcal{A}}$.

Simulation in FG-2. Initially, \mathcal{S} is given an instance (y, e, n) of RSA inversion problem defined in Definition 7 and simulates the environment as follows.

(i) H_1 Query of \mathcal{O}_{H_1}

Initially, every blank record in \mathcal{L}_{H_1} can be represented as (\perp, \perp, \perp) . When \mathcal{A} sends m for querying the hash value $H_1(m)$, \mathcal{S} will check the list \mathcal{L}_{H_1} :

- (a) if $m = m_i$ for some i , then \mathcal{S} retrieves the corresponding ρ_i and returns it to \mathcal{A} ;
- (b) else if $m = H_1(m_i)$ and $H_1^2(m_i) \neq \perp$ for some i , then \mathcal{S} retrieves the corresponding ς and returns $(\varsigma^e \pmod{n})$ to \mathcal{A} ;
- (c) else if $m = H_1(m_i)$ and $H_1^2(m_i) = \perp$ for some i , then \mathcal{S} selects a random $\varsigma \in \mathbb{Z}_n$, returns $(\varsigma^e \pmod{n})$ to \mathcal{A} , and then fills the record $(m_i, H_1(m_i), \perp)$ as $(m_i, H_1(m_i), \varsigma)$ in \mathcal{L}_{H_1} ;
- (d) otherwise, \mathcal{S} selects a random $\rho \in \mathbb{Z}_n$, records (m, ρ, \perp) in \mathcal{L}_{H_1} , and returns ρ to \mathcal{A} .

(ii) H_2 Query of \mathcal{O}_{H_2}

When \mathcal{A} asks for H_2 query by sending D to \mathcal{S} , \mathcal{S} will look up the list \mathcal{L}_{H_2} :

- (a) if $D = D_i$ for some i , the corresponding τ will be retrieved and \mathcal{S} will send $(\tau^e \pmod{n})$ back to \mathcal{A} ;
- (b) otherwise, \mathcal{S} will select a random $\tau \in \mathbb{Z}_n$, record (D, τ) in \mathcal{L}_{H_2} , and return $(\tau^e \pmod{n})$ back to \mathcal{A} .

(iii) H_3 Query of \mathcal{O}_{H_3}

While \mathcal{A} sends (σ, D) to \mathcal{S} for $H_3(\sigma \parallel D)$, \mathcal{S} will look up the list \mathcal{L}_{H_3} :

- (a) if $(\sigma, D) = (\sigma_i, D_i)$ for some i , the corresponding $H_3(\sigma_i \parallel D_i)$ will be retrieved and returned to \mathcal{A} ;
- (b) otherwise, \mathcal{S} will select a random $\eta \in \mathbb{Z}_n$, set $H_3(\sigma \parallel D) = (\eta^e y \pmod{n})$, record $((\sigma, D), \eta, H_3(\sigma \parallel D))$ in \mathcal{L}_{H_3} , and return $H_3(\sigma \parallel D)$ back to \mathcal{A} .

(iv) E-Cash Producing Query of $\mathcal{O}_{\mathcal{S}}$

Let ℓ_{D_i} be a counter to record the number of queries on each expiration date D_i , which is initialized by 0. When \mathcal{A} sends (α, ϵ, D) to \mathcal{S} , \mathcal{S} will do the following steps:

- (1) decrypt ϵ , obtain (k, ID) ;
- (2) randomly select r_j and prepare $\sigma = \hat{E}_{pk_j}(\text{ID} \parallel r_j)$;
- (3) choose $\eta \in_R \mathbb{Z}_n$, set $H_3(\sigma \parallel D) = (\alpha \eta^e \pmod{n})$, and store $((\sigma, D), \perp, (\alpha \eta^e \pmod{n}))$ and (σ, D) in \mathcal{L}_{H_3} and \mathcal{L}_x , respectively;
- (4) select $b \in_R \mathbb{Z}_n^*$ and compute $\beta = (b^e \alpha \eta^e)^{-1} \pmod{n}$;
- (5) retrieve or assign τ such that $H_2(D) = (\tau^e)$ as the \mathcal{O}_{H_2} query described above;
- (6) compute $t \equiv (\alpha \beta \tau^e)^d \equiv ((b \eta)^{-1} \tau) \pmod{n}$;
- (7) set $\ell_D = \ell_D + 1$ and return $(t, \tilde{E}_k(b, \sigma, r_j))$ back to \mathcal{A} .

Eventually, assume that \mathcal{A} can successfully output $\ell_{D'} + 1$ e-cash tuples for some expiration date D'

$$\{(s_1, m_1, \sigma_1, D') \cdots (s_{\ell_{D'}+1}, m_{\ell_{D'}+1}, \sigma_{\ell_{D'}+1}, D')\} \quad (20)$$

such that $s_i^e H_1^2(m_i) H_3(\sigma_i \parallel D') = H_2(D') \pmod{n}, \forall i, 1 \leq i \leq \ell_{D'} + 1$, after $\ell_{D'}$ times to query $\mathcal{O}_{\mathcal{S}}$ on D' , with nonnegligible probability $\epsilon_{\mathcal{A}}$.

Assume some (σ_i, D') , $1 \leq i \leq \ell_{D'} + 1$, is not recorded in \mathcal{L}_x ; then by the \mathcal{L}_{H_1} , \mathcal{L}_{H_2} , and \mathcal{L}_{H_3} , \mathcal{S} can compute and retrieve

$$\begin{aligned} (s_i)^e &\equiv (H_1^2(m_i) H_3(\sigma_i \parallel D'))^{-1} H_2(D') \\ &\equiv ((\varsigma_i^e) (\eta_i^e y))^{-1} (\tau_i^e) \pmod{n}, \\ x &\equiv y^d \equiv (s_i \varsigma_i \eta_i)^{-1} \tau_i \pmod{n} \end{aligned} \quad (21)$$

and solve the RSA inversion problem with nonnegligible probability at least $\epsilon_{\mathcal{A}}$. \square

4.3. E-Cash Conditional-Traceability. In this section, we will prove that the ID information embedded in e-cash(s) cannot be replaced or moved out by any user against being traced after some misbehavior or criminals. The details of our proof model are illustrated in Figure 8.

Definition 10 (Tampering Game (TG)). Let $l_k \in \mathbb{N}$ be a security parameter and \mathcal{A} be an adversary in $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$. $\mathcal{O}_{\mathcal{S}}$ is an oracle which plays the role of bank in $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$

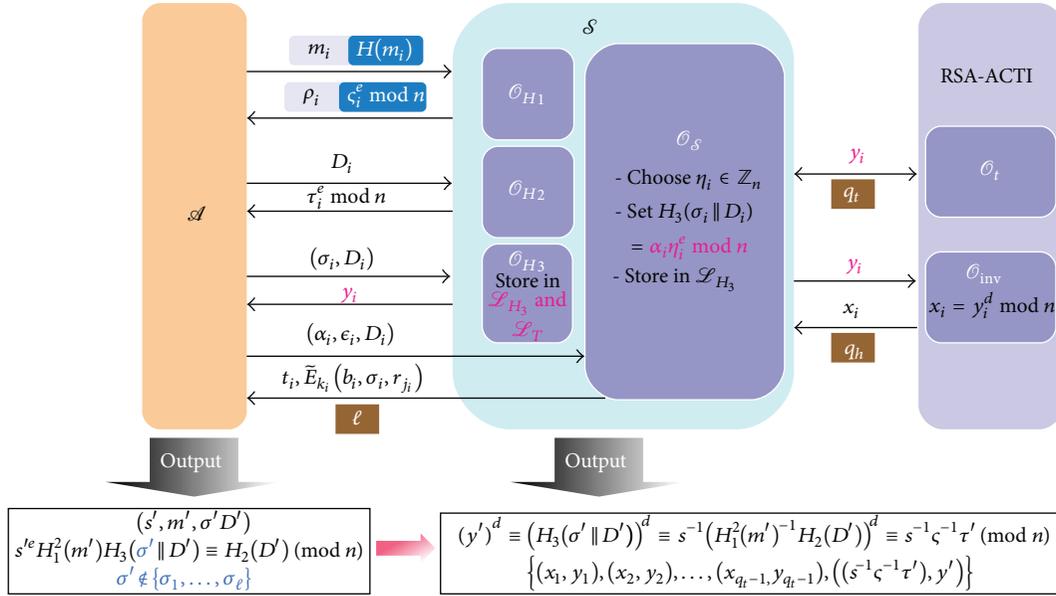


FIGURE 8: The proof model of TG.

Experiment $\text{Exp}_{\mathcal{A}}^{\text{TG}}(l_k)$
 $(pk_j, sk_j, g_1, g_2, e_b, d_b, p_b, q_b, n_b, H_1, H_2, H_3, H_4, H_5) \leftarrow \text{Setup}(l_k)$
 $(s', m', \sigma', D') \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{S}}} (pk_{\text{TA}}, e_R, n_R, H_1, H_2)$
 $\{\sigma_1, \dots, \sigma_\ell\} \leftarrow \mathcal{O}_{\mathcal{S}}$
 if the following two checks are true, **return 1**;
 (i) $\sigma' \notin \{\sigma_1, \dots, \sigma_\ell\}$
 (ii) $s'^e H_1^2(m') H_3(\sigma' \parallel D') = H_2(D') \pmod n$
 else **return 0**;

ALGORITHM 4

to record parameters from the queries of \mathcal{A} and issue e-cash(s) (i.e., (s, m, σ, D) , where $m = (w_1, y_1, w_2, y_2, r_3, D)$) accordingly. \mathcal{A} is allowed to query $\mathcal{O}_{\mathcal{S}}$ for ℓ times; consider Algorithm 4.

\mathcal{A} wins the game if the probability $\Pr[\text{Exp}_{\mathcal{A}}^{\text{TG}}(k) = 1]$ of \mathcal{A} is nonnegligible.

Definition 11 (E-Cash Traceability). If there exists no probabilistic polynomial-time adversary who can win the tracing game TG, then $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ satisfies the E-Cash Traceability.

Definition 12 (Alternative Formulation of RSA Known-Target Inversion Problem (RSA-AKTI)). Let $k \in \mathbb{N}$ be a security parameter and \mathcal{A} be an adversary who is allowed to access the RSA-inversion oracle \mathcal{O}_{inv} and the target oracle \mathcal{O}_t . \mathcal{A} is allowed to query \mathcal{O}_t and \mathcal{O}_{inv} for q_t and q_h times ($q_h < q_t$), respectively. Consider Algorithm 5.

We say \mathcal{A} breaks the RSA-AKTI problem if the probability $\Pr[\text{Exp}_{\mathcal{A}}^{\text{RSA-AKTI}}(k) = 1]$ of \mathcal{A} is nonnegligible.

Theorem 13. For a polynomial-time adversary \mathcal{A} who can win the tracing game TG with nonnegligible probability, there exists

Experiment $\text{Exp}_{\mathcal{A}}^{\text{RSA-AKTI}}(k)$
 $(N, e, d) \xleftarrow{R} \text{KeyGen}(k)$
 $(y_1, \dots, y_{q_t}) \leftarrow \mathcal{O}_t(N, e, k)$
 $\{(x_1, y_1), \dots, (x_{q_t}, y_{q_t})\} \leftarrow \mathcal{A}^{\mathcal{O}_{\text{inv}}, \mathcal{O}_t}(N, e, k)$
 if $x_i^e \equiv y_i \pmod N, \forall i \in \{1, \dots, q_t\}$, **return 1**;
 else **return 0**;

ALGORITHM 5

another adversary \mathcal{S} who can break the RSA-AKTI problem with nonnegligible probability.

Proof. \mathcal{S} simulates the environment of $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ by controlling three hash oracles, \mathcal{O}_{H_1} , \mathcal{O}_{H_2} , \mathcal{O}_{H_3} , to respond hash queries and an e-cash producing oracle $\mathcal{O}_{\mathcal{S}}$ of $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ to respond e-cash producing queries from \mathcal{A} , respectively, in the random oracle model. Eventually, \mathcal{S} will take advantage of \mathcal{A} 's capability to solve RSA-AKTI problem. Then, for consistency, \mathcal{S} maintains three lists \mathcal{L}_{H_1} , \mathcal{L}_{H_2} , and \mathcal{L}_{H_3} to record every response of \mathcal{O}_{H_1} , \mathcal{O}_{H_2} , and \mathcal{O}_{H_3} , respectively.

Besides, in the proof model, \mathcal{S} is allowed to query the oracles \mathcal{O}_{inv} (i.e., $(\cdot)^d$) and \mathcal{O}_t of the RSA-AKTI problem defined in Definition 12 for helping \mathcal{S} produce valid e-cash(s) and the corresponding verifying key is (e, n) .

Here we will do the simulation for game TG to prove that $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ satisfies the e-cash traceability. Details are described as follows.

(i) H_1 Query of \mathcal{O}_{H_1}

Initially, every blank record in \mathcal{L}_{H_1} can be represented as (\perp, \perp, \perp) . When \mathcal{A} sends m for querying the hash value $H_1(m)$, \mathcal{S} will check the list \mathcal{L}_{H_1} :

- (a) if $m = m_i$ for some i , then \mathcal{S} retrieves the corresponding $H_1(m_i)$ and return it to \mathcal{A} ;
- (b) else if $m = H_1(m_i)$ and $H_1^2(m_i) \neq \perp$ for some i , then \mathcal{S} retrieves the corresponding ζ_i and returns $(\zeta_i^e \bmod n)$ to \mathcal{A} ;
- (c) else if $m = H_1(m_i)$ and $H_1^2(m_i) = \perp$ for some i , then \mathcal{S} chooses $\zeta \in_R \mathbb{Z}_n$, sets $H_1^2(m_i) = (\zeta^e \bmod n)$, and returns $H_1^2(m_i)$ to \mathcal{A} then fills the original record $(m_i, H_1(m_i), \perp)$ as $(m_i, H_1(m_i), \zeta)$ in \mathcal{L}_{H_1} ;
- (d) otherwise, \mathcal{S} selects a random $\rho \in \mathbb{Z}_n$, sets $H_1(m_i) = \rho$, records $(m, H_1(m_i), \perp)$ in \mathcal{L}_{H_1} , and returns ρ to \mathcal{A} .

(ii) H_2 Query of \mathcal{O}_{H_2}

When \mathcal{A} asks for H_2 query by sending D to \mathcal{S} , \mathcal{S} will look up the list \mathcal{L}_{H_2} :

- (a) if $D = D_i$ for some i , the corresponding τ will be retrieved and \mathcal{S} will send $(\tau^e \bmod n)$ back to \mathcal{A} ;
- (b) otherwise, \mathcal{S} will select a random $\tau \in \mathbb{Z}_n$, record (D, τ) in \mathcal{L}_{H_2} , and return $(\tau^e \bmod n)$ back to \mathcal{A} .

(iii) H_3 Query of \mathcal{O}_{H_3}

While \mathcal{A} sends (σ, D) to \mathcal{S} for $H_3(\sigma)$, \mathcal{S} will look up the list \mathcal{L}_{H_3} :

- (a) if $(\sigma, D) = (\sigma_i, D_i)$ for some i , the corresponding y_i will be retrieved and returned to \mathcal{A} ;
- (b) otherwise, \mathcal{S} will query \mathcal{O}_t to get an instance y ; record y and $((\sigma, D), y)$ in \mathcal{L}_T and \mathcal{L}_{H_3} , respectively;
- (c) return y back to \mathcal{A} .

(iv) E-Cash Producing Query of $\mathcal{O}_{\mathcal{S}}$

While \mathcal{A} sends (α, ϵ, D) to \mathcal{S} , \mathcal{S} will do the following steps:

- (1) decrypt ϵ , obtain (k, ID) ;
- (2) randomly select r_j and prepare $\sigma = \hat{E}_{pk_j}(\text{ID} \parallel r_j)$;
- (3) choose $\eta \in_R \mathbb{Z}_n$, set $H_3(\sigma \parallel D) = (\alpha\eta^e \bmod n)$, and store $((\sigma, D), H_3(\sigma \parallel D))$ in \mathcal{L}_{H_3} ;

- (4) select $b \in_R \mathbb{Z}_n^*$ and compute $\beta = (b^e \alpha \eta^e)^{-1} \bmod n$;
- (5) retrieve or assign τ such that $H_2(D) = (\tau^e)$ as the \mathcal{O}_{H_2} query described above;
- (6) compute $t \equiv (\alpha\beta\tau^e)^d \equiv ((b\eta)^{-1}\tau) \pmod n$;
- (7) return $(t, \hat{E}_k(b, \sigma, r_j))$ back to \mathcal{A} .

Assume that \mathcal{A} can successfully output an e-cash tuples (s', m', σ', D') , where σ' never appears as a part for some $\mathcal{O}_{\mathcal{S}}$ query such that $s'^e H_1^2(m') H_3(\sigma' \parallel D') \equiv H_2(D') \pmod n$; then by \mathcal{L}_{H_1} , \mathcal{L}_{H_2} , and \mathcal{L}_{H_3} , \mathcal{S} can derive

$$\begin{aligned} (y')^d &\equiv (H_3(\sigma' \parallel D'))^d \equiv s'^{-1} (H_1^2(m')^{-1} H_2(D'))^d \\ &\equiv s'^{-1} \zeta'^{-1} \tau' \pmod n. \end{aligned} \tag{22}$$

Let $|\mathcal{L}_T| = q_t$ and $\mathcal{L}_T = \{y_1, \dots, y_{q_t}\}$. \mathcal{S} sends $y_i \in (\mathcal{L}_T - \{y'\})$, $1 \leq i \leq (q_t - 1)$, to \mathcal{O}_{inv} and obtains $q_t - 1$ x_i such that $x_i = y_i^d \bmod n$.

Eventually \mathcal{S} can output q_t RSA-inversion instances

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_{q_t-1}, y_{q_t-1}), ((s'^{-1} \zeta'^{-1} \tau'), y')\} \tag{23}$$

after querying \mathcal{O}_{inv} for q_h times, where $q_h = q_t - 1 < q_t$ and thus, it breaks the RSA-AKTI problem with nonnegligible probability at least $\epsilon_{\mathcal{A}}$. \square

4.4. E-Cash No-Swindling. In typical online e-cash transactions, when an e-cash has been spent in previous transactions, another spending will be detected immediately owing to the double-spending check procedure. However, in an offline e-cash model, the merchant may accept a transaction involving a double-spent e-cash first and then do the double-spending check later. In this case, the original owner of the e-cash may suffer from loss. Therefore, a secure offline e-cash scheme should guarantee the following two events.

- (i) No one, except the real owner, can spend a fresh and valid offline e-cash successfully.
- (ii) No one can double spend an e-cash successfully.

Roughly, it can be referred to as *e-cash no-swindling* property. In this section, we will define the no-swindling property and formally prove that our scheme is secure against swindling attacks.

Definition 14 (Swindling Game in $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$). Let $l_k \in \mathbb{N}$ be a security parameter and \mathcal{A} be an adversary in $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$. \mathcal{O}_B is an oracle issuing generic e-cash(s) (i.e., $(s, y_1, w_1, x_2, r_2, r_3, \sigma, D)$) of $\mathcal{D}\mathcal{A}\mathcal{O}\mathcal{E}\mathcal{C}\mathcal{S}$ to \mathcal{A} . \mathcal{O}_{off} is an oracle to show the expanding form $(s, y_1, w_1, x_2, r_2, r_3, \sigma, D, r_s, s')$ for the payment according to the input (s, m, σ, D) . Consider the two experiments SWG-1 and SWG-2 shown in Algorithms 6 and 7, respectively.

\mathcal{A} wins the game if the probability $\Pr[\text{Exp}_{\mathcal{A}}^{\text{SWG-1}}(l_k) = 1]$ or $\Pr[\text{Exp}_{\mathcal{A}}^{\text{SWG-2}}(l_k) = 1]$ of \mathcal{A} is nonnegligible.

Experiment $\text{Exp}_{\mathcal{A}}^{\text{SWG-1}}(I_k)$

$(pk_j, sk_j, g_1, g_2, e_b, d_b, p_b, q_b, n_b, p, q, H_1, H_2, H_3, H_4, H_5) \leftarrow \text{Setup}(I_k)$
 $\{(s, w_1, y_1, w_2, y_2, r_3, \sigma, D, r_u, r_s, s')\} \leftarrow \mathcal{A}^{\mathcal{O}_B, \mathcal{O}_{\text{off}}}(pk_j, g_1, g_2, e_b, n_b, p, q, H_1, H_2, H_3, H_4, H_5)$
 if the following checks are true, **return 1**;
 (i) $s^{e_b} H_1^2(y^{H_4(r_u \| r_s)} g^{s'}) \bmod p \parallel y_1 \parallel w_2 \parallel y_2 \parallel D \parallel r_3) H_3(\sigma \parallel D) = H_2(D) \bmod n_b$;
 (ii) $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D)$ never be a query to \mathcal{O}_{off}
 else **return 0**;

ALGORITHM 6: Experiment SWG-1.

Experiment $\text{Exp}_{\mathcal{A}}^{\text{SWG-2}}(I_k)$

$(pk_j, sk_j, g_1, g_2, e_b, d_b, p_b, q_b, n_b, p, q, H_1, H_2, H_3, H_4, H_5) \leftarrow \text{Setup}(I_k)$
 $\{(s, w_1, y_1, w_2, y_2, r_3, \sigma, D, r_u, r_s, s')\} \leftarrow \mathcal{A}^{\mathcal{O}_B, \mathcal{O}_{\text{off}}}(pk_j, g_1, g_2, e_b, n_b, p, q, H_1, H_2, H_3, H_4, H_5)$
 if the following checks are true, **return 1**;
 (i) $s^{e_b} H_1^2(y^{H_4(r_u \| r_s)} g^{s'}) \bmod p \parallel y_1 \parallel w_2 \parallel y_2 \parallel D \parallel r_3) H_3(\sigma \parallel D) = H_2(D) \bmod n_b$;
 (ii) $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D)$ is allowed to be queried to \mathcal{O}_{off} for once;
 (iii) $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D, r_s, s')$ is not obtained from \mathcal{O}_{off}
 else **return 0**;

ALGORITHM 7: Experiment SWG-2.

Definition 15 (E-Cash No-Swindling). If there exists no probabilistic polynomial-time adversary who can win the swindling game defined in Definition 14, then \mathcal{DAOSES} satisfies e-cash no-swindling.

Theorem 16. For a polynomial-time adversary \mathcal{A} who can win the swindling game SWG with nonnegligible probability, there exists another adversary \mathcal{S} who can solve the discrete logarithm problem with nonnegligible probability.

Proof. Consider the swindling game defined in Definition 14. \mathcal{S} simulates the environment by controlling the hash oracles, \mathcal{O}_{H_4} , to respond hash queries on H_4 of \mathcal{DAOSES} in the random oracle model. Eventually, \mathcal{S} will take advantage of \mathcal{A} 's capability to solve the discrete logarithm problem. Then, for consistency, \mathcal{S} maintains a list \mathcal{L}_{H_4} to record every response of \mathcal{O}_{H_4} . \mathcal{S} is given all parameters $(pk_j, sk_j, g_1, g_2, e_b, d_b, p_b, q_b, n_b, p, q, H_1, H_2, H_3, H_4, H_5)$ of \mathcal{DAOSES} and an instance y^* of discrete logarithm problem (i.e., $y^* = g^{x^*} \bmod p$). Here we will describe the simulations for the two experiments $\text{Exp}_{\mathcal{A}}^{\text{SWG-1}}$ and $\text{Exp}_{\mathcal{A}}^{\text{SWG-2}}$, individually.

The simulation for $\text{Exp}_{\mathcal{A}}^{\text{SWG-1}}$ is illustrated in Figure 9 and each oracle is constructed as follows.

(i) Oracle \mathcal{O}_B

Initially, \mathcal{S} guesses that the generic e-cash produced from ν th query will be the attack target. When \mathcal{A} sends i th query to \mathcal{O}_B for an e-cash, \mathcal{O}_B will do the following:

- (a) select $r_1, x_1, r_3 \in_R \mathbb{Z}_q$ and $y_2, w_2 \in_R \mathbb{Z}_p$;
- (b) if $i = \nu$,
 - (1) compute $(w_1 = (y^*)^{r_1} \bmod p)$ and $(y_1 = g^{x_1} \bmod p)$;

(c) if $i \neq \nu$,

- (1) compute $(w_1 = g^{r_1} \bmod p)$ and $(y_1 = g^{x_1} \bmod p)$;

- (d) prepare $s = ((H_1^2(m)H_3(\sigma \parallel D))^{-1}H_2(D))^{d_b} \bmod n_b$, where $m = (w_1, y_1, w_2, y_2, r_3, D)$;

- (e) record $(i, (s, m, \sigma, D), (r_1, x_1))$ in list \mathcal{L}_B and return (s, m, σ, D) to \mathcal{A} .

(ii) Oracle \mathcal{O}_{off}

When \mathcal{A} sends a valid e-cash tuple $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D, r_s)$ to \mathcal{O}_{off} , it will look up the list \mathcal{L}_B :

- (a) if $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D)$ exists with prefix index ν , then abort;

- (b) otherwise, \mathcal{O}_{off} will retrieve the corresponding (r_1, x_1) ; choose a random r_u , compute $u = H_4(r_u \parallel r_s)$ and $(s' = r_1 - ux_1 \bmod q)$, and send $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D, r_u, r_s, s')$ back to \mathcal{A} .

Assume that \mathcal{A} can successfully output a valid offline e-cash expansion tuple $(s^*, w_1^*, y_1^*, w_2^*, y_2^*, r_3^*, \sigma^*, D^*, r_u^*, r_s^*, s'^*)$, where $(s^*, w_1^*, y_1^*, w_2^*, y_2^*, r_3^*, \sigma^*, D^*)$ is prefixed with ν and postfixed with (r_1^*, x_1^*) in \mathcal{L}_B . Then, since $w_1^* = y_1^{*H_4(r_u^* \| r_s^*)} g^{s'^*} \bmod p$ and $w_1^* = (y^*)^{r_1^*}$, \mathcal{S} can derive

$$x^* = (r_1^*)^{-1} (x_1^* H_4(r_u^* \| r_s^*) + s'^*) \bmod q \quad (24)$$

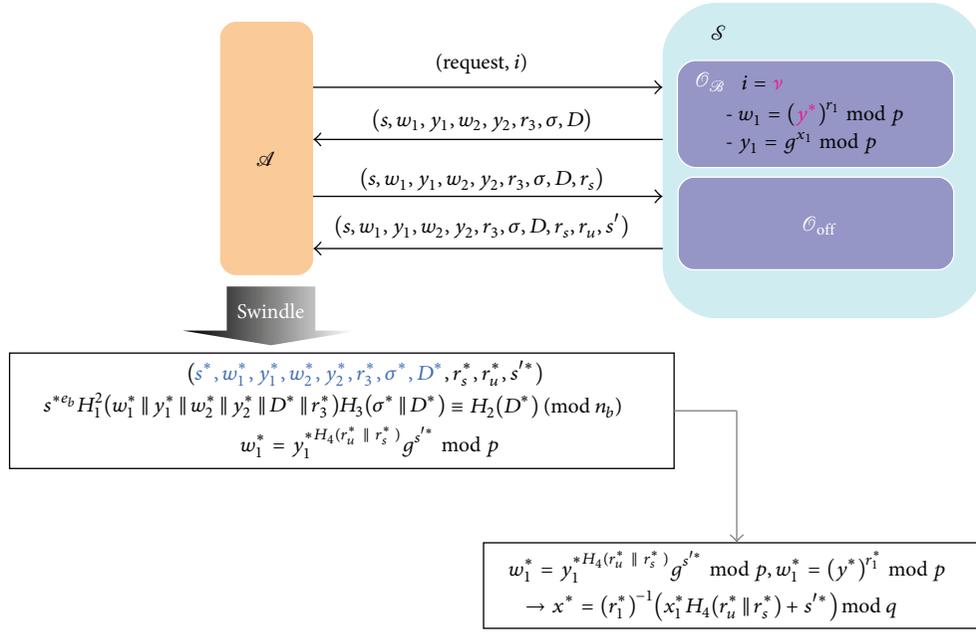


FIGURE 9: The proof model of SWG-1.

and solve the discrete logarithm problem with nonnegligible probability at least $(1/q_{\mathcal{O}_B})\epsilon_{\mathcal{A}}$, where $q_{\mathcal{O}_B}$ is the total number of \mathcal{O}_B query.

The simulation for $\text{Exp}_{\mathcal{A}}^{\text{SWG-2}}$ is illustrated in Figure 10 and each oracle is constructed as follows.

(i) Oracle \mathcal{O}_B

Initially, \mathcal{S} guesses that the generic e-cash produced from ν th query will be the attack target. When \mathcal{A} sends i th query to \mathcal{O}_B for an e-cash, \mathcal{O}_B will do the followings.

(a) if $i = \nu$:

- (1) select $s', u, x_1, r_3 \in_R \mathbb{Z}_q$ and $y_2, w_2 \in_R \mathbb{Z}_p$;
- (2) compute $(y_1 = (y^*)^{x_1} \bmod p)$ and $(w_1 = y_1^u g^{s'} \bmod p)$;
- (3) prepare $s = ((H_1^2(m)H_3(\sigma \parallel D))^{-1} H_2(D))^{d_b} \bmod n_b$, where $m = (w_1, y_1, w_2, y_2, r_3, D)$;
- (4) record $(i, (s, m, \sigma, D), (u, s'))$ in list $\mathcal{L}_{\mathcal{O}_B}$;

(b) if $i \neq \nu$:

- (1) select $r_1, x_1, r_3 \in_R \mathbb{Z}_q$ and $y_2, w_2 \in_R \mathbb{Z}_p$;
- (2) compute $(w_1 = g^{r_1} \bmod p)$ and $(y_1 = g^{x_1} \bmod p)$;
- (3) prepare $s = ((H_1^2(m)H_3(\sigma \parallel D))^{-1} H_2(D))^{d_b} \bmod n_b$, where $m = (w_1, y_1, w_2, y_2, r_3, D)$;
- (4) record $(i, (s, m, \sigma, D), (r_1, x_1))$ in list $\mathcal{L}_{\mathcal{O}_B}$;

(c) return (s, m, σ, D) to \mathcal{A} .

(ii) Oracle \mathcal{O}_{off}

A status parameter sta is initialized by 0. When \mathcal{A} sends a valid e-cash tuple $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D, r_s)$ to \mathcal{O}_{off} , it will look up the list $\mathcal{L}_{\mathcal{O}_B}$:

(a) if $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D)$ exists with prefix index ν and $\text{sta} = 0$, \mathcal{O}_{off} will perform the following procedures:

- (1) set $\text{sta} = 1$
- (2) retrieve the corresponding (u, s') from $\mathcal{L}_{\mathcal{O}_B}$ and choose a random r_u ;
- (3) set $H_4(r_u \parallel r_s) = u$ and record $((r_u \parallel r_s), u)$ in \mathcal{L}_H ;
- (4) record $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D, r_u, r_s, s')$ in list \mathcal{L}_{off} ;
- (5) send $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D, r_u, r_s, s')$ back to \mathcal{A} ;

(b) if $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D)$ exists with prefix index $\neq \nu$, \mathcal{O}_{off} will retrieve the corresponding (r_1, x_1) , choose random r_u and u , set $H_4(r_u \parallel r_s) = u$, record $((r_u \parallel r_s), u)$ in \mathcal{L}_H , compute $(s' = r_1 - ux_1 \bmod q)$, and send $(s, w_1, y_1, w_2, y_2, r_3, \sigma, D, r_u, r_s, s')$ back to \mathcal{A} .

(c) Otherwise, abort.

(iii) Oracle \mathcal{O}_{H_4}

While \mathcal{A} sends $(r_u \parallel r_s)$ to query for $H_4(r_u \parallel r_s)$, \mathcal{O}_{H_4} will check the list \mathcal{L}_H :

(a) if $(r_u \parallel r_s)$ exists as the prefix of some record, \mathcal{O}_{H_4} will retrieve the corresponding u and return it to \mathcal{A} ;

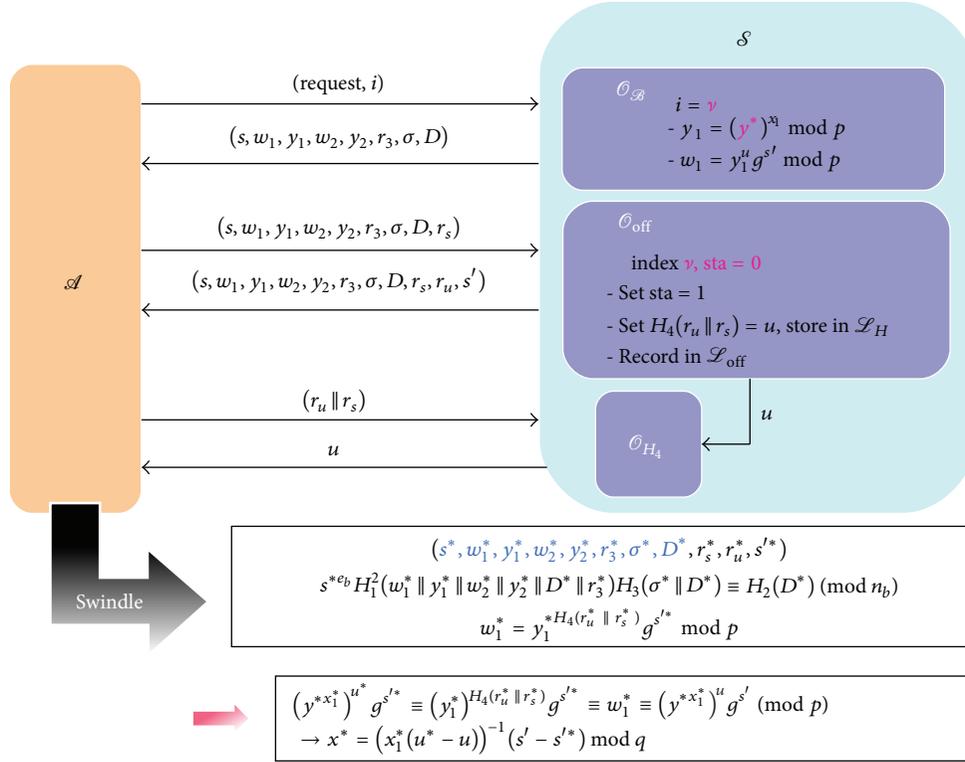


FIGURE 10: The proof model of SWG-2.

(b) otherwise, \mathcal{O}_{H_4} will choose a random u , record $((r_u \| r_s), u)$ in \mathcal{L}_H , and return u to \mathcal{A} .

Assume that \mathcal{A} can successfully output a valid offline e-cash expansion tuple $(s^*, w_1^*, y_1^*, w_2^*, y_2^*, r_3^*, \sigma^*, D^*, r_u^*, r_s^*, s'^*)$, where $(s^*, w_1^*, y_1^*, w_2^*, y_2^*, r_3^*, \sigma^*, D^*)$ is prefixed with ν and postfixed with (u, s') in $\mathcal{L}_{\mathcal{B}}$ and $H_4(r_u^* \| r_s^*) \neq u$.

Then, via $\mathcal{L}_{\mathcal{H}}$, since

$$\begin{aligned} (y^{*x_1^*})^{u^*} g^{s'^*} &\equiv (y_1^*)^{H_4(r_u^* \| r_s^*)} g^{s'^*} \equiv w_1^* \\ &\equiv (y^{*x_1^*})^u g^{s'} \pmod{p}, \end{aligned} \tag{25}$$

\mathcal{S} can derive

$$x^* = (x_1^*(u^* - u))^{-1} (s' - s'^*) \pmod{q} \tag{26}$$

and solve the discrete logarithm problem with nonnegligible probability at least $(1/q_{\mathcal{O}_B})\epsilon_{\mathcal{A}}$, where $q_{\mathcal{O}_B}$ is the total number of \mathcal{O}_B query. \square

Summarize the proof models for the two experiments shown above, if there exists a polynomial-time adversary who can win the swindling game with nonnegligible probability, then there exists another one who can solve the discrete logarithm problem with nonnegligible probability. It implies that there exists no p.p.t. adversary who can win the swindling game, and our proposed offline e-cash scheme \mathcal{DAOES} satisfies no-swindling property.

5. E-Cash Advanced Features and Performance Comparisons

In this section, we compare the e-cash features and performance of our proposed scheme with other schemes given in [9, 13–15, 21, 22, 27, 38–40]. We analyze the features and performance of the aforementioned schemes and form a table (Table 1) for the summary.

5.1. Features Comparisons. All the schemes mentioned above fulfill the basic security requirements stated in Section 1, which are anonymity, unlinkability, unforgeability, and no double-spending. Besides these features, there can be other advanced features on an e-cash system discussed in the literatures. We focus on three other advanced features, which are traceability, date attachability, and no-swindling, and we compare the proposed scheme with the aforementioned schemes.

We also propose an e-cash renewal protocol for users to exchange a new valid e-cash with their unused but expired e-cash(s); therefore, users do not have to deposit the e-cash before it expires and withdraw a new e-cash again. Our proposed e-cash renewal protocol reduces the computation cost by 49.5% as compared to withdrawal and deposit protocols, which is almost half of the effort of getting a new e-cash, at the user side. It does a great help to the users since their devices usually have a weaker computation capability, such as smart phones.

TABLE I: Advanced features and performance comparisons.

	[38]	[14]	[15]	[9]	[21]	[22]	[39]	[40]	[13]	[27]
On/off-line	Off	Off	Off	On	Off	Off	Off	Off	On	Off
Conditional-traceability	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Date attachability	Yes	No	No	Yes	Yes	No	No	No	No	Yes
No-swindling	Yes	No	No	—	No	Yes	No	No	—	No
Renewal protocol	Yes	Yes	—	No	Yes	Yes	—	—	—	Yes
Formal proof	Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	No
				Performance						
Transaction cost*	$5E + 7M$ $+7H+1inv$ $+1A$ $\approx 1454M$	$14E + 14M$ $+1H + 5A$ $\approx 3375M$	$23E + 14M$ $+1A$ $\approx 5534M$	$2E + 2M$ $+2H$ $\approx 966M$	$5E + 9M$ $+1H + 1inv$ $+2A$ $\approx 1450M$	$2E \approx 480M$	$18E + 15M$ $+2H + 8A$ $\approx 4337M$	$31E + 22M$ $+6H + 10A$ $\approx 7468M$	$22E + 11M$ $+4A$ $\approx 5291M$	$6E + 8M$ $+1H$ $\approx 1449M$
Communication cost \diamond	1092	576	939	769	644	300	828	968	1536	728

According to [41], $H \approx M$, $E \approx inv \approx 240M$.

E : a modular exponentiation; M : a modular multiplication; H : a hash operation; zkp : a zero-knowledge proof.

A : a modular addition; inv : a modular inversion.

* The computation cost of withdrawal and payment protocols at user side.

\diamond The communication cost of each transaction at user side in bytes.

5.2. *Performance Comparisons.* According to [41], we can summarize and induce the computation cost of all operations as follows. The computation cost of a modular exponentiation computation is about 240 times of the computation cost of a modular multiplication computation, while the computation cost of a modular inversion almost equals to that of a modular exponentiation. Also, the computation cost of a hash operation almost equals to that of a modular multiplication.

With the above assumptions, the total computation cost of users during withdrawal and payment phases of our proposed scheme can be induced as 1452 times of a modular multiplication computation, while other works [9, 13–15, 21, 22, 27, 38–40] need 3375, 1448, 5534, 966, 1450, 480, 4337, 7468, 5291, and 1449 times of a modular multiplication computation to finish withdrawal and payment phases at the user ends.

According to [15], we assume the RSA parameters n , p , q are 1024, 512, and 512 bits, respectively. We adopt AES and SHA-1 as the symmetric cryptosystem and one-way hash function used in all protocols, respectively; therefore, the signed message and hash message are in 128 and 160 bits, respectively. We assume the expiration date is in 32 bits.

With the above assumptions, we compute the communication cost of each offline transaction, withdrawal, and payment, at the user side. Our scheme needs 2048 bits for withdrawing an e-cash and 6688 bits for spending an e-cash, which is 1092 bytes for each transaction.

The details of the comparisons are summarized in Table 1.

6. Conclusion

In this paper, we have presented earlier a provably secure offline electronic cash scheme with an expiration date and a deposit date attached to it. Besides, we have also designed an e-cash renewal protocol, where users can exchange their unused and expired e-cash(s) for new ones more efficiently. Compared with other similar works, our scheme is efficient from the aspect of considering computation cost of the user side and satisfying all security properties, simultaneously. Except for anonymity, unlinkability, unforgeability, and no double-spending, we also formally prove that our scheme achieves conditional-traceability and no-swindling. Not only does our scheme help the bank to manage their huge databases against unlimited growth, but also it strengthens the preservation of users' privacy and rights as well.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Science Council of Taiwan under Grants NSC 102-2219-E-110-002,

NSYSU-KMU Joint Research Project (NSYSUKMU 2013-1001), and Aim for the Top University Plan of the National Sun Yat-sen University and Ministry of Education, Taiwan.

References

- [1] H. Chen, P. P. Y. Lam, H. C. B. Chan, T. S. Dillon, J. Cao, and R. S. T. Lee, "Business-to-consumer mobile agent-based internet commerce system (MAGICS)," *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 37, no. 6, pp. 1174–1189, 2007.
- [2] S. C. Fan and Y. L. Lai, "A study on e-commerce applying in Taiwan's restaurant franchise," in *Proceedings of the IET International Conference on Frontier Computing. Theory, Technologies and Applications*, pp. 324–329, August 2010.
- [3] D. R. W. Holton, I. Nafea, M. Younas, and I. Awan, "A class-based scheme for E-commerce web servers: formal specification and performance evaluation," *Journal of Network and Computer Applications*, vol. 32, no. 2, pp. 455–460, 2009.
- [4] Z. Jie and X. Hong, "E-commerce security policy analysis," in *Proceedings of the International Conference on Electrical and Control Engineering (ICECE '10)*, pp. 2764–2766, June 2010.
- [5] D. R. Liuy and T. F. Hwang, "An agent-based approach to flexible commerce in intermediary-Centric electronic markets," *Journal of Network and Computer Applications*, vol. 27, no. 1, pp. 33–48, 2004.
- [6] S. J. Lin and D. C. Liu, "An incentive-based electronic payment scheme for digital content transactions over the Internet," *Journal of Network and Computer Applications*, vol. 32, no. 3, pp. 589–598, 2009.
- [7] H. Wang, Y. Zhang, J. Cao, and V. Varadharajan, "Achieving Secure and Flexible M-Services through Tickets," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 33, no. 6, pp. 697–708, 2003.
- [8] C. Yue and H. Wang, "Profit-aware overload protection in E-commerce Web sites," *Journal of Network and Computer Applications*, vol. 32, no. 2, pp. 347–356, 2009.
- [9] C. C. Chang and Y. P. Lai, "A flexible date-attachment scheme on e-cash," *Computers and Security*, vol. 22, no. 2, pp. 160–166, 2003.
- [10] C. L. Chen and J. J. Liao, "A fair online payment system for digital content via subliminal channel," *Electronic Commerce Research and Applications*, vol. 10, no. 3, pp. 279–287, 2011.
- [11] C. I. Fan, W. K. Chen, and Y. S. Yeh, "Date attachable electronic cash," *Computer Communications*, vol. 23, no. 4, pp. 425–428, 2000.
- [12] C. I. Fan and W. Z. Sun, "Efficient encoding scheme for date attachable electronic cash," in *Proceedings of the 24th Workshop on Combinatorial Mathematics and Computation Theory*, pp. 405–410, 2007.
- [13] T. Nakanishi, M. Shiota, and Y. Sugiyama, "An efficient online electronic cash with unlinkable exact payments," *Information Security*, vol. 3225, pp. 367–378, 2004.
- [14] Y. Baseri, B. Takhtaei, and J. Mohajeri, "Secure untraceable offline electronic cash system," *Scientia Iranica*, vol. 20, pp. 637–646, 2012.
- [15] J. Camenisch, S. Hohenberger, and A. Lysyanskaya, "Compact e-cash," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT '05)*, pp. 302–321, May 2005.

- [16] J. Camenisch, S. Hohenberger, and A. Lysyanskaya, "Balancing accountability and privacy using E-cash," in *Security and Cryptography for Networks*, vol. 4116 of *Lecture Notes in Computer Science*, pp. 141–155, 2006.
- [17] J. Camenisch, A. Lysyanskaya, and M. Meyerovich, "Endorsed e-cash," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 101–115, May 2007.
- [18] S. Canard, A. Gouget, and J. Traoré, "Improvement of efficiency in (unconditional) anonymous transferable E-cash," in *Financial Cryptography and Data Security*, vol. 5143 of *Lecture Notes in Computer Science*, pp. 202–214, 2008.
- [19] D. Chaum, A. Fiat, and M. Naor, "Untraceable electronic cash," in *Advances in Cryptology-CRYPTO '88*, vol. 403 of *Lecture Notes in Computer Science*, pp. 319–327, Springer, Berlin, Germany, 1990.
- [20] G. Davida, Y. Frankel, Y. Tsiounis, and M. Yung, "Anonymity control in E-cash systems," in *Proceedings of the First International Conference on Financial Cryptography*, pp. 1–16, 1997.
- [21] Z. Eslami and M. Talebi, "A new untraceable off-line electronic cash system," *Electronic Commerce Research and Applications*, vol. 10, no. 1, pp. 59–66, 2011.
- [22] C. I. Fan, V. S. M. Huang, and Y. C. Yu, "User efficient recoverable off-line e-cash scheme with fast anonymity revoking," *Mathematical and Computer Modelling*, vol. 58, pp. 227–237, 2013.
- [23] X. Hou and C. H. Tan, "Fair traceable off-line electronic cash in wallets with observers," in *Proceedings of the 6th International Conference on Advanced Communication Technology*, pp. 595–599, February 2004.
- [24] X. Hou and C. H. Tan, "A new electronic cash model," in *Proceedings of the International Conference on Information Technology: Coding and Computing*, pp. 374–379, April 2005.
- [25] W. S. Juang, "A practical anonymous off-line multi-authority payment scheme," *Electronic Commerce Research and Applications*, vol. 4, no. 3, pp. 240–249, 2005.
- [26] J. K. Liu, V. K. Wei, and S. H. Wong, "Recoverable and untraceable e-cash," in *International Conference on Trends in Communications (EUROCON '01)*, vol. 1, pp. 132–135, 2001.
- [27] C. Wang, H. Sun, H. Zhang, and Z. Jin, "An improved off-line electronic cash scheme," in *Proceedings of the 5th International Conference on Computational and Information Sciences (ICCCIS '13)*, pp. 438–441, 2013.
- [28] W. S. Juang, "D-cash: a flexible pre-paid e-cash scheme for date-attachment," *Electronic Commerce Research and Applications*, vol. 6, no. 1, pp. 74–80, 2007.
- [29] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology-CRYPTO '82*, *Lecture Notes in Computer Science*, pp. 199–203, Springer, Berlin, Germany, 1983.
- [30] H. Krawczyk and T. Rabin, "Chameleon signatures," in *Proceedings of the Network and Distributed System Security Symposium (NDSS '00)*, pp. 143–154, 2000.
- [31] S. Pearson, *Trusted Computing Platforms: TCPA Technology in Context*, Prentice Hall, New York, NY, USA, 2002.
- [32] S. Pearson, "Trusted computing platforms: the next security solution," Tech. Rep. HPL-2002-221, Hewlett-Packard Laboratories, 2002.
- [33] C. I. Fan and V. S. M. Huang, "Provably secure integrated on/off-line electronic cash for flexible and efficient payment," *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 40, no. 5, pp. 567–579, 2010.
- [34] S. Bajikar, Trusted platform module (TPM) based security on notebook pcs—white paper, Mobile Platform Group, Intel Corporation, 2002.
- [35] M. Abe and T. Okamoto, "Provably secure partially blind signatures," in *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '00)*, pp. 271–286, Springer, 2000.
- [36] A. Juels, M. Luby, and R. Ostrovsky, "Security of blind digital signatures," in *Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '97)*, pp. 150–164, Springer, 1997.
- [37] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko, "The one-more-RSA-inversion problems and the security of chaum's blind signature scheme," *Journal of Cryptology*, vol. 16, no. 3, pp. 185–215, 2003.
- [38] S. Brands, "Untraceable off-line cash in wallets with observers (extended abstract)," *CRYPTO*, pp. 302–318, 1993.
- [39] Y. Hanatani, Y. Komano, K. Ohta, and N. Kunihiro, "Provably secure electronic cash based on blind multisignature schemes," *Financial Cryptography*, vol. 4107, pp. 236–250, 2006.
- [40] C. Popescu, "An off-line electronic cash system with revokable anonymity," in *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference*, pp. 763–767, May 2004.
- [41] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, New York, NY, USA, 1997.

Research Article

A Secure RFID Authentication Protocol Adopting Error Correction Code

Chien-Ming Chen,^{1,2} Shuai-Min Chen,³ Xinying Zheng,¹ Pei-Yu Chen,³ and Hung-Min Sun³

¹ School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China

² Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen 518055, China

³ Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan

Correspondence should be addressed to Hung-Min Sun; hmsun@cs.nthu.edu.tw

Received 7 February 2014; Accepted 10 March 2014; Published 18 May 2014

Academic Editors: M. Ivanovic and F. Yu

Copyright © 2014 Chien-Ming Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

RFID technology has become popular in many applications; however, most of the RFID products lack security related functionality due to the hardware limitation of the low-cost RFID tags. In this paper, we propose a lightweight mutual authentication protocol adopting error correction code for RFID. Besides, we also propose an advanced version of our protocol to provide key updating. Based on the secrecy of shared keys, the reader and the tag can establish a mutual authenticity relationship. Further analysis of the protocol showed that it also satisfies integrity, forward secrecy, anonymity, and untraceability. Compared with other lightweight protocols, the proposed protocol provides stronger resistance to tracing attacks, compromising attacks and replay attacks. We also compare our protocol with previous works in terms of performance.

1. Introduction

RFID (radio frequency identification) is a technique used for identifying objects via radio frequency. It has become very popular in many applications such as access control systems, supply chain management systems, transportation, ticketing systems, and animal identification. The global transaction of RFID system was US\$2.65 billion in 2005 [1] and increased to US\$5.56 billion in 2009 [2]. At present, RFID technology has become one of the fastest growing markets in radio communication industries.

The RFID systems are composed of three components: a set of tags, RFID readers, and one or more backend servers. A backend server is responsible for storing the related information of tags, calculating the computational processes when authenticates a tag; in addition, a backend server is usually with a more powerful computation ability than RFID reader and tags. An RFID reader (called a reader in this paper) can access the backend server via secure network channel and then acquire the information related to the tags. Generally, backend servers and readers are treated as a whole entity since they are usually connected with each other through a

wired line. RFID tags are small electronic devices composed of antennas, microprocessors, and memory storages. A tag can communicate with a reader by using radio frequency signals transmitting from the reader. Normally, RFID tags can be classified into three types: active tag, semiactive tag, and passive tag. Active tags contain batteries that can actively communicate with the readers. Semiactive tags also have batteries, but they will remain silent until they receive query from a reader. Passive tags contain no battery, and their energies come from the reader's radio signals through antennas. Regarding the cost of the tag, the active and semiactive tags are expensive and each costs about US\$20, while the passive tags are usually considered as *low-cost RFID tags* which cost about US\$0.05 each. Since RFID tags usually play the roles as tickets or ID cards, most of the RFID-tagged products are small and portable, and people carry them in their daily life. For example, the e-passports combine traditional paper passports and embedded RFID chips which contain personal biometric information. They are carried by travelers from over 60 countries in the world.

While RFID technology offers convenience, security and privacy issues are still the number one concern of most RFID

applications today. Since an RFID tag can be continuously scanned within a 10 meter radius, the tag carrier's location can be easily traced without awareness; thus privacy becomes an important issue in RFID applications. Moreover, RFID tags may contain sensitive information about the carrier in which the information should not be revealed to anyone, especially to an attacker. In other words, tags should first authenticate the reader's validation before sending private data. Meanwhile, readers should also be able to authenticate tags to prevent counterfeit tags.

To address these problems, researchers have proposed many RFID protocols to achieve mutual authentication, untraceability, and other security requirements. However, with limited computational ability and insufficient memory storage on its embedded chip, low-cost RFID protocol design still remains a challenge. Previous studies showed that the number of logic gates available for security functionality on a low-cost RFID tag is 400 to 4000 [3], which is not enough to implement most public key or symmetric key cryptosystems. Therefore, an RFID protocol should be as computationally lightweight as possible.

In this paper, we propose a lightweight mutual authentication protocol based on error correction codes to provide a secure RFID mechanism. More specifically, our protocol provides mutual authenticity and untraceability to protect the security and privacy of tag carriers. We also present an evaluation on the security and performance level of our proposed protocol. Compared to other previous works, our protocol not only meets the fundamental security requirements but is also lightweight enough to be implemented on low-cost RFID tags.

The rest of this paper is organized as follows. Section 2 reviews the related works of RFID protocols. Section 3 describes a brief introduction of the error correction codes used in this paper. Our proposed RFID mutual authentication protocol is presented in Section 4. In Section 5, we analyze the security constraints of our protocol, followed by an evaluation of the performance of our protocol in Section 6. Finally, a conclusion is given.

2. Related Work

With the rapid growth of network technology, security issues have been a matter of concern in various network environments [4–12] such as wireless sensor networks, social networks, and Internet of Things. In the RFID environment, security and privacy issues also receive increasing attention recently.

There are many RFID protocols using one-way hash functions (e.g., [13, 14]) to perform their authentication process by hashing random challenges, tag identity, and/or secret key into one message. However, hardware implementations of hash functions such as SHA-1 and MD5 are generally considered too expensive to be implemented on low-cost RFID tags. However, literatures [3, 15] describe some of these implementation issues in which some of them proposed their lightweight hash functions that can be implemented on low-cost RFID tags. These lightweight hash functions include

Tav-128 proposed by Peris-Lopez et al. [16], low-cost SHA-1 proposed by O'Neill [17], and H-PRESENT-128 proposed by Bogdanov et al. [18].

The RFID authentication protocol can be classified into 4 classes. The first class refers to those protocols that apply conventional cryptographic functions, such as symmetric encryption or public key algorithm. The second class refers to those protocols that apply random number generator and one-way hash function. The third class refers to those protocols that apply random number generator and cyclic redundancy code (CRC) checksum. The last one refers to those protocols that apply simple bitwise operations (such as XOR, AND, OR, etc.). Generally, the third class is treated as lightweight level. Although our protocol has to adopt one hash function, we can simply apply the lightweight hash functions mentioned in the previous paragraph to achieve the goal of lightweight computation. Hence, by applying those lightweight hash functions, we propose our lightweight RFID protocol.

Lightweight authentication protocols aim to achieve mutual authentication through simple operations like bitwise XOR and binary addition. In 2005, Juels and Weis proposed a multiround lightweight authentication protocol called HB^+ [19], which is an improvement of *HumanAut*, a human-to-computer authentication protocol designed by Hopper and Blum [20]. Nevertheless, Gilbert et al. proved that the HB^+ protocol is vulnerable to a man-in-the-middle attack [21]. There are currently many improvements of the HB^+ protocol, for example, the HB^{++} protocol proposed by Bringer et al. in 2006 [22], the $HB-MP$ protocol proposed by Munilla and Peinado in 2007 [23], and the $HB\#$ protocol proposed by Gilbert et al. in 2008 [24].

The EPCglobal Class 1 Generation 2 UHF Air Interface Protocol Standard (generally known as Gen2 standard) [25] is a standard that defines the physical and logical requirements of RFID systems. In Gen2 standard, an RFID tag maintains the computational abilities to perform simple bitwise operations, 16-bit cyclic redundancy checks (CRC) and 16-bit pseudorandom number generator (PRNG) function. In 2009, Sun and Ting presented the $Gen2^+$ protocol [26] for Gen2 standard. In this protocol, each tag stores a string called key pool, which is shared with a backend server. $Gen2^+$ protocol is appropriate for Gen2 standard; however, Burmester et al. demonstrated an attack to break this protocol in 2009 [27].

3. Preliminary

In information theory and coding theory of computer science, error correction code (ECC) is a technique that enables the communication parties to correct the transmission errors which are incurred by the channel noise. This technique has been studied over 50 years, and substantial coding algorithms are proposed. In the following, we provide a brief introduction to one of the subclasses of ECC, called a linear block codes; in addition, if a linear block code fulfills some properties, it will form a special case of linear block codes, called perfect code. We will have a short description of perfect code in the end of this section as well.

3.1. Linear Block Codes. During the transmission, the information source, or sender, will encode a k -bit message blocks into n -bit *codewords* by using channel encoding algorithm, where $n > k$. There are total 2^k distinct messages and corresponding 2^k distinct codewords. These 2^k fixed length codewords are called a set of block codes and is denoted by $C(n, k)$. A $C(n, k)$ block code is called linear block code if it satisfies Definition 1.

Definition 1. A block code of 2^k codewords of each n -bit in length is called a linear block code if and only if these 2^k codewords form a k -dimension vector subspace over the Galois Field $GF(2)$.

Because a linear block code $C(n, k)$ is a k -dimension vector subspace, it is possible to find k linearly independent codewords in $C(n, k)$ that every codeword in $C(n, k)$ is a linear combination of these k codewords. We write these codewords into k row vectors g_0, g_1, \dots, g_{k-1} and form a $k \times n$ matrix G as follows:

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}, \quad (1)$$

where $g_i = (g_{i,0}, g_{i,1}, \dots, g_{i,n-1})$ for $0 \leq i \leq k - 1$. For a message $m = (m_0, m_1, \dots, m_{k-1})$, the corresponding codeword v can be computed as follows:

$$v = m \cdot G = (m_0, m_1, \dots, m_{k-1}) \cdot \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix}. \quad (2)$$

To decode a codeword, we first construct a $(n - k) \times n$ matrix H , which is composed of $n - k$ linearly independent rows such that any linear combination of rows in G is orthogonal to the rows of H . This implies that any codeword v in $C(n, k)$ generated by G must satisfy the following property.

Definition 2. A vector v is a codeword in $C(n, k)$ generated by G if and only if $v \cdot H^T = 0$.

Let $r = v + e$ be the received message, where v is the codeword and $e = (e_0, e_1, \dots, e_{n-1})$ is the error vector incurred by the channel noise. For a received message r , the receiver first computes a $(n - k)$ -bit vector $s = r \cdot H^T = (s_0, s_1, \dots, s_{n-k-1})$ called syndrome, which can be calculated as $s = r \cdot H^T = (v + e) \cdot H^T = v \cdot H^T + e \cdot H^T = e \cdot H^T$. If there is no error, the syndrome s will be zero and the receiver recognizes that r is the correct codeword. Nonetheless, if s is nonzero, the receiver has to determine the error vector e from s . The methods to find the error vector are different according to each coding algorithm, but we can always put every possible error pattern into the computation, get the corresponding syndromes, and construct a lookup table for the receiver in advance. Once the receiver obtained an error vector, it can recover the original codeword by computing $v = r + e$.

Hamming weight of a binary vector is defined as the number of 1 in the vector. We further define Hamming weight function $Hw(\cdot)$ to be a function that returns the Hamming weight from an input vector. Hamming distance is the number of positions that two vectors differ from each other, denoted as $Dis(\cdot)$. For instance, let $v = 1011$ and $u = 0110$ be two binary vectors; then $Dis(v, u) = 3$ since they differ in the first, second, and fourth positions.

The error correcting ability of a linear block code depends on the minimum Hamming distance (denoted as d) of every two codewords. We denote $C(n, k, d)$ as an error correction code where its codeword length, message length, and minimum Hamming distance are n, k , and d , respectively. A $C(n, k, d)$ code is capable of correcting all the error vectors which have the Hamming weight less than or equal to $t = \lfloor (d - 1)/2 \rfloor$.

3.2. Perfect Code. For a $C(n, k, d)$ code, there are 2^k codewords each with a n -bit length, and each codeword might have errors that occurred in t positions at most. Therefore, there will have total $2^k \times \sum_{i=0}^t \binom{n}{i}$ messages that can be corrected to be a valid codeword in $C(n, k, d)$. Typically, this number is no greater than the number of totally 2^n possible messages. If $C(n, k, d)$ satisfies $2^k \times \sum_{i=0}^t \binom{n}{i} = 2^n$, it is called a *perfect code*. That is, every possible message can be corrected to be a valid codeword.

4. The Proposed Protocols

In this section, we propose a lightweight RFID authentication protocol. Our main idea is to provide a mutual authentication between reader and tag. Our protocol is designed for low-cost RFID tags; therefore, the requirement for implementing our protocol will not overload the capabilities of the tags. Besides, we also propose an advanced version of our protocol to provide key updating.

Our protocol is suitable for large scale RFID systems, such as ticketing systems, transportation systems, and supply chain systems. These applications are generally composed of millions of RFID tags and readers. More importantly, the proposed protocol is appropriated for the reader to find out a specific tag from a large group of tags. For example, an airport employee desires to find a specific RFID tagged luggage from a loaded cargo truck. The proposed scheme checks whether the specific tag is in this area. In these large scale systems, readers are normally held by authorized persons or are used under supervision. They can easily connect to servers and synchronize their data. The tags in these systems are generally carried by humans or attached to goods and baggage. They are frequently scanned by the valid readers, and, in some situations, the tags can be brought back to a secure check (e.g., the RFID tagged tickets can be recycled). Before introducing the proposed protocol, the notations used are presented in the Notation section at the end of the paper.

4.1. Initialization. Initially, the administrator generates a pseudorandom number generator $g(\cdot)$, a one-way hash function $h(\cdot)$, and a $C(n, k, d)$ error correction code, with the $k \times n$

(1) R	:	Compute $C'_R = C_R + e_R$, where $\text{Hw}(e_R) \leq t$ Generate a random challenge N_R
(2) $R \rightarrow T_i$:	C'_R, N_R
(3) T_i	:	Compute $s = C'_R \cdot H^T$ IF s satisfies any pattern in s_i Generate a random challenge N_i Compute $C'_i = C_i + e_i$, where $\text{Hw}(e_i) \leq t$ $V_i = g(s \oplus N_R \oplus h(N_i \oplus k_i))$ ELSE Set $C'_i =$ random value Set $V_i =$ random value
(4) $R \leftarrow T_i$:	$k_i \oplus C'_i, V_i, N_i$
(5) R	:	Decode C'_i IF C'_i can be decoded Verify V_i to authenticate T_i IF V_i is correct $V_R = g(s \oplus N_i \oplus h(N_R \oplus k_i))$ ELSE Set $V_R =$ random value ELSE Set $V_R =$ random value Ignore V_i
(6) $R \rightarrow T_i$:	V_R
(7) T_i	:	Verify V_R to authenticate R

ALGORITHM 1: The proposed protocol.

generator matrix G and the $(n-k) \times n$ parity check matrix H . Each tag, denoted as T_i , $i = 0, 1, \dots$, has its unique identifier. We also denote their identifiers as T_i just for simplicity. For each tag T_i , the backend server S randomly generates secret keys k_i . Let s_i be a k -bit long binary vector which is a possible syndrome pattern induced by H . Each tag T_i is assigned with a syndrome pattern s_i . Then, S stores the tags' identifiers, T_i and corresponding k_i and s_i in its database. Finally, S writes $g(\cdot)$, $h(\cdot)$, T_i , k_i , s_i , G , and H into the storage memory of tag T_i in a secure environment (e.g., at RFID tags manufacturer). For every authorized reader, S also writes $g(\cdot)$, $h(\cdot)$, T_i , k_i , s_i , G , and H into their storage memory.

4.2. Authentication Protocol: Basic Version. The main objective of this protocol (Algorithm 1) is to establish a mutual authentication relationship between a reader R and a specific target tag T_x in a group of tags. Since the reader may receive a substantial amount of tags' responses for a single query, our protocol adds a filtering mechanism based on error correction codes to prevent the reader from having to examine every responding message.

At the beginning, R selects its target tag, T_x , and retrieves the corresponding k_x and s_x from database. In step 1, R randomly generates a codeword in $C(n, k, d)$, denoted as C_R . Then R generates an error vector e_R with Hamming weight less than or equal to $t = \lfloor (d-1)/2 \rfloor$, which is the maximum error correcting ability of $C(n, k, d)$. Finally, R computes the masked codeword C'_R by adding C_R with e_R . The error vector generated in this step must be selected carefully so that the

syndrome derived from C'_R will equal the preassigned pattern s_x .

In step 2, R broadcasts a query to tags, along with a random challenge N_R and the masked codeword C'_R . In Step 3, the tags attempt to decode C'_R with the parity check matrix H and compute a syndrome $s = C'_R \cdot H^T$. If a tag T_i finds that s is equal to the pattern stored in its storage memory, it randomly generates a codeword C_i in $C(n, k, d)$ and a challenge N_i . Then T_i computes a verifier message $V_i = g(s \oplus N_R \oplus h(N_i \oplus k_i))$ and the masked codeword $C'_i = C_i + e_i$, where e_i is a random error vector with $\text{Hw}(e_i) \leq t$. Since s is shorter than N_R , s should be padded before XORing with N_R . For the other tags that cannot find s in its preassigned pattern, the verifier message V_i and masked codeword C'_i are set to a random value.

Finally, no matter what their preassigned syndromes are, the tags respond $k_i \oplus C'_i, V_i$, and N_i to the reader in step 4. Note that the masked codeword C'_i is further masked with the key k_i to prevent possible tracing attack.

In Step 5, R authenticates T_i by examining the received messages. First, R uses k_x to unmask (XOR with k_x) the received messages and tries to decode every masked codeword C'_i . If R finds a codeword that cannot be decoded with the decoding algorithm, R simply ignores it and proceeds to the next masked codeword. Since the nontarget tags will always generate uncorrectable masked codewords, this method will filter out all the unnecessary messages sent by the nontarget tags, which reduces the computational loads of R . If one of these masked codewords C'_i sent by T_i can be decoded,

R uses the stored secret keys k_x , N_R , N_T , and s to verify if the corresponding V_i is sent from T_i . If V_i is correct, R computes another verifier message $V_R = g(s \oplus N_i \oplus h(N_R \oplus k_i))$. Since s is shorter than N_i , s should be padded before XORing with N_i . At this step, R has authenticated T_i to be the target tag T_x . If either C_i cannot be decoded or V_i is incorrect, R will not recognize T_i as its target, so R assigns V_R a random value. Whether T_i is the target tag or not, R always sends V_R to T_i (step 6).

In step 7, T_i verifies the received V_R to authenticate R . Only the target tag T_x that has the key k_x can accept V_R as the valid message and authenticate R by using k_x , N_i , N_R , and s . At this step, both R and T_i have authenticated each other.

4.3. Error Vector Selecting. As we stated before, the error vector generated by the reader must be selected carefully so that T_x can derive a syndrome that equals the preassigned syndrome pattern s_x . It is straightforward since the syndromes are originally used by decoding algorithms to find corresponding error vectors. That is, R can simply use the decoding algorithm to find the corresponding error vector of a specific syndrome. This error vector is exactly the error vector that should be used to mask the codeword generated by R in the first step.

4.4. Session Key. Typically, the reader and the tag would exchange data after completing the authentication process. These data are sometimes considered private; for example, the tag used in a hospital would contain the records of its carrier. The threat of eavesdropping attacks makes the tag carriers feel insecure about transmitting sensitive data. To address this problem, we construct a mechanism to establish a session key and use it to encrypt the sensitive data. We suggest that the reader and the tag use the session key $sk = g(k_i \oplus N_R \oplus N_i)$ to encrypt the messages. Without the secret key k_i , the adversary cannot decrypt the session sk to break the encrypted messages.

4.5. Secret Key Update. The secret key should not be used permanently. In fact, if the key is compromised, the messages encrypted with this key are also compromised. Hence, both the probability of messages compromised and the probability of financial loss will increase with the length of time in which a key is in use. We think that the secret keys stored in the readers and the tags should update regularly. Previous works use two approaches to perform this updating procedure. One possible approach is to have tags carriers bring their tags back to an authorized institution so that the new keys can be written into the tags in a secure environment. Another approach is to have the tags use the one-way hash functions stored in them to calculate new keys by hashing the older one.

The first approach could be combined with our authentication protocol in some RFID systems like ticketing systems and supply chain systems, since the tags are generally returned to the backend server. The second approach is also adequate for our protocol. Both the tag and the reader can hash their current secret key k_i into the new one after a successful authentication process. More precisely, the tag will

update its secret key after verifying V_R at step 7, and the reader will update its key before sending V_R to the tag (step 6). We suggest the entities update the key by computing $h(k_i \parallel N_R \parallel s)$, where \parallel denotes the string concatenation operation. The new secret key k_i is then assigned to this hashing value. Note that the session key construction process should be performed prior to updating the secret key.

If the tag does not receive the verifier message V_R , the keys between the reader and the tag might be desynchronized. This means that next time this tag's verifier message will be rejected by the reader. To address this problem, the reader should store the previous key before updating. Once the reader discovers that C_i can be decoded but V_i is incorrect, it can attempt to verify the message by using the older key. This mechanism can help the system resist desynchronization attacks.

4.6. Advanced Protocol: With Secret Key Update. Now we present a modification of our protocol with the secret key updating mechanism in it. The steps of the modified protocol are depicted in Algorithm 2. The terms k_i^{cur} and k_i^{old} represent the current secret key and the previous secret key for T_i . Note that the value k_i stored in the tag may be either k_i^{cur} or k_i^{old} . After a successful authentication process, the reader constructs the session key by using either k_i^{cur} or k_i^{old} , depending on which key is used to authenticate the tag. And the tag constructs the session key by using k_i . Then, the reader updates its secret keys by setting $k_i^{\text{cur}} = k_i^{\text{new}}$ and $k_i^{\text{old}} = k_i^{\text{cur}}$, while the tag updates the secret key by setting $k_i = h(k_i \parallel N_R \parallel s)$.

Our protocol provides a convenient method for the tag and the reader to authenticate each other before exchanging data. Since the reader will receive many messages sent from other tags at the same time, our protocol uses the properties of error correction code to filter out the unnecessary messages. Therefore, the computational load of the reader is reduced. After mutual authentication, the relation between the reader and the tag is established. They will both update their secret keys to the new ones in order to defend against possible attacks. Furthermore, the two entities can also construct a session key to protect the message transmitted later.

5. Security Analysis

In this section, we show that our protocols fulfill the security requirements for RFID systems.

5.1. Mutual Authenticity. A reader can easily authenticate the tag's identity since only the valid tag has the secret key needed to construct the correct verifier message. The random challenge N_R sent by the reader prevents the attackers from pretending to be the target tag and thus it ensures reader-to-tag authenticity. Since the reader must authenticate itself to the server before retrieving any keying information from the server, the tag can trust the reader who has the correct secret key. In other words, tag-to-reader authenticity is achieved indirectly via server-to-reader authenticity.

(1) R	:	Compute $C'_R = C_R + e_R$, where $\text{Hw}(e_R) \leq t$ Generate a random challenge N_R
(2) $R \rightarrow T_i$:	C'_R, N_R
(3) T_i	:	Compute $s = C'_R \cdot H^T$ IF s satisfies any pattern in s_i Generate a random challenge N_i $C'_i = C_i + e_i$, where $\text{Hw}(e_i) \leq t$ $V_i = g(s \oplus N_R \oplus h(N_i \oplus k_i))$ ELSE Set $C'_i =$ random value Set $V_i =$ random value
(4) $R \leftarrow T_i$:	$k_i \oplus C'_i, V_i, N_i$
(5) R	:	Compute $C_x = k_i^{\text{cur}} \oplus k_i \oplus C'_i$ Decode C_x IF C_x can be decoded Verify V_i by using k_i^{cur} IF V_i is correct $V_R = g(s \oplus N_i \oplus h(N_R \oplus k_i^{\text{cur}}))$ $k_i^{\text{new}} = h(k_i^{\text{cur}} \parallel N_R \parallel s)$ ELSE Verify V_i by using k_i^{old} IF V_i is correct $V_R = g(s \oplus N_i \oplus h(N_R \oplus k_i^{\text{old}}))$ $k_i^{\text{new}} = h(k_i^{\text{old}} \parallel N_R \parallel s)$ ELSE Set $V_R =$ random value and ignore V_i
(6) $R \rightarrow T_i$:	V_R
(7) T_i	:	Verify V_R to authenticate R

ALGORITHM 2: The proposed protocol with secret key updating.

5.2. *Integrity.* The integrity of the exchanged messages is guaranteed since the messages are encrypted by the session keys. The modification of these messages will produce meaningless plaintext, and both reader and tag can detect such modifications. During the authentication process, the adversary can also eavesdrop and modify the exchanged messages. Nevertheless, any modification on $k_i \oplus C'_i, V_i$, or V_R will lead to an incorrect verifying result on either the reader or the tag. When an adversary attempts to modify the random challenge N_i , the reader can still find the inconsistencies of N_i and V_i and thus reject the message. However, the modification of C'_R and N_R cannot be discovered by the tags because these messages are independent. This modification causes the tags to produce incorrect responses. But since the modification on C'_R will change its underlying s , all the verifier messages V_i are invalid to the reader. These messages cannot be used to perform any further attacks on the RFID system. Although we cannot guarantee the integrity of C'_R and N_R , the result of the modification on these messages is nothing but a denial-of-service attack.

5.3. *Forward Secrecy.* Our protocols maintain forward secrecy. Since the keys were updated by using one-way hash function in every session, the attacker cannot acquire the previous secret keys used in the prior sessions. Therefore,

the previous session keys and the exchanged messages are secure.

5.4. *Anonymity and Untraceability.* Our protocols do not leak the tag's identifier or any sensitive information. Therefore, our protocols fulfill the requirement of anonymity. During the authentication protocol, T_i will send messages $k_i \oplus C'_i, V_i$, and N_i to R . The adversary is able to eavesdrop all the messages sent from its target tag. With the help of these collected messages, if the adversary is able to distinguish the target tag's messages from the other tags' messages, it is able to trace this tag. Obviously, the random challenge N_i is indistinguishable from any other random number, so the adversary cannot use it to trace the tag. The verifier message V_i is constructed by a PRNG with N_i as its seed; thus it is also a random number.

Every tag stores the same generator matrix; therefore, all of them share the same probability of producing the same codeword. However, different tags will add different error vectors. As a result, the masked codewords produced by some tags can be decoded correctly while the others cannot. Once the parity check matrix is known by the adversary, this property may be used by the adversary to trace the tag. To defend against this, the tags further mask their messages with the secret keys. The adversary cannot apply decoding algorithm to the messages without first unmasking them.

Hence, we can guard against tracing attacks as long as the target tag's key is secure.

5.5. Confidentiality. Now we analyze the probability that an attacker will successfully guess one secret key of a tag with different advantages provided. First, if the adversary knows no additional information, the success probability is surely $1/2^n$. If the adversary acquires generator matrix G by compromising a tag or a reader, it will have some advantages in constructing the codewords. Now the adversary attempts to guess the C'_i to derive k_i from the message $k_i \oplus C'_i$ sent in step 4 of the proposed protocol. The number of all valid codewords C_i is 2^k . With the error vector e added in which $\text{Hw}(e) \leq t$, the number of all possible $C'_i = C_i \oplus e$ is $\sum_{i=0}^t \binom{n}{i} \times 2^k$. Therefore, the success probability of guessing the correct C'_i and k_i is $1/(\sum_{i=0}^t \binom{n}{i} \times 2^k)$. Notice that the adversary is able to verify whether the guess is correct or not by rapidly substituting the keys into the verifier messages V_R , sending it to T_i , and validating the response V_i . ISO standard 14443 specifies the data exchange rate between the reader and the tag, which is 106 kbit to 848 kbit [28]. Based on this data, we can calculate the relationship between the different codes, the amount of messages the tag transmitted, and the response time, where the response time is the time required for a tag to respond to reader's query. The result is depicted in Table 1.

Assume the adversary tries to launch the guessing attack by rapidly querying the tag before the tag's stored key can be updated by the valid reader. Generally, in real-world applications, the adversary is unable to rapidly query a specific tag for a long time because of the mobility of the tag's carrier. Therefore, attacks that require more than one hour may be regarded as useless. Nonetheless, the adversary may steal a tag from the system to avoid side effects caused by carriers. Nevertheless, in some existing RFID systems, tags will be recycled regularly. For example, in the public transportation systems, the RFID tagged tickets will be recycled and calculated every day. The system manager can find that if a tag has been stolen and remove that tag from the system. As a result, the stolen tag will be unusable hereafter, and the attacker can no longer threaten the system with the tag. In other words, if the required time of an attack is higher than one day, the system can be considered secure. In Table 2, we estimate the success probability of key guessing attack if the attacker performs the attack by rapidly querying the tag either within one hour or within one day. Based on the above arguments and analysis, we choose $C(47, 24, 11)$, $C(63, 57, 3)$, $C(63, 39, 9)$, $C(63, 24, 15)$, and $C(127, 36, 39)$ as the candidates for implementing our protocol since they provide better security. In some systems with intensive surveillance, $C(31, 26, 3)$ can also be taken into consideration.

5.6. Comparison. In the following, we show the comparisons between our protocol and other related protocols in terms of the security requirements. We take Chien's SASI protocol [29] and Chien-Lai's ECC-based protocol [30], Juels-Weis' HB^+ protocol [19], and Sun-Ting's Gen2^+ protocol [26] into comparison. These lightweight protocols are similar to our

TABLE 1: Estimated response time in different error correction codes.

Error correction code	Messages amount (bits)	Response time (μs)
$C(7, 4, 3)$	21	24.8~198.1
$C(15, 5, 7)$	45	53.1~424.5
$C(24, 12, 8)$	72	84.9~679.2
$C(31, 26, 3)$	93	110.0~877.4
$C(31, 6, 15)$	93	110.0~877.4
$C(47, 24, 11)$	141	166.3~1330.2
$C(63, 57, 3)$	189	222.9~1783.0
$C(63, 39, 9)$	189	222.9~1783.0
$C(63, 24, 15)$	189	222.9~1783.0
$C(127, 36, 29)$	381	449.3~3594.3
$C(255, 187, 19)$	765	902.1~7217.0

protocol in basic assumptions. The comparison results of security requirements are shown in Table 3.

SASI protocol was proposed in 2007. This ultralightweight authentication protocol requires only PRNG and simple bitwise operations which are supported by EPC Gen2 tags. However, studies [31, 32] showed that SASI is vulnerable to desynchronizing and tracing attacks. Chien-Lai's ECC-based lightweight authentication protocol was proposed in 2009. However, this protocol cannot defend against the tracing attacks [33]. Juels-Weis' HB^+ protocol is a multiround lightweight mutual authentication protocol. It requires the tags and the readers to share the same secret to perform its authentication protocol. Studies have proved that HB^+ protocol is vulnerable to a man-in-the-middle attack [21]. In this attack, the attacker can retrieve the entire secret and impersonate the valid tag. Therefore, HB^+ cannot satisfy authenticity. And, without a secret key update scheme, this protocol also cannot maintain forward secrecy. Sun-Ting's Gen2^+ protocol is another lightweight mutual authentication protocol suitable for Gen2 standard. In [27], the authors proved that the attacker can calculate a fake message to pass the authentication process by replaying the previous messages. As a result, Gen2^+ is unable to fulfill authenticity requirement.

5.7. Summary. We had analyzed the security of our protocol and showed that our protocol provides high security against the common security threats of the RFID systems. We also analyzed the adversary's success probability of recovering the secret key. With careful parameter selection, the attacker will need a long time to break the protocol. Therefore, in most application scenarios, our protocol provides a good solution for securing the RFID system.

6. Evaluation

In this section, we will first describe the hardware constraints on selecting parameters for our lightweight protocol. Then we will have a discussion on the computational loads of the reader and the tag. Finally, based on the analysis, we

TABLE 2: Estimated success probability for key guessing attack.

Error correction code	Success probability of different attack periods	
	Within one hour	Within one day
$C(7, 4, 3)$	1	1
$C(15, 5, 7)$	1	1
$C(24, 12, 8)$	0.56~1	1
$C(31, 26, 3)$	0.002~0.02	0.05~0.37
$C(31, 6, 15)$	0.02~0.14	0.43~1
$C(47, 24, 11)$	$9.3 \times 10^{-8} \sim 7.5 \times 10^{-7}$	$2.2 \times 10^{-6} \sim 1.8 \times 10^{-5}$
$C(63, 57, 3)$	$2.2 \times 10^{-13} \sim 1.8 \times 10^{-12}$	$5.3 \times 10^{-12} \sim 4.2 \times 10^{-11}$
$C(63, 39, 9)$	$5.8 \times 10^{-12} \sim 4.6 \times 10^{-11}$	$1.4 \times 10^{-10} \sim 1.1 \times 10^{-9}$
$C(63, 24, 15)$	$1.9 \times 10^{-10} \sim 1.5 \times 10^{-9}$	$4.6 \times 10^{-9} \sim 3.7 \times 10^{-8}$
$C(127, 36, 29)$	$8.3 \times 10^{-24} \sim 6.6 \times 10^{-23}$	$2.0 \times 10^{-22} \sim 1.6 \times 10^{-21}$

TABLE 3: Comparison of security properties.

	Our Protocol	Chien's [29]	Chien and Lai's [30]	Juels and Weis's [19]	Sun and Ting's [26]
Authenticity	✓	✓	✓	✗	✗
Integrity	✓	✓	✓	✓	✓
Forward secrecy	✓	✓	✓	✗	✓
Anonymity	✓	✓	✓	✓	✓
Untraceability	✓	✗	✗	✓	✓
Resistance to compromising	✓	✓	✓	✓	✓
Resistance to desynchronizing	✓	✗	✓	✓	✓

✓: satisfied; ✗: unsatisfied.

will compare our protocol with previous works in terms of performance.

6.1. Parameter Selection. We analyze the memory storage and computational capability on the low-cost RFID tags in this section. Based on the analysis, we will select parameters that provide enough security to our protocol and show that the protocol is lightweight enough to be implemented on the tags.

Since our protocol requires tag to store the generator matrix G and the parity check matrix H , the size of the matrices should not exceed the size of the tag's storage memory. Fortunately, most passive RFID tags have 1 Kbytes–8 Kbytes of storage; some may even have up to 64 Kbytes of storage [15]. This is sufficient for storing our matrices, which only require about 1 Kbytes–2 Kbytes. With the secret keys and other information added, the requirement is still within the tag's capability.

Next we turn our attention to the tag's computational power. As estimated in [15], the cost of an RFID tag should range from US\$0.05 to US\$0.10, and the area of a silicon chip is limited to approximately 0.25 mm^2 – 0.5 mm^2 . Under these constraints, the number of logical gates that can be mounted on the chip is limited. Researchers from Auto-ID Labs have estimated that only 400–4000 gate equivalents (GE) can be used for the security related functionality [3].

When running our protocol, the tag has to perform vector-matrix multiplication for decoding and encoding.

According to [34], this multiplication can actually be performed by broadcasting columns of the matrix and multiplying them with the corresponding row elements of the vector. Therefore, the operation is simply to rapidly read a column of the matrix from the memory, XOR it with the vector, and accumulate them into a buffer until all the columns are multiplied. The only operation required in the vector-matrix multiplication is a bitwise XOR, which is not an obstacle for the RFID tags. However, during the operation, the elements need to be loaded into the registers. This implies that our protocol requires at least $3n$ bits of registers for buffer implementation. We also need n bitwise XOR logical gates for the multiplication. The other operations, like adding error vector, can also be performed by using these buffers and XOR gates. One bit register takes 6 GE to implement, and a XOR logical gate costs 2.67 GE. Besides, in our protocol, a one-way hash function is required to compute the verifier messages. Implementation of a lightweight hash function costs about 2500 GE [16]. Based on the above analysis, we now estimate the number of required GE for each parameter set we suggested in Section 5.5. The result is listed in Table 4. Most of these do not exceed the limitation of 4000 GE.

6.2. Performance. It is difficult to implement our protocol on the current low-cost RFID tags, since most of the RFID modules are not user-programmable. They run merely the processes that set in manufacturer phase. Therefore, we cannot evaluate the time consuming on the real tags. Hence,

TABLE 4: Estimated gate equivalents for different parameters.

Error correction code	Required gate equivalents
C(31, 26, 3)	3141
C(47, 24, 11)	3471
C(63, 24, 15)	3802
C(63, 39, 9)	3802
C(63, 57, 3)	3802
C(127, 36, 39)	5125

TABLE 5: Estimated transmitting time for different parameters.

Error correction code	Required transmitting time (ms)		
	$N = 1$	$N = 10$	$N = 100$
C(31, 26, 3)	0.2~1.8	0.5~4.4	3.8~30.7
C(47, 24, 11)	0.3~2.7	0.8~6.7	5.8~46.6
C(63, 24, 15)	0.4~3.6	1.1~8.9	7.8~62.4
C(63, 39, 9)	0.4~3.6	1.1~8.9	7.8~62.4
C(63, 57, 3)	0.4~3.6	1.1~8.9	7.8~62.4
C(127, 36, 39)	0.9~7.2	2.2~18.0	15.7~125.8

we calculated the average amount of transmitted messages in our protocol to estimate the average time of communicating.

Assume that a reader is going to authenticate a tag from N tags. We denote L as the length of the secret key. In our protocol, the secret key length L is equal to the length of the message, n . For each tag, it will send $k \oplus C'_i$, V_i , and N_i to respond to the reader's single query. All of them are L bits in length. For the reader, it will broadcast C'_R and N_R to tags ($2L$ bits). After receiving one response message from a tag, the reader will try to decode it. Whatever the decoding result is, the reader always sends a L -bit message V_R to the tag. Since the reader will receive at most N responses from the tags, it will broadcast at most NL bits of V_R messages. As a result, the total amount of transmitted messages of the reader and the tags during the authentication process is $NL + 2L$ and $3NL$, respectively.

Now we can estimate the running time of our protocol. First note that all tags compute and transmit their messages in parallel; therefore, we should use the amount of total message of a single tag ($3L$ bits) for our calculation. Also, based on the fact that the data rate specified in ISO 14443 standard is 106 Kbits to 848 Kbits, we can compute the required data transmitting time of our protocol. The result is shown in Table 5. Even in the worst case scenario, the longest transmitting time is still about 0.13 seconds, which is negligible for most users.

In order to minimize its computational load, the reader will attempt to filter out the unnecessary verifier messages V_i . At step 3, when the tag discovers the syndrome s it computed is not matching with the syndrome pattern it stored, the tag will assign a random value to the masked codeword. Even though the probability is small, this random value may be recognized as a valid codeword by the reader. If a random number is recognized as the codeword, the reader has to verify an extra verifier message, thus adding its load.

TABLE 6: Probability of mistaking the random number as valid codeword.

Error correction code	Probability
C(31, 26, 3)	1
C(47, 24, 11)	0.206
C(63, 24, 15)	0.001
C(63, 39, 9)	0.038
C(63, 57, 3)	1
C(127, 36, 39)	9.1×10^{-6}

TABLE 7: Estimated number of unnecessary verifier messages.

Error correction code	Average number of extra verifier messages		
	$N = 1$	$N = 10$	$N = 100$
C(31, 26, 3)	1	10	100
C(47, 24, 11)	0.2	2.1	20.6
C(63, 24, 15)	1.1×10^{-3}	1.1×10^{-2}	0.1
C(63, 39, 9)	3.8×10^{-2}	0.4	3.8
C(63, 57, 3)	1	10	100
C(127, 36, 39)	9.1×10^{-6}	9.1×10^{-5}	9.1×10^{-4}

N : number of tags.

The probability can be computed by dividing the number of all possible C'_i by the number of all possible random values; that is, $(\sum_{i=0}^t \binom{n}{i} \times 2^k) / 2^n$. In Table 6, we show the probability that the random number is recognized as a valid codeword between different codes. Note that C(31, 26, 3) and C(63, 57, 3) are perfect codes. Therefore, their probability of mistake is 1 since every random message can be corrected to a valid codeword.

Because the number of possible syndrome patterns is limited, a pattern might be shared by many tags. In other words, tags might store the same syndrome pattern. If the reader wants to authenticate one of these tags, each of them will respond with a valid codeword and verifier message. If that is the case, the reader will have to verify unnecessary verifier messages. The number of tags that share the same syndrome pattern is $N/2^k$, if the syndrome patterns are randomly distributed to the tags. Taking the mistaking probability shown in Table 6 and the number of unnecessary responses into consideration, we estimate the average number of verifier messages V_i in which a reader has to verify in an authentication process. The result is shown in Table 7. The greater the number, the heavier the reader's computational load. Notice that the target tag might not be the one of these N tags in real-world applications; therefore, we have to remove the target tag from the experiment in order to get a fair result. Depending on the above evaluation, C(63, 24, 15), C(63, 39, 9), and C(127, 36, 39) provide good balance for the reader in both security and performance.

6.3. *Comparisons.* We compare the amount of transmitted messages between different authentication protocols as follows. Still taking Chien's SASI protocol [29] and Chien-Lai's ECC-based protocol [30], Juels-Weis's HB⁺ protocol [19], and

TABLE 8: Comparison of total messages transmitted.

Authentication protocol	Total amount of transmitted messages (bit)		
	$N = 1$	$N = 10$	$N = 100$
Our protocol	$6L$	$42L$	$402L$
Chien's SASI [29]	$5L$	$14L$	$104L$
Chien and Lai's ECC-based [30]	$4D + 2L$	$40D + 20L$	$400D + 200L$
Juels and Weis's HB ⁺ [19]	$Q \times (1 + 2L)$	$10Q \times (1 + 2L)$	$100Q \times (1 + 2L)$
Sun and Ting's Gen2 ⁺ [26]	$32Q + 96$	$10 \times (32Q + 96)$	$100 \times (32Q + 96)$

N : number of tags; L : key length; Q : number of rounds; D : length of random number.

Sun-Ting's Gen2⁺ protocol [26] into comparison, assume that the reader needs to authenticate a specific tag from a group of N tags. The amounts of messages sent between different number of tags and protocols are presented in Table 8.

In SASI protocol, the tags first send pseudonyms IDS s to the reader, and the reader replies with the messages A , B , and C to the target tag. Finally, the tag responds with message D to the reader. Each message is Kbits in length. In this protocol, the reader is able to find its target tag from the tags' responding IDS s. Therefore, the reader does not need to transmit any unnecessary message to the nontarget tags. In Chien-Lai's ECC-based protocol, the exchanged messages including a random number N_R , the message sets $\{(\bar{C}_i, \bar{V}_T), (\bar{C}_i, \bar{V}_T)\}$ and V_S . \bar{C}_i and \bar{C}_i are Kbits in length. On the other hand, N_R , \bar{V}_T , \bar{V}_T , and V_S are generated by a PRNG. We denote their length in bits as D . When the reader wants to find a tag from a group of tags, it has to authenticate every tag until it finds its target tag. In HB⁺ protocol, the reader and the tag exchange two random numbers and one bit message z in a single round. But the reader is still required to authenticate each tag to find its target tag. In Gen2⁺ protocol, the tag transmits 16-bit message set $(a, b, check)$ to the reader, and the reader responds with 16-bit ck' to the tag in a single round. After running Q rounds, the tag eventually responds with a 96-bit EPC to the reader. In this protocol, the reader has to authenticate each tag until it finds its target tag.

Compared with these protocols, the total amount of messages our protocol sent is no greater than most of the existing protocols. Although SASI protocol provides a very efficient identification mechanism based on tags' pseudonyms, the fixed pseudonyms make the tags vulnerable to tracing attack before they can be updated again.

7. Conclusion

Security and privacy issues on RFID have been studied in recent years due to the rapid growth of RFID systems. Many researchers worry about the disadvantages of RFID technology, such as keeping their location privacy and confidentiality of private information. On the other hand, manufacturers do not provide security functionality on their products because of the native limitation of RFID tags. As a result, researchers have proposed substantial lightweight authentication protocols for securing low-cost RFID tags.

Some real-world RFID application scenarios require a reader to find out and authenticate a tag from a group of tags.

In previous works, the reader has to authenticate each tag individually until the reader found the target one, thus greatly increasing the communication and computation time. To address this problem, our protocol provides an error correction codes based mechanism to minimize the computational load of reader. When receiving query, the tags respond with verifier messages along with different codewords in which some of them cannot be decoded. The reader can filter out the unnecessary verifier messages by examining these codewords, therefore improving its performance.

In this paper, we presented a single-round lightweight mutual authentication protocol. The protocol is designed with decoding and encoding operations on error correction codes, pseudorandom number generating, and a hash function. These operations are proved lightweight enough to be implemented on low-cost RFID tags or can be realized by using simple bitwise operations. Based on the secrecy of shared keys, the reader and the tag can establish a mutual authenticity relationship. Further analysis of the protocol showed that it also satisfies integrity, forward secrecy, anonymity, and untraceability. Compared with other lightweight protocols, the proposed protocol provides stronger resistance to tracing attacks, compromising attacks, and replay attacks.

Notation

S :	RFID backend server
R :	RFID reader
T_i :	A RFID tag
s :	Syndrome pattern
s_i :	A syndrome pattern of T_i
k_i :	A secret key of T_i
$g()$:	Pseudorandom number generator
$h()$:	One-way hash function
G :	Generator matrix
H :	Parity check matrix
C_R, C_i :	ECC codeword from R and T_i , respectively
e_R, e_i :	ECC error vector from R and T_i , respectively
V_R, V_i :	Verifier message from R and T_i , respectively
N_R, N_i :	Random nonce from R and T_i , respectively
$Hw()$:	Hamming weight.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The work of Chien-Ming Chen was supported in part by the Project HIT.NSRIF.2014098 supported by Natural Scientific Research Innovation Foundation in Harbin Institute of Technology, in part by Shenzhen Peacock Project, China, under Contract KQC201109020055A, and in part by Shenzhen Strategic Emerging Industries Program under Grant ZDSY20120613125016389. The work of H.-M. Sun was supported in part by the National Science Council, Taiwan, under Grant NSC 100-2628-E-007-018-MY3.

References

- [1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, John Wiley & Sons, New York, NY, USA, 2003.
- [2] R. Das and P. Harrop, "RFID forecasts, players and opportunities 2009–2019," IDTechEx Report, 2009.
- [3] D. Ranasinghe, D. Engels, and P. Cole, "Low-cost RFID systems: confronting security and privacy," in *Proceedings of the Auto-ID Labs Research Workshop*, pp. 54–77, 2004.
- [4] C. M. Chen, Y. H. Lin, Y. H. Chen, and H. M. Sun, "Sashimi: secure aggregation via successively hierarchical inspecting of message integrity on wsn," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 1, pp. 57–72.
- [5] E. K. Wang, Y. Ye, and X. Xu, "Locationbased distributed group key agreement scheme for vehicular ad hoc network," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 759601, 8 pages, 2014.
- [6] W. C. Ku, C. M. Chen, and H. L. Lee, "Cryptanalysis of a variant of peyavian-zunic's password authentication scheme," *IEICE Transactions on Communications*, vol. E86-B, no. 5, pp. 1682–1684, 2003.
- [7] C. W. Lin, T. P. Hong, C. C. Chang, and S. L. Wang, "A greedy-based approach for hiding sensitive itemsets by transaction insertion," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 4, pp. 201–227, 2013.
- [8] C. M. Chen, Y. H. Chen, Y. H. Lin, and H. M. Sun, "Eliminating rouge femtocells based on distance bounding protocol and geographic information," *Expert Systems with Applications*, vol. 41, no. 2, pp. 426–433, 2014.
- [9] B. Z. He, C. M. Chen, Y. P. Su, and H. M. Sun, "A defence scheme against identity theft attack based on multiple social networks," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2345–2352, 2014.
- [10] H. M. Sun, H. Wang, K. H. Wang, and C. M. Chen, "A native APIs protection mechanism in the kernel mode against malicious code," *IEEE Transactions on Computers*, vol. 60, no. 6, pp. 813–823, 2011.
- [11] T. Y. Wu and Y. M. Tseng, "Further analysis of pairing-based traitor tracing schemes for broadcast encryption," *Security and Communication Networks*, vol. 6, no. 1, pp. 28–32, 2013.
- [12] C. M. Chen, K. H. Wang, T. Y. Wu, J. S. Pan, and H. M. Sun, "A scalable transitive humanverifiable authentication protocol for mobile devices," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1318–1330, 2013.
- [13] M. Conti, R. Di Pietro, L. V. Mancini, and A. Spognardi, "RIPP-FS: An RFID identification, privacy preserving protocol with forward secrecy," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 229–234, March 2007.
- [14] B. Song and C. J. Mitchell, "RFID authentication protocol for low-cost tags," in *Proceedings of the 1st ACM Conference on Wireless Network Security*, pp. 140–147, April 2008.
- [15] J. Guajardo, P. Tuyls, N. Bird et al., "RFID security: cryptography and physics perspectives," *RFID Security*, pp. 103–130, 2008.
- [16] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "An efficient authentication protocol for RFID systems resistant to active attacks," in *Emerging Directions in Embedded and Ubiquitous Computing*, vol. 4809 of *Lecture Notes in Computer Science*, pp. 781–794, 2007.
- [17] M. O'Neill, "Low-cost SHA-1 hash function architecture for RFID tags," in *Proceedings of the International Conference on RFID Security*, 2008.
- [18] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, and Y. Seurin, "Hash functions and RFID tags: mind the gap," in *Cryptographic Hardware and Embedded Systems—CHES 2008*, vol. 5154 of *Lecture Notes in Computer Science*, pp. 283–299, 2008.
- [19] A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols," in *Advances in Cryptology—CRYPTO 2005*, vol. 3621 of *Lecture Notes in Computer Science*, pp. 293–308, 2006.
- [20] N. Hopper and M. Blum, "Secure human identification protocols," in *Proceedings of the 7th International Conference on Theory and Application of Cryptology and Information Security*, pp. 52–66, 2001.
- [21] H. Gilbert, M. Robshaw, and H. Sibert, "Active attack against HB+: a provably secure lightweight authentication protocol," *Electronics Letters*, vol. 41, no. 21, pp. 1169–1170, 2005.
- [22] J. Bringer, H. Chabanne, and E. Dottax, "HB++: a lightweight authentication protocol secure against some attacks," in *Proceedings of the 2nd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, pp. 28–33, June 2006.
- [23] J. Munilla and A. Peinado, "HB-MP: a further step in the HB-family of lightweight authentication protocols," *Computer Networks*, vol. 51, no. 9, pp. 2262–2267, 2007.
- [24] H. Gilbert, M. Robshaw, and Y. Seurin, "HB#: increasing the security and efficiency of HB+," in *Proceedings of the 27th International Conference on Theory and Applications of Cryptographic Techniques*, pp. 361–378, 2008.
- [25] "EPCglobal," <http://www.epcglobalinc.org>.
- [26] H. M. Sun and W. C. Ting, "A Gen2-based RFID authentication protocol for security and privacy," *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1052–1062, 2009.
- [27] M. Burmester, B. de Medeiros, J. Munilla, and A. Peinado, "Secure EPC Gen2 compliant radio frequency identification," *Ad-Hoc, Mobile and Wireless Networks*, vol. 5793, pp. 227–240, 2009.
- [28] Identification Cards—Contactless Integrated Circuit Cards—Proximity Cards, ISO, 14443 Std.
- [29] H. Y. Chien, "SAS: a new ultralightweight RFID authentication protocol providing strong authentication and strong integrity," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 4, pp. 337–340, 2007.
- [30] H. Y. Chien and C. S. Lai, "ECC-based lightweight authentication protocol with untraceability for low-cost RFID," *Journal of Parallel and Distributed Computing*, vol. 69, no. 10, pp. 848–853, 2009.

- [31] H. M. Sun, W. C. Ting, and K. H. Wang, "On the security of Chien's ultralightweight RFID authentication protocol," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 2, pp. 315–317, 2011.
- [32] R. W. Phan, "Cryptanalysis of a new ultralightweight RFID authentication protocolSASI," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, no. 4, pp. 316–320, 2009.
- [33] C. -M. Chen, S. -M. Chen, X. Zheng, L. Yan, H. Wang, and H. -M. Sun, "Pitfalls in an ECC-based lightweight authentication protocol for low-cost RFID," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 4, 2014.
- [34] S. Qasim, A. Telba, and A. AlMazroo, "FPGA design and implementation of matrix multiplier architectures for image and signal processing applications," *International Journal of Computer Science and Network Security*, vol. 10, no. 2, pp. 168–176, 2010.

Research Article

High Capacity Reversible Watermarking for Audio by Histogram Shifting and Predicted Error Expansion

Fei Wang,¹ Zhaoxin Xie,^{2,3} and Zuo Chen¹

¹ School of Information Science and Engineering, Hunan University, Lushan South Road, Changsha 410082, China

² College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China

³ Chu Kochen Honors College, Zhejiang University, Hangzhou 310058, China

Correspondence should be addressed to Zhaoxin Xie; xzxiverson@126.com

Received 6 March 2014; Accepted 22 April 2014; Published 14 May 2014

Academic Editor: Fei Yu

Copyright © 2014 Fei Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Being reversible, the watermarking information embedded in audio signals can be extracted while the original audio data can achieve lossless recovery. Currently, the few reversible audio watermarking algorithms are confronted with following problems: relatively low SNR (signal-to-noise) of embedded audio; a large amount of auxiliary embedded location information; and the absence of accurate capacity control capability. In this paper, we present a novel reversible audio watermarking scheme based on improved prediction error expansion and histogram shifting. First, we use differential evolution algorithm to optimize prediction coefficients and then apply prediction error expansion to output stego data. Second, in order to reduce location map bits length, we introduced histogram shifting scheme. Meanwhile, the prediction error modification threshold according to a given embedding capacity can be computed by our proposed scheme. Experiments show that this algorithm improves the SNR of embedded audio signals and embedding capacity, drastically reduces location map bits length, and enhances capacity control capability.

1. Introduction

Reversible watermarking is a technique that means after watermark information was embedded into the host data, original data can be restored while watermark information is extracted through a series of processes. Because the reversible watermarking schemes can achieve lossless recovery, they can be widely used in military communication, medical image, and digital forensics fields [1–3]. At present, the reversible watermarking technology research mainly concentrates on image [4–6]. Tian [7] proposed an image reversible watermarking algorithm using difference expansion technique. This algorithm first applied an integer wavelet transform on two adjacent pixels in the image and then embedded the watermark data in host pixel by expanding the difference of adjacent pixel. Finally, the algorithm restored pixel data through the inverse integer transform. In [8], a reversible watermarking algorithm was presented using prediction error expansion. First, the algorithm predicted a pixel value

according to three pixels adjacent to the current pixel and then extended the difference between prediction and current value to achieve watermark information embedding. The algorithm has higher embedding capacity capability and better quality of embedded image than Tian's scheme. Digital speech is one of the important forms of digital media, but reversible watermarking algorithm for digital audio is relatively less. Reversible audio watermarking schemes mainly face the following problems: (1) audio in WAV format is a floating number ranged from -1 to 1 . In order to transform these data, we must map the data to integer space; (2) human audio system is more sensitive than the human visual system, so even slight alteration of audio signal can easily be perceived; (3) the process of audio signal is much trickier than digital image [9]. In [10], Yan and Wang proposed a reversible audio watermarking scheme based on prediction error expansion method. First, three previous samples of the audio sample were multiplied by integer coefficients individually and then the sum of three results was used to

predict the current sample. Second, watermarking information was embedded by expanding the difference between the prediction and current amplitude. This algorithm mapped the audio data to integer interval and achieved recovery of original audio signals. In the algorithm, a location map was used to mark the samples without watermark bits. However, the method suffers from lack of embedding capacity due to large auxiliary embedded location information. In [11], a reversible audio data hiding algorithm based on linear prediction and error expansion was presented by A. Nishimura. Relative to [10], this paper improved the prediction coefficients using Burg method and obtained a higher SNR and embedding capacity. However, its capacity has the same drawback with [10] and capacity control capability was not considered.

Differential evolution (DE) algorithm is an optimization algorithm based on swarm intelligence. It can simulate the biological natural selection and evolution. This algorithm was first presented by Storn and Price in 1995 [12]. DE algorithm can solve multiparameters optimization without depending on the continuity of objective function and the conductivity of the constraints. Meanwhile, this algorithm has good robustness and universality. It is widely used in many fields [13–16].

The application of histogram to reversible watermarking technique first appeared in image field. Ni et al. [17] proposed a reversible watermarking algorithm based on histogram shifting, where watermark is embedded into the histogram bins. A novel watermarking technique has been presented by Thodi and Rodríguez in [18], where location information is dramatically compressed by using histogram shifting technique. To our best knowledge, however, reversible audio watermarking based on histogram technique has not been reported yet. So, we cannot apply the histogram technique of image field to audio data directly.

In this paper, we proposed a novel reversible audio watermarking scheme based on improved prediction error expansion and histogram shifting which combined with the characteristics of digital audio data. Different from using the fixed set of linear prediction coefficients, we propose the optimization model of prediction coefficients and then use the differential optimization algorithm to determine the optimal linear prediction coefficients, so that the overall prediction errors are as small as possible. In addition, according to the characteristics of audio data, we introduced histogram shifting technique which can dramatically reduce the embedding location information. Meanwhile, we present the method which can compute the prediction error modification threshold according to a given embedding capacity. The experiments show that this algorithm can enhance capacity control capability.

The organization of this paper is as follows. In Section 2, we propose prediction error expansion based on differential evolution. Section 3 describes how information embedding and extraction are done by using histogram shifting and prediction error expansion. The experiment results and analysis are presented in Sections 4 and 5, respectively. The conclusions are in Section 6.

2. Prediction Error Expansion Based on Differential Evolution

2.1. Differential Evolution Algorithm. Differential evolution algorithm has the advantages of high efficiency, good convergence, and robustness. It can perform well when deal with multiparameter optimization problem, especially when the solution space is stochastic. Differential evolution algorithm can be divided into the following operations.

- (1) The mutation: the mutation of DE is driven by the difference between the parent individuals; the mutation of vector u_i^g is as follows:

$$v_i^{g+1} = u_{r_1}^g + F(u_{r_2}^g - u_{r_3}^g), \quad r_1 \neq r_2 \neq r_3 \neq i, \quad (1)$$

where r_1, r_2, r_3 are uniform random integers $\in [1, N_p]$, $F \in [0, 2]$ represents the control parameters, and g means the index of generation.

- (2) Crossover: in the DE algorithm, the crossover operation guarantees the diversity of population. When applying crossover on mutant vector v_i^{g+1} and current individual u_i^g , we can get new vector $w_{i,j}^{g+1}$. The formula of crossover operation is as follows:

$$w_{i,j}^{g+1} = \begin{cases} v_{i,j}^{g+1}, & (\text{rand}() \leq C_R \text{ or } j = l_i) \\ u_{i,j}^g, & \text{otherwise,} \end{cases} \quad (2)$$

where $j \in (1, 2, \dots, D)$ and D is the dimension of the problem, $\text{rand}()$ means the random values $\in [0, 1]$, and C_R represents the probability of crossover operation. $l_i \in [1, 2, \dots, D]$ is a randomly chosen value.

- (3) Selection: the selection operation of DE algorithm is based on a greedy selection scheme. We choose the individual which has better fitness between the vector w_i^{g+1} and the individual of original population u_i^g . The formula is described as follows:

$$u_i^{g+1} = \begin{cases} w_i^{g+1}, & f(w_i^{g+1}) \leq f(u_i^g) \\ u_i^g, & \text{otherwise,} \end{cases} \quad (3)$$

where $f(\cdot)$ represents fitness function.

2.2. The Optimization of Linear Prediction Coefficients Based on DE Algorithm. Figure 1 presents a typical histogram of prediction errors for classic music. The prediction errors histogram for most natural audio would be similar to this. Consider a process of expansion embedding. The smaller the magnitude of the prediction errors, the smaller the distortion of the audio. In the sense, the prediction errors occur more frequently in the central region. Since embedding capacity and distortion performance are most important indicators for reversible audio watermarking scheme, the optimization aim is to maximize SNR of embedded audio and capacity. The formula is presented as follows:

$$F = 10 \log_{10} \left(\frac{\sum_{i=1}^L S^2(i)}{\sum_{i=1}^L [S^2(i) - S'^2(i)]} \right) \times \frac{l}{L}, \quad (4)$$

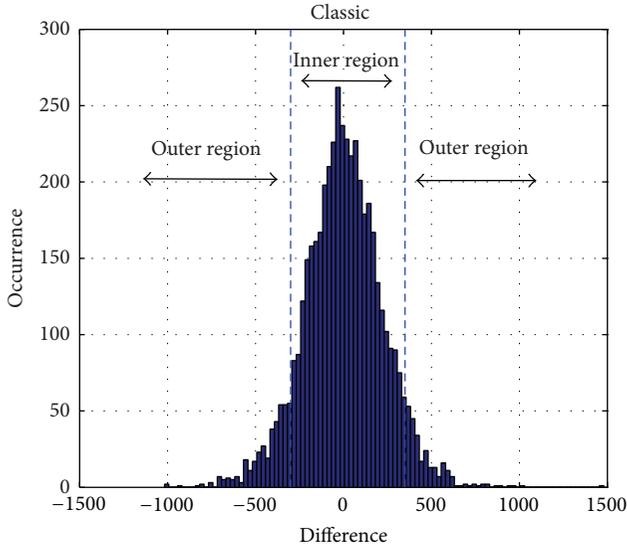


FIGURE 1: Histogram of prediction errors for classic music.

where $S^2(i)$ and $S'^2(i)$ mean original and embedded audio, respectively, L is the length of the audio data, and l means the number of embedded bits.

3. Prediction Error Expansion Based on Histogram Shifting

3.1. Histogram Shifting. In [18], histogram of prediction error for a host data is divided into two nonoverlapping regions, the outer and the inner regions, which are shown in Figure 1. Expansion of the prediction errors expands the histogram of the inner region. As is shown in Figure 2, after a process of expansion, the peak of the histogram has fallen by almost 50%. Because the histogram of prediction errors for audio does not strictly satisfy the continuity, the algorithm in [18], in which the determination of embedded location was based on symmetrical histogram, cannot be applied to audio reversible watermarking algorithm. In this paper, we assume two thresholds T_l and T_r , which can be used to control the boundaries of the inner region and outer region. We first set $T_l = 0$, $T_r = 1$. Given a capacity, we can determine the boundaries of inner region and outer region according to the Algorithm 1.

Where D means all the prediction errors, function $\text{num}(\cdot)$ returns the index of the given value. In the process of determining the thresholds, we considered the cumulative impact by all expansive bins rather than only the current expansive bin.

3.2. Embedding and Extraction. All steps of the watermark embedding procedure based on improved prediction error expansion and histogram shifting are shown in Figure 3. The details are described as follows.

Step 1. Assume a host audio data $A = \{a_1, a_2, a_3, \dots, a_N \mid a_i \in [-1, 1]\}$. For audio sample a_i , we define $a_{i-3}, a_{i-2}, a_{i-1}$ as

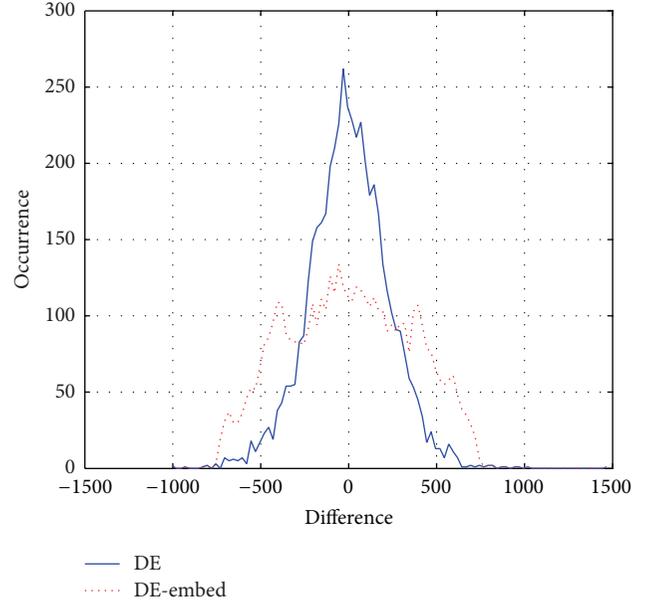


FIGURE 2: Histogram after expansion of inner region and original prediction errors.

```

sort(D)
set  $T_l = 0, T_r = 1$ 
set  $i = \text{num}(T_l), j = \text{num}(T_r)$ 
while( $P < P_{\text{set}}$ )
if  $(\text{num}_l + \text{num}(i)) / \text{cnt}_l < (\text{num}_r + \text{num}(j)) / \text{cnt}_r$ 
 $P = P + \text{num}(j + 1)$ 
 $\text{num}_r = \text{num}_r + \text{num}(j + 1)$ 
 $j = j + 1$ 
 $\text{cnt}_r = \text{cnt}_r + 1$ 
else
 $P = P + \text{num}(i - 1)$ 
 $\text{num}_l = \text{num}_l + \text{num}(i - 1)$ 
 $i = i - 1$ 
 $\text{cnt}_l = \text{cnt}_l + 1$ 
end
 $T_l = D(i)$ 
 $T_r = D(j)$ 
    
```

ALGORITHM 1

its neighborhood samples and a'_i as its predicted value. As in formula (5), $F(\cdot)$ represents predictor:

$$a'_i = F(a_{i-3}, a_{i-2}, a_{i-1}). \quad (5)$$

Step 2. Define p_i as prediction error $p_i = a_i - a'_i$; then we embedded watermarking bit b by expansion of prediction error p_i . The formula is as follows:

$$p'_i = 2 \times p_i + b. \quad (6)$$

Step 3. Determine the linear prediction coefficients based on DE algorithm. We can define $F(\cdot) = \text{round}(c_1 \times a_{i-3} + c_2 \times a_{i-2} + c_3 \times a_{i-1})$ ($i > 3$). According to the optimization scheme

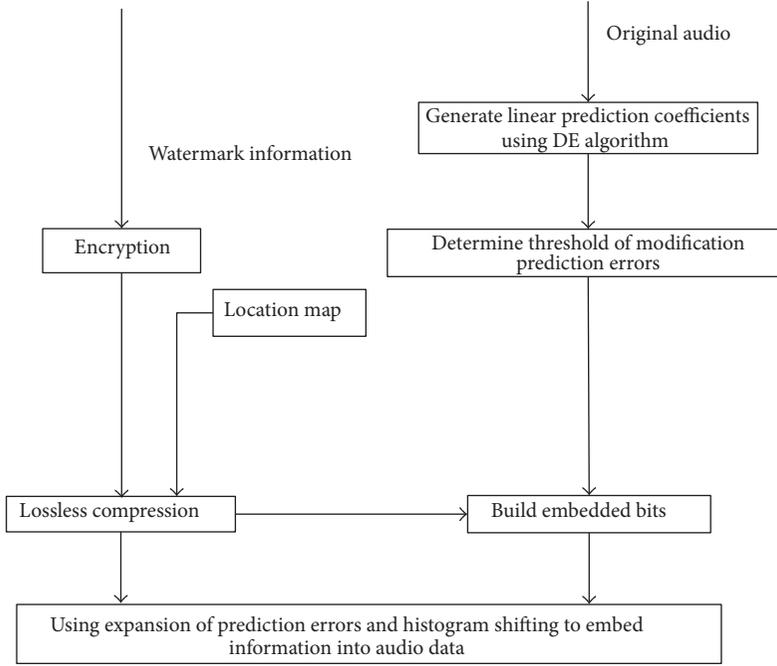


FIGURE 3: Diagram of watermark embedding process.

proposed in Section 2, we can obtain optimal prediction coefficients $C = [c_{1\text{best}}, c_{2\text{best}}, c_{3\text{best}}]$.

Step 4. Give capacity P . We can obtain threshold T_l, T_r based on Algorithm 1.

Step 5. Embed watermarking information using histogram shifting and expansion of prediction errors. T_{eml} and T_{emr} are left and right boundaries of histogram. We can obtain inner region $[T_l, T_r]$ and outer regions $[T_{\text{eml}}, T_l) \cup (T_r, T_{\text{emr}}]$. Finally, embed the watermarking information using expansion of prediction errors or achieve location marking using histogram shifting. The formula is described as follows:

$$p'_i = \begin{cases} p_i - \text{abs}(T_l) & \text{if } p_i \in [T_{\text{eml}}, T_l) \\ 2 \times p_i + b & \text{else if } p_i \in [T_{\text{eml}}, T_{\text{emr}}] \\ p_i + \text{abs}(T_r) + 1 & \text{otherwise.} \end{cases} \quad (7)$$

Step 6. Build location map. Since $a_i \in [32767, -32768]$, in order to prevent overflow and underflow problems during embedding process, we assign a value "1" in location map when the above situations occur. Otherwise we assign a value "0". The location map should be lossless compressed. We use run-length coding and Huffman coding scheme.

In the process of decoding, we can extract the watermark information and restore host audio data. The details are as follows.

Step 1. The current audio sample is obtained from (5) by using restored host data $a(i)$ ($i > 3$). If the value of location map is "1," keep the audio sample unchanged. For prediction errors

ranges in $[2T_l, 2T_r + 1]$, we can extract the watermark bit and restore audio sample using

$$\begin{aligned} b &= \text{mod}(p'_i, 2), \\ p_i &= \left\lfloor \frac{p'_i - b}{2} \right\rfloor, \\ a_i &= a'_i - p_i. \end{aligned} \quad (8)$$

Step 2. For histogram shifting, we can restore the audio data by the following formula:

$$p_i = \begin{cases} p'_i + \text{abs}(T_l) & \text{if } p'_i < 2 \times T_l \\ p'_i - \text{abs}(T_r) - 1 & \text{if } p'_i > 2 \times T_r + 1. \end{cases} \quad (9)$$

4. Experiment Set

In this section, we choose five common audio types to demonstrate the performance of the proposed algorithm. All audio files are 16-bit mono in wave format and the sampling rates of them are 44.1 kHz. The detailed description is listed in Table 1. For optimization of linear prediction coefficients based on DE algorithm, we set the size of initial population to 65. The iteration is set to 50. The probability of crossover and mutation are 0.9 and 0.09, respectively.

4.1. Evaluation Method. In order to evaluate the performance of the proposed algorithm, we choose two common indicators SNR and capacity. SNR is a statistical difference metric

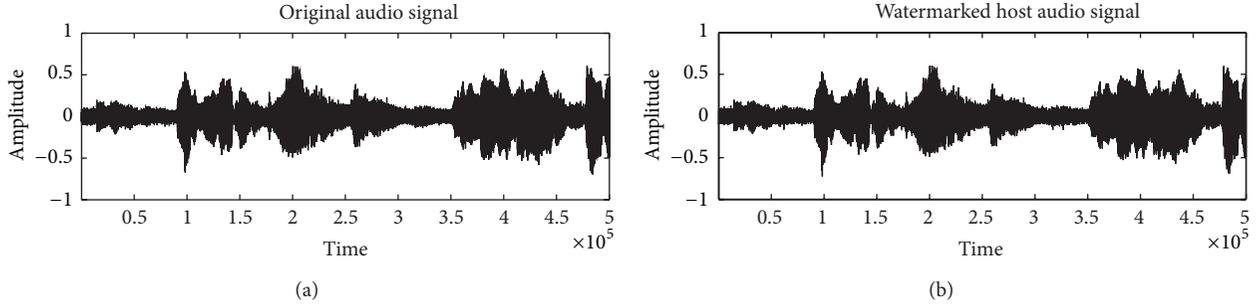


FIGURE 4: Original pop audio and the watermarked audio signal.

TABLE 1: Test database 16 bits.

Index	Audio description
1	Blues
2	Classic
3	Jazz
4	Pop
5	Country

to measure the perceptual similarity between original audio and embedded audio. The formula is as follows:

$$SNR(S, S') = 10 \log_{10} \left(\frac{\sum_{i=1}^L S^2(i)}{\sum_{i=1}^L [S(i) - S'(i)]^2} \right), \quad (10)$$

where $S(i)$, $S'(i)$ are original audio sample and embedded audio sample, respectively. Capacity is used to measure how much information can be embedded into a host data. The formula is as follows: where N_L means the required number of audio sample when embedding one bit watermark during embedding procedure

$$P = \frac{1}{N_L}. \quad (11)$$

5. The Experimental Results and Analysis

Using the ED algorithm settings in Section 4, the convergence curves of object function with DE algorithm are shown in Figure 5. The best fitness of an individual can converge fast to a stable value in the twenty-fifth generation. The comparison of SNR and capacity between embedded audio using the optimal linear prediction coefficients and best linear prediction coefficients proposed in [10] was shown in Table 2. We can see that the linear prediction coefficients generated by DE algorithm achieve better SNR and capacity than fixed coefficients for all the types of test audios in the same condition. Figure 6 presents the performance of the 64 prediction coefficient sets and optimal prediction coefficient set for classic audio. It can be seen that linear prediction coefficients obtained by DE have obvious advantage in SNR and capacity. The original and watermarked audio signals in time domain are shown in Figure 4; the difference between them is invisible.

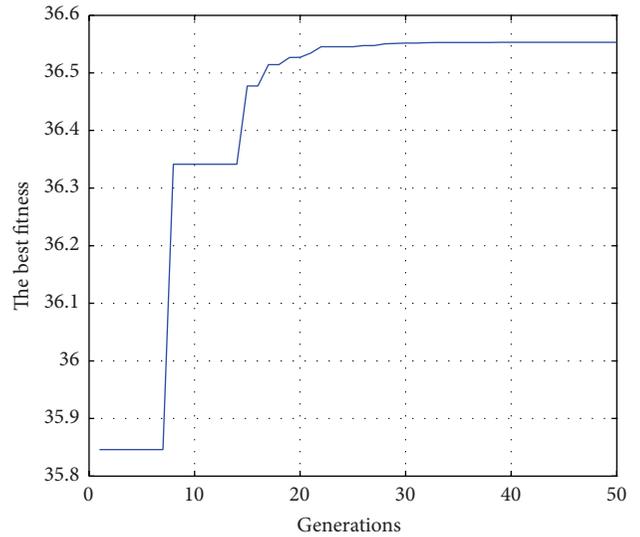


FIGURE 5: Convergence curves of object function with DE algorithm.

TABLE 2: The comparison of SNR and capacity between embedded audio using the optimal linear prediction coefficients and best linear prediction coefficients proposed in [10].

Audio	Set $T = 5000$			
	Ours		[10]	
	SNR	Capacity	SNR	Capacity
Blues	38.14	0.99	37.00	0.99
Classic	15.32	0.99	10.06	0.99
Jazz	16.73	0.99	16.07	0.97
Pop	18.13	0.99	15.87	0.99
Country	28.48	0.99	26.64	0.99

Figure 7 graphically depicts the embedding capacity versus the embedded audio quality curve of five types of audio. We can see that the SNR of embedded audio decreased with the increase of embedding capacity. According to the given embedding capacity, we can conveniently determine and modify the threshold of prediction error. That enhances capacity control capability. In order to evaluate the performance of proposed algorithm based on histogram shifting, we use run-length coding and Huffman coding scheme to

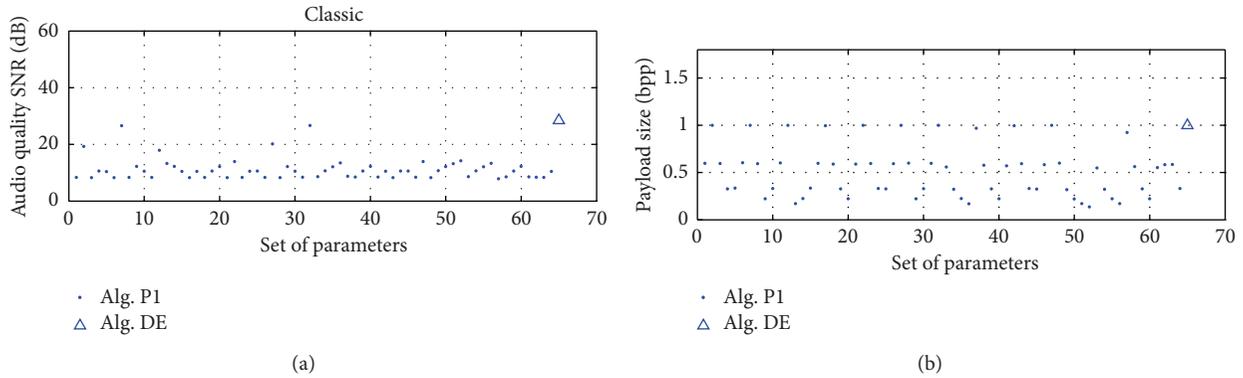


FIGURE 6: The performance of the 64 prediction coefficient sets and optimal prediction coefficient set for classic audio.

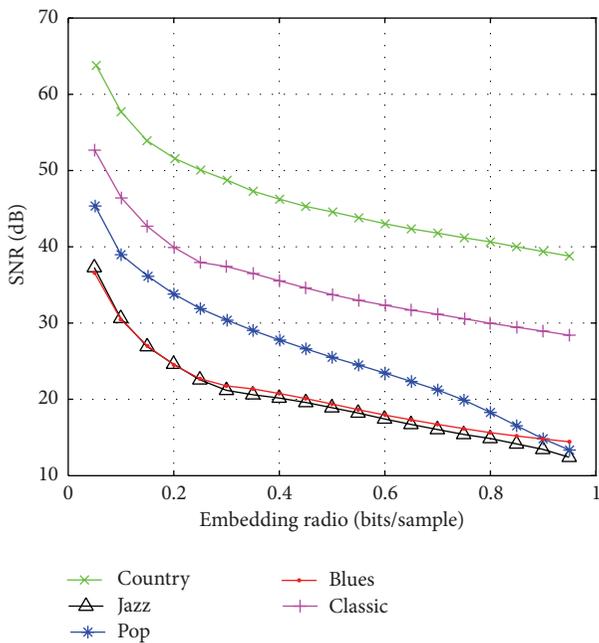


FIGURE 7: The embedding capacity versus the embedded.

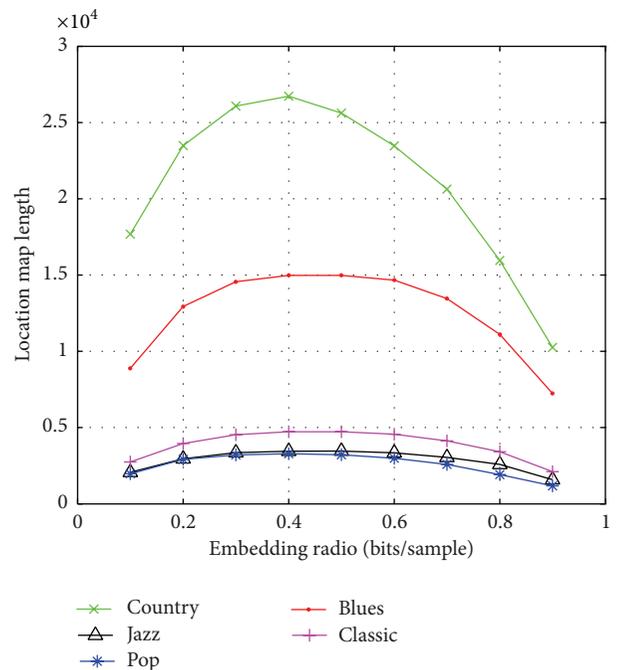


FIGURE 8: The embedding capacity versus compressed location map length curve of five types audio quality curve of five types audio.

compress the location map. Figure 8 presents the embedding capacity versus compressed location map length curve of five types of audio. The location map is generated using algorithm proposed in [11]. We can see that location map of algorithm compared has a large length and the compressibility of location map is low. The large auxiliary location map will take a lot of capacity. In this paper, we adopt histogram shifting to reduce the location map. As we know, we can map the audio data to integer ranged from -32768 to 32767 . As the range of audio data is very wide, the overflow phenomenon rarely occurs. Meanwhile, the location map of proposed algorithm only needs overflow location information, so the length of location map can be reduced drastically. The experiments showed that the length of location map using proposed algorithm is negligible.

6. Conclusion

This paper proposed a novel reversible audio watermarking algorithm based on improved prediction error expansion and histogram shifting. As the performance of reversible audio scheme using prediction error expansion is affected by the predictor, we proposed the optimization of linear prediction coefficients using differential evolution algorithm. The experiments have shown that the proposed scheme can achieve better embedded audio quality and higher capacity. In order to reduce the length of location map, we introduced histogram shifting scheme. In addition, we proposed the scheme which can be used to compute prediction error modification threshold according to a given embedding capacity. The simulation results verify our scheme can drastically

reduce location map bits length and enhance capacity control capability.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was sponsored by National Nature Science Foundation of China (61173106), Specialized Research Fund for the Doctoral Program of Higher Education, China (20100161120021), and the Young Teacher's Growth Program of Hunan University.

References

- [1] J. B. Feng, I. C. Lin, C. S. Tsai, and Y. P. Chu, "Reversible watermarking: current status and key issues," *International Journal of Network Security*, vol. 2, no. 3, pp. 161–170, 2006.
- [2] M. Voigt, B. Yang, and C. Busch, "Reversible watermarking of 2D-vector data," in *Proceedings of the Multimedia and Security Workshop*, pp. 160–165, ACM, September 2004.
- [3] D. Coltuc, "Low distortion transform for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 412–417, 2012.
- [4] S. Lee, C. D. Yoo, and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 321–330, 2007.
- [5] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, 2010.
- [6] D. Coltuc and J.-M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 255–258, 2007.
- [7] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.
- [8] D. M. Thodi and J. J. Rodríguez, "Reversible watermarking by prediction-error expansion," in *Proceedings of the 6th IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 21–25, March 2004.
- [9] H. Zhao, F. Wang, Z. Chen, and J. Liu, "A robust audio watermarking algorithm based on SVD-DWT," *Electronics & Electrical Engineering*, vol. 20, no. 1, pp. 75–80, 2014.
- [10] D. Yan and R. Wang, "Reversible data hiding for audio based on prediction error expansion," in *Proceedings of the 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP '08)*, pp. 249–252, August 2008.
- [11] A. Nishimura, "Reversible audio data hiding using linear prediction and error expansion," in *Proceedings of the 7th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP '11)*, pp. 318–321, October 2011.
- [12] R. Storn and K. Price, "Minimizing the real functions of the ICEC'96 contest by differential evolution," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 842–844, 1996.
- [13] J. Vesterstrøm and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, vol. 2, pp. 1980–1987, June 2004.
- [14] B. V. Babu and R. Angira, "Modified differential evolution (MDE) for optimization of non-linear chemical processes," *Computers and Chemical Engineering*, vol. 30, no. 6-7, pp. 989–1002, 2006.
- [15] B. Liu, J. Lu, Y. Wang, and Y. Tang, "An effective parameter extraction method based on memetic differential evolution algorithm," *Microelectronics Journal*, vol. 39, no. 12, pp. 1761–1769, 2008.
- [16] W. Zhang, X. Wang, B. Lu, and T. Kim, "Secure encapsulation and publication of biological services in the cloud computing environment," *BioMed Research International*, vol. 2013, Article ID 170580, 8 pages, 2013.
- [17] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–361, 2006.
- [18] D. M. Thodi and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721–730, 2007.

Research Article

Dual Key Speech Encryption Algorithm Based Underdetermined BSS

Huan Zhao,¹ Shaofang He,^{1,2} Zuo Chen,¹ and Xixiang Zhang¹

¹ School of Information Science and Technology, Hunan University, Changsha, Hunan 410082, China

² Science College, Hunan Agricultural University, Changsha, Hunan 410128, China

Correspondence should be addressed to Zuo Chen; chenzuo@hnu.edu.cn

Received 10 March 2014; Accepted 27 April 2014; Published 14 May 2014

Academic Editor: Fei Yu

Copyright © 2014 Huan Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

When the number of the mixed signals is less than that of the source signals, the underdetermined blind source separation (BSS) is a significant difficult problem. Due to the fact that the great amount data of speech communications and real-time communication has been required, we utilize the intractability of the underdetermined BSS problem to present a dual key speech encryption method. The original speech is mixed with dual key signals which consist of random key signals (one-time pad) generated by secret seed and chaotic signals generated from chaotic system. In the decryption process, approximate calculation is used to recover the original speech signals. The proposed algorithm for speech signals encryption can resist traditional attacks against the encryption system, and owing to approximate calculation, decryption becomes faster and more accurate. It is demonstrated that the proposed method has high level of security and can recover the original signals quickly and efficiently yet maintaining excellent audio quality.

1. Introduction

As speech communications in our daily life become more and more common, the importance of providing a high level of security is sharply increasing. For that reason, a series of speech encryption methods have been proposed. Among which, the analogue encryption is one of the most popular encryption techniques widely used in speech communication. Generally, there are four categories of cryptographic algorithms in speech communication: frequency-domain scrambling (e.g., the frequency inverter and the band splitter), time-domain scrambling (e.g., the time element scrambling), amplitude scrambling (also known as the masking technique that covers the speech signal by the linear addition of pseudorandom amplitudes), and two-dimensional scrambling that combines the frequency-domain scrambling with the time-domain scrambling [1, 2]. In addition, there are many other analogue speech encryption algorithms in the transform domain, for example, discrete cosine transform, fast Fourier transform, wavelet transform, and so forth [3–5]. Up to date, many new speech encryption algorithms including BSS-based [2, 6], chaotic cryptosystem [7–10], and encryption using circulant transformations [11] have been

developed. Due to the fact that the great amount data of speech communications and real-time communication has been required, it is not suitable to utilize traditional encryption methods directly for speech communication encryption. As such, to explore speech encryption methods that have a high level of security and efficiency and high speed in decryption while retaining excellent audio quality is an urgent issue.

Blind source separation (BSS) is used to recover unknown source or signals that are independent mutually of their observed mixtures without knowing the mixing coefficients. So, it is also known as independent component analysis (ICA). Recently, signals encryption has been more applied on the image cryptosystems [12–14] but less on the speech encryption. Speech encryption method based BSS, of which the security dependent on the difficulty of solving the underdetermined BSS problem where the number of the observed mixed signals is less than that of the source signals. The sufficient condition for constructing the underdetermined mixing matrix for encryption is presented based on the source inseparability of BSS [15].

One-time pad [16] is a simple and completely unbreakable symmetric cipher, and it has two perfect characteristics: the key is random and has the same length as the message.

The key space is large enough to resist brute-force attacks as long as the key is long enough. So, the message is secure as long as the key is protected. Motivated by the randomness and initial conditions' sensitivity of chaotic signals, we present an underdetermined BSS-based dual key speech encryption scheme in this paper. The dual key of which are random signals (one-time pad) and chaotic signals, namely, key signals I and II, respectively. The main purpose of the algorithm is to mask the original speech signals by mixing the original speech with key signals I and II. In the decryption process, approximate calculation method is used to recover the original speech signals. The underdetermined blind source separation is a significant challenge in blind source separation (BSS) where the number of the source signals is greater than that of the mixed signals. In addition, the using of the key signals I and II (one-time pad and chaotic signal) ensures high security of the algorithm. Both extensive computer simulations and performance analysis results show that the proposed method has high level of security while retaining excellent audio quality.

The rest of this paper is organized as follows. In Section 2, firstly, we introduce the BSS mixing model and the underdetermined BSS problem briefly, secondly, the details of speech encryption and decryption are described, and finally, we analyze the feasibility of approximate calculation in the decryption process. Sections 3 and 4 conduct computer simulations to illustrate and analyze the performance of the method. We conclude this paper in Section 5.

2. Proposed Method

2.1. BSS Mixing Model and Underdetermined Problem [2]. Suppose that $s_1(t), s_2(t), \dots, s_M(t)$ is M independent source signals and N observed mixtures of the source signals are $x_1(t), x_2(t), \dots, x_N(t)$ ($M \ll N$). The linear BSS mixing model is represented as follows:

$$x(t) = A_e s(t), \quad (1)$$

where $s(t) = [s_1(t), s_2(t), \dots, s_M(t)]^T$, which is $M \times 1$ column vector collected from the source signals, similarly, $N \times 1$ column vector $x(t) = [x_1(t), x_2(t), \dots, x_N(t)]^T$ collects the observed signals, and A is an $N \times M$ mixing matrix that contains the mixing coefficients. The aim of BSS is to find a $M \times N$ demixing matrix W such that output vector:

$$u(t) = Wx(t) = WAs(t) = PDs(t), \quad (2)$$

where $P \in R^{M \times M}$ is a permutation matrix and $D \in R^{M \times M}$ is a diagonal scaling matrix. When the number of the mixed signals is less than that of the source signals; that is, $M > N$, BSS becomes a difficult case of the underdetermined problem, in which the complete separation of the source signals is impossible.

2.2. Encryption. The main idea of the proposed algorithm is to construct the intractable underdetermined BSS problem in encryption, and in decryption it can only be solved with

the dual key. The block diagram of the underdetermined BSS-based speech encryption scheme is shown in Figure 1.

Two main steps in the encryption process are the segment splitter and the underdetermined mixing. Suppose that the original speech is divided into frames, and every frame is encrypted, respectively; q is the frame pointer. The frame q is encrypted as follows.

- (1) Segment splitter: the segment splitter first partitions the frame q into P segments $s_1(t), s_2(t), \dots, s_p(t)$, $t = 1, \dots, T$, where T is the segment length.
- (2) Underdetermined mixing: the source signals are composed of three parts, they are original speech signals, key signals I generated by pseudorandom number generator (PRNG) with secret seed I_0 , and key signals II from chaotic system. $s(t) = [s_1(t), s_2(t), \dots, s_p(t)]^T$ denotes original speech, key signals I are $k(t) = [k_1(t), k_2(t), \dots, k_p(t)]^T$, and $h(t) = [h_1(t), h_2(t), \dots, h_p(t)]^T$ is key signals II. Therefore, $3p \times 1$ column vector of the source signals is $[s^T(t), k^T(t), h^T(t)]^T$. A $p \times 3p$ underdetermined mixing matrix $A_e = [B \ \alpha B \ \beta B]$ for encryption is first generated randomly, where B is a $P \times P$ matrix of full rank, which is pseudorandomly generated with normal distribution between -1 and 1 , $1 \ll \alpha$, $\beta \ll 2$ are scalar values to make the original speech be covered well by the dual key signals. The encryption equation can be represented as follows:

$$\begin{aligned} x(t) &= A_e [s^T(t), k^T(t), h^T(t)]^T \\ &= [B \ \alpha B \ \beta B] \begin{pmatrix} s(t) \\ k(t) \\ h(t) \end{pmatrix} \\ &= Bs(t) + \alpha Bk(t) + \beta Bh(t), \end{aligned} \quad (3)$$

where $x(t) = [x_1(t), x_2(t), \dots, x_p(t)]^T$ is the observed signals. (Parameters P, T , secret seed I_0 , initial condition of chaotic system, and scalar α, β are inserted into the head data of the encryption speech in a definite format for transmission.)

2.3. Decryption. Once the mixture signals $x(t) = [x_1(t), x_2(t), \dots, x_p(t)]^T$ are received, the key signals I are regenerated by the secret seed I_0 and the key signals II are produced by the chaotic system using the initial conditions. Usually, BSS is then performed [2, 17] to recover original signals. But in this paper, we employ approximate calculation to recover original signals.

2.3.1. The Approximate Calculation for Decryption. Multiply $k^T(t)$ at both sides of (3), and we get equation:

$$x(t) k^T(t) = Bs(t) k^T(t) + \alpha Bk(t) k^T(t) + \beta Bh(t) k^T(t). \quad (4)$$

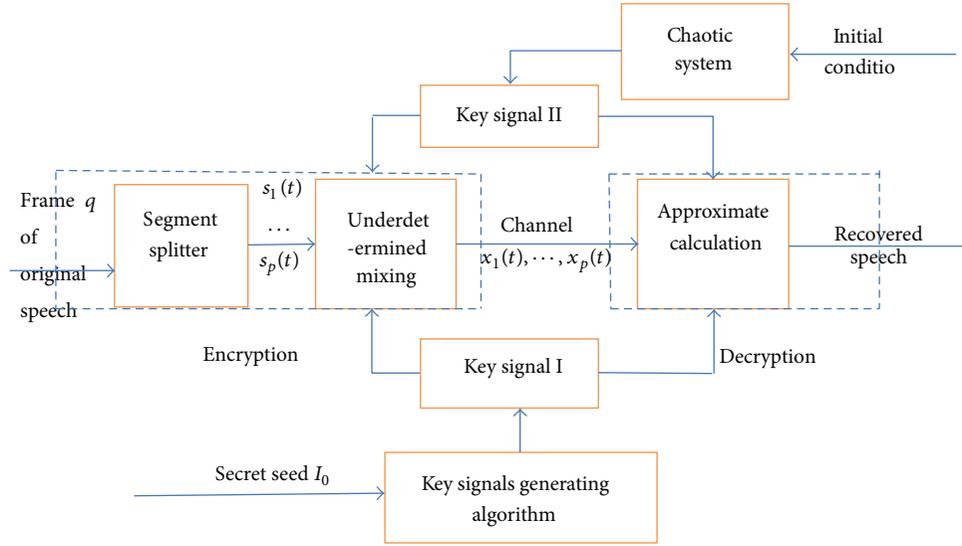


FIGURE 1: Block diagram of underdetermined BSS-based double key speech encryption.

Similarly, multiply $h^T(t)$ at both sides of (3), and then get equation

$$x(t)h^T(t) = Bs(t)h^T(t) + \alpha Bk(t)h^T(t) + \beta Bh(t)h^T(t) \quad (5)$$

denoted by

$$\begin{aligned} R_{xk} &= x(t)k^T(t), & R_{sk} &= s(t)k^T(t), \\ R_{kk} &= k(t)k^T(t), & R_{hk} &= h(t)k^T(t), \\ R_{xh} &= x(t)h^T(t), & R_{sh} &= s(t)h^T(t), \\ R_{kh} &= k(t)h^T(t), & R_{hh} &= h(t)h^T(t). \end{aligned} \quad (6)$$

Equations (4) and (5) can be represented by (7) and (8), respectively,

$$R_{xk} = BR_{sk} + \alpha BR_{kk} + \beta BR_{hk}, \quad (7)$$

$$R_{xh} = BR_{sh} + \alpha BR_{kh} + \beta BR_{hh}. \quad (8)$$

Since the original speech signals are independent statistically of the key signals I and II, we have $R_{sk} \ll R_{kk}$, $R_{sh} \ll R_{hh}$, that is, $BR_{sk} \ll \alpha BR_{kk}$, $BR_{sh} \ll \beta BR_{hh}$, $1 \ll \alpha, \beta \ll 2$; therefore, (7) and (8) are represented approximately by (9) and (10):

$$R_{xk} \approx \alpha BR_{kk} + \beta BR_{hk}, \quad (9)$$

$$R_{xh} \approx \alpha BR_{kh} + \beta BR_{hh}. \quad (10)$$

Plus (9) and (10), we can get equation:

$$R_{xk} + R_{xh} \approx B(\alpha R_{kk} + \beta R_{hk} + \alpha R_{kh} + \beta R_{hh}). \quad (11)$$

So we obtain an estimate for B as follows:

$$\hat{B} = (R_{xk} + R_{xh})(\alpha R_{kk} + \beta R_{hk} + \alpha R_{kh} + \beta R_{hh})^{-1}. \quad (12)$$

Substituting B in (3) with (11), original signals $s(t)$ can be estimated as

$$\begin{aligned} \hat{s}(t) &= (\alpha R_{kk} + \beta R_{hk} + \alpha R_{kh} + \beta R_{hh}) \\ &\times (R_{xk} + R_{xh})^{-1} x(t) - \alpha k(t) - \beta h(t). \end{aligned} \quad (13)$$

2.3.2. The Steps of Calculating Original Signals for Decryptions.

Parameters P , T and scalar α , β are transmitted together with the secret seed I_0 and initial condition of chaotic system in the head data of the encryption speech. The original speech signals can be decrypted with very high quality by employing the decryption equation (13). Supposing that the mixed signals are received and the double key signals are regenerated, the original signals can be calculated as follows:

- (a) calculate R_{xk} , R_{kk} , R_{xh} , R_{hk} , R_{kh} , R_{hh} respectively;
- (b) calculate $(\alpha R_{kk} + \beta R_{hk} + \alpha R_{kh} + \beta R_{hh})$ and $(R_{xk} + R_{xh})^{-1}$;
- (c) calculate $\hat{s}(t)$ using (13).

2.3.3. Analysis of the Approximate Calculation for Decryption.

Key signals I and II are generated by pseudorandom number generator (PRNG) and chaotic system, respectively. They are both independent statistically of the original signals. For illustrating the feasibility of the approximate calculation, we compute values of R_{sk} , R_{kk} , R_{sh} , and R_{hh} in the example. The original signals of digital "1" in English are regarded as a frame, which is divided into two segments, that is, $P = 2$, $T = 16000$, the original signals $s(t) = [s_1(t), s_2(t)]^T$, correspondingly key signals I and II are $k(t) = [k_1(t), k_2(t)]^T$, $h(t) = [h_1(t), h_2(t)]^T$. The key signals I are generated by PRNG with the secret seed $I_0 = p \times T = 32000$, and choosing initial condition $(a, b, c) = (35, 3, 28)$, $[h(1), y(1), z(1)] =$

TABLE 1: Comparison of average values of diagonal elements and upper triangular elements for R_{sk} , R_{kk} , R_{sh} , and R_{hh} .

Segment	Average values of diagonal elements				Average values of upper triangular elements			
	R_{kk}	R_{sk}	R_{hh}	R_{sh}	R_{kk}	R_{sk}	R_{hh}	R_{sh}
$p = 2, T = 16000$	5365.35	1.3775	876.29	-0.5212	4914.5	1.4676	488.4	0.087
$p = 4, T = 8000$	2682.7	0.6888	438.15	-0.2606	2281.4	0.5392	163.1	0.170
$P = 8, T = 4000$	1343.96	0.0645	219.08	-0.1303	1092.6	0.1002	57.99	0.051

$[0, 1.001, 0]$, the key signals II ($h(t)/40$) are generated by Chen-Lee chaotic system [18]:

$$\begin{aligned} \dot{h} &= -yz + ah, \\ \dot{y} &= hz + by, \\ \dot{z} &= \frac{1}{3}hy + cz. \end{aligned} \quad (14)$$

Finally, we calculate the values of R_{sk} , R_{kk} , R_{sh} , R_{hh} and get the average values of diagonal elements value of $R_{sk} = s(t)k^T(t)$, $R_{kk} = k(t)k^T(t)$, $R_{sh} = s(t)h^T(t)$, and $R_{hh} = h(t)h^T(t)$ are 1.3775, 5365.35, -0.5212, and 876.29, respectively.

In the example, the original signals are also splitted into other different numbers of segments, which are $P = 4, 8$, corresponding to $T = 8000, 4000$, use the same key signals I and II, and compute R_{sk} , R_{kk} , R_{sh} , and R_{hh} , respectively. Table 1 shows the results of the average value of diagonal elements and upper-triangular for comparison of R_{sk} , R_{kk} , R_{sh} , and R_{hh} in three different cases.

From Table 1 we can see that the average values of diagonal elements of R_{sk} are considerably much smaller than those of R_{kk} , and the average values of upper-triangular elements of R_{sh} are also much smaller than those of R_{hh} , and obviously, $R_{sk} \ll R_{kk}$, $R_{sh} \ll R_{hh}$; therefore, using approximate calculation for decryption is feasible, which means that the decryption method in this paper not only have the characteristic of computing simply and quickly but also maintaining excellent audio quality.

3. Computer Simulations

In order to illustrate the feasibility of the proposed scheme, we carry out extensive computer simulation. In common experiments, recorded audio files in wave format are adopted and transmitted within local area network. In our experiment, we use the speech file recording a man saying the digit "1" in English. The speech signals are sampled at 16 KHz, as shown in Figure 2(a). For the purpose of simplifying experiment process, we regard the speech signals as one frame directly and separate them into two segments; that is, $p = 2$, $T = 16000$, as shown in Figure 2(a). The key signals I are generated by PRNG with the secret seed $I_0 = p \times T = 32000$, and Figure 2(b) is the split wave of key signals I. Choosing initial condition $(a, b, c) = (35, 3, 28)$ and $[h(1), y(1), z(1)] = [0, 1.001, 0]$, the key signals II ($h(t)/40$) are generated by Chen-Lee chaotic system [18]:

$$\dot{h} = -yz + ah,$$

$$\dot{y} = hz + by,$$

$$\dot{z} = \frac{1}{3}hy + cz, \quad (15)$$

and the split wave of which is showed in Figure 2(c). Choosing $\alpha = \beta = 2$, the underdetermined mixing matrix $A_e = [B \ \alpha B \ \beta B]$ used for simulation is

$$\begin{aligned} A_e &= \begin{pmatrix} 0.9501 & 0.6068 & 1.9003 & 1.2137 & 1.9003 & 1.2137 \\ 0.2311 & 0.4860 & 0.4623 & 0.9720 & 0.4623 & 0.9720 \end{pmatrix} \\ & \quad (16) \end{aligned}$$

using (3), and two cipher texts are deduced quickly. Figure 2(d) shows the two cipher-text segments. Obviously, the original speech signals are well covered with the mixed sets of key signals I and II. In the decryption process, the mixed signals are received and the double key signals are regenerated; we can regain the original speech signals according to the steps of approximate calculation decryption method. The recovered signals are showed in Figure 2(e).

4. Performance Analysis

4.1. Signal-to-Noise Ratio Computation. For the purpose of quantifying the performance of the proposed method, we calculated the signal-to-noise ratio (SNR) index of original signals segments in each encrypted signal segments and decrypted signals segments. Particularly, the SNR index of original segments in the decrypted segments is represented as follows:

$$\text{SNR (dB)} = 10 \log \left\{ \frac{\sum_{t=0}^T [s(t) - \hat{s}(t)]^2}{\sum_{t=0}^T s^2(t)} \right\}, \quad (17)$$

where $s(t)$ is original signals and $\hat{s}(t)$ is decrypted original signals that are calculated by the approximate calculation method. If $\hat{s}(t)$ is replaced by $x(t)$, which denotes encrypted signals, we can obtain the SNR index of encrypted signals. Employing the data in computer simulations, we can get SNR of two original signals segments in two encrypted segments and two decrypted signals segments. Table 2 shows the results.

These SNR indexes show that in the encrypted segments dual key signals have well masked the original segments, and the original signals in the decrypted segments that are recovered by approximate calculation method have excellent quality.

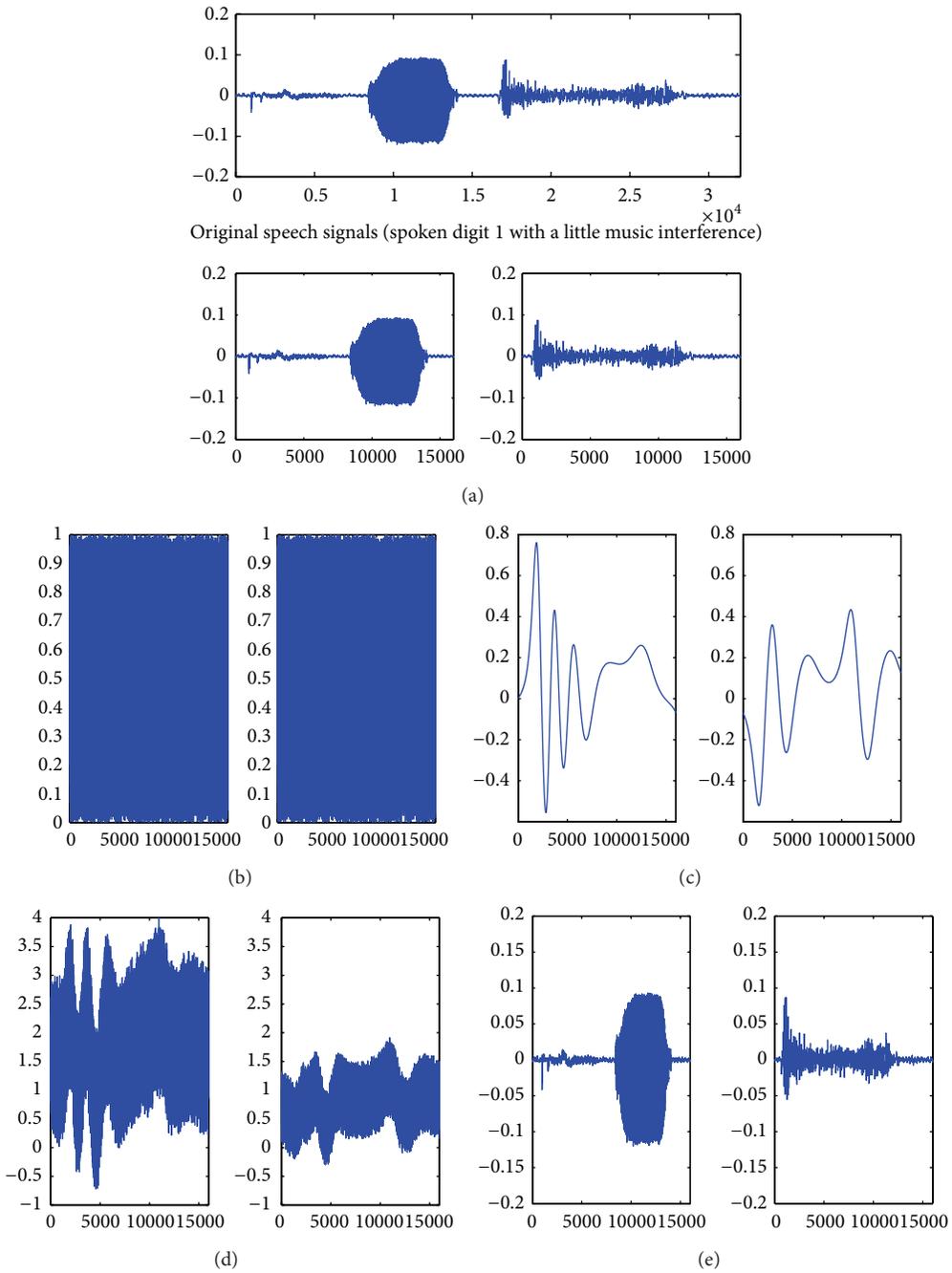


FIGURE 2: (a) The original speech signals and two segments $s_1(t)$, $s_2(t)$; (b) two segments of key signals I $k_1(t)$, $k_2(t)$; (c) two segments of key signals II $h_1(t)$, $h_2(t)$; (d) two encrypted segments $x_1(t)$, $x_2(t)$; (e) two recovered segments by approximate calculation $\hat{s}_1(t)$, $\hat{s}_2(t)$.

TABLE 2: SNR (dB) of two original signals segments in two encrypted segments and two decrypted signals segments.

Original signals segments	Encrypted signals segments		Decrypted signals segments	
	x_1	x_2	\hat{s}_1	\hat{s}_2
s_1	81.4021	65.5826	-94.595	1.0563
s_2	105.586	89.7596	25.2364	-101.7182

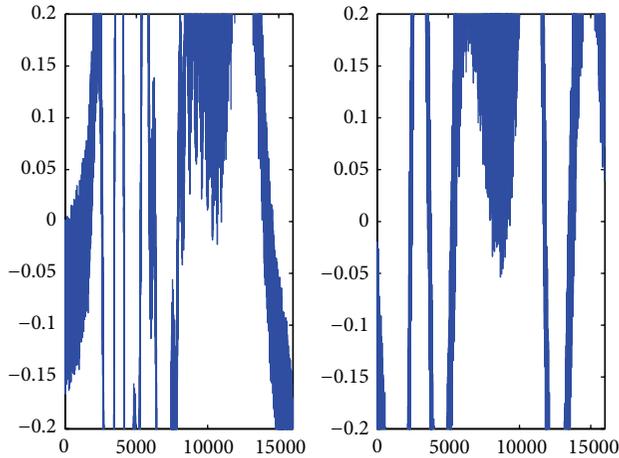


FIGURE 3: Decrypted signals of the proposed method with mismatch of secret keys.

4.2. Security Analysis. We take advantage of the underdetermined BSS problem to propose a dual key encryption algorithm in this paper. There are three aspects to ensure the security of the algorithm. Firstly, the intractability of the underdetermined BSS problem can be ensured by the mixing matrix for encryption. Secondly, the key signals I that as long as the original speech signals have the perfect property of the one-time pad cipher, which is statistically independent and non-Gaussian characteristics.

Finally the key signals II are generated from the Chen-Lee chaotic system that has the characteristic of power randomness and high sensitivity to initial condition. In order to illustrate the sensitivity of the proposed encryption algorithm to secret keys, we choose an estimate of the initial condition that is $(a, b, c) = (35, -3, 28)$, $[h(1), y(1), z(1)] = [0.001, 1, 0.001]$, in which there is a slight mismatch with the real initial condition and use the same key signals I with secret seed $I_0 = 32000$. Figure 3 shows two segments of the recovered signals utilizing the approximate calculation for decryption. Obviously, the recovered signals with the wrong secret key are totally different from the original speech signals. In short, the proposed method is sensitive to secret keys and immune against the ordinary attacks on cryptosystems, for example, cipher-text-only attack, known-plaintext attack, chosen-plaintext attack, and brute-force attack.

5. Conclusions

In this paper, we proposed a new dual key encryption scheme based on the underdetermined BSS problem. Since the mixing matrix for encryption ensures the intractability of the underdetermined BSS problem, and the key signals I approximately have the perfect property of the one-time pad cipher, and the key signals II are generated from the chaotic system that has the characteristic of power randomness and high sensitivity to initial condition. In the decryption process, using approximate calculation decryption method can recover the original signals quickly and efficiently yet maintaining high level audio quality. The design of this

encryption method has five merits: (1) it is impossible to recover the original signals without the parameters in the head data of the encryption speech signals; (2) the approximate calculation method used in decryption process ensures the recovery of the original signals efficiently yet maintaining excellent speech quality; (3) the key signals I approximately has the perfect property of the one-time pad cipher, in which the length of the key is the same as the original speech signals; hence, the space of the keys is so large that all brute-force attacks against the system are infeasible; (4) the key signals II generated from chaotic system that provides the present scheme having the property of cipher text are very sensitive to secret keys, and (5) it can resist all kinds of traditional attacks against cryptosystems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was sponsored by the National Nature Science Foundation of China (61173106), Specialized Research Fund for the Doctoral Program of Higher Education of China (20100161120021), the Natural Science Foundation of Hunan Province of China (14JJ3062), and Young Teacher's Growth Program of Hunan University.

References

- [1] H. J. Beker and F. C. Piper, *Secure Speech Communications*, Academic Press, London, UK, 1985.
- [2] Q.-H. Lin, F.-L. Yin, T.-M. Mei, and H. Liang, "A blind source separation based method for speech encryption," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 6, pp. 1320–1328, 2006.
- [3] B. Goldberg, S. Sridharan, and E. Dawson, "Design and cryptanalysis of transform-based analog speech scramblers," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, pp. 735–744, 1993.
- [4] A. Matsunaga, K. Koga, and M. Ohkawa, "Analog speech scrambling system using the FFT technique with high-level security," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 4, pp. 540–547, 1989.
- [5] F.-L. Ma, J. Chen, and Y.-M. Wang, "Wavelet transform-based analogue speech scrambling scheme," *Electronics Letters*, vol. 32, no. 8, pp. 719–721, 1996.
- [6] L.-J. Sheu, H.-S. Chiou, and W.-C. Chen, "A semi-one time pad using blind source separation for speech encryption," *World Academy of Science, Engineering and Technology*, vol. 5, no. 8, 2011.
- [7] K. Li, Y. C. Soh, and Z. G. Li, "Chaotic cryptosystem with high sensitivity to parameter mismatch," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, no. 4, pp. 579–583, 2003.
- [8] S. M. H. Alwabhani and E. B. M. Bashier, "Speech scrambling based on chaotic maps and one time pad," in *Proceedings of the International Conference on Computing, Electrical and Electronics Engineering (ICCEEE '13)*, pp. 128–133, August 2013.

- [9] L. J. Sheu, "A speech encryption using fractional chaotic systems," *Nonlinear Dynamics*, vol. 65, no. 1-2, pp. 103–108, 2011.
- [10] Y. P. Zhang, F. Duan, and X. Liu, "The research of applying chaos theory to speech communicating encryption system," in *Advances in Multimedia, Software Engineering and Computing Vol.2*, vol. 129 of *Advances in Intelligent and Soft Computing*, pp. 197–202, Springer, Berlin, Germany, 2011.
- [11] G. Manjunath and G. V. Anand, "Speech encryption using circulant transformations," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '02)*, vol. 1, pp. 553–556, 2002.
- [12] W. Kasprzak and A. Cichocki, "Hidden image separation from incomplete image mixtures by independent component analysis," in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 2, pp. 394–398, Vienna, Austria, 1996.
- [13] C.-C. Chang, M.-S. Hwang, and T.-S. Chen, "A new encryption algorithm for image cryptosystems," *Journal of Systems and Software*, vol. 58, no. 2, pp. 83–91, 2001.
- [14] K. D. Rao, K. P. Kumar, and P. V. M. Krishna, "A new and secure cryptosystem for image encryption and decryption," *IETE Journal of Research*, vol. 57, no. 2, pp. 165–171, 2011.
- [15] X.-R. Cao and R. Liu, "General approach to blind source separation," *IEEE Transactions on Signal Processing*, vol. 44, no. 3, pp. 562–571, 1996.
- [16] B. Schneier, *Applied Cryptography: Protocols, Algorithms and Source Code in C*, John Wiley & Sons, 1996.
- [17] D.-P. Guo and Q.-H. Lin, "Fast decryption utilizing correlation calculation for BSS-based speech encryption system," in *Proceedings of the 6th International Conference on Natural Computation (ICNC '10)*, vol. 3, pp. 1428–1432, August 2010.
- [18] H.-K. Chen and C.-I. Lee, "Anti-control of chaos in rigid body motion," *Chaos, Solitons & Fractals*, vol. 21, no. 4, pp. 957–965, 2004.

Research Article

Efficient Certificate-Based Signcryption Secure against Public Key Replacement Attacks and Insider Attacks

Yang Lu and Jiguo Li

College of Computer and Information Engineering, Hohai University, No. 8, Focheng Xi Road, Jiangning District, Nanjing, Jiangsu 211100, China

Correspondence should be addressed to Yang Lu; luyangnsd@163.com

Received 12 March 2014; Accepted 24 April 2014; Published 12 May 2014

Academic Editor: Tianjie Cao

Copyright © 2014 Y. Lu and J. Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Signcryption is a useful cryptographic primitive that achieves confidentiality and authentication in an efficient manner. As an extension of signcryption in certificate-based cryptography, certificate-based signcryption preserves the merits of certificate-based cryptography and signcryption simultaneously. In this paper, we present an improved security model of certificate-based signcryption that covers both public key replacement attack and insider security. We show that an existing certificate-based signcryption scheme is insecure in our model. We also propose a new certificate-based signcryption scheme that achieves security against both public key replacement attacks and insider attacks. We prove in the random oracle model that the proposed scheme is chosen-ciphertext secure and existentially unforgeable. Performance analysis shows that the proposed scheme outperforms all the previous certificate-based signcryption schemes in the literature.

1. Introduction

Public key cryptography (PKC) is an important technique to realize network and information security. In traditional PKC, a public key infrastructure (PKI) is used to provide an assurance to the users about the relationship between a public key and the holder of the corresponding private key by certificates. However, the need for PKI-supported certificates is considered the main difficulty in the deployment and management of traditional PKC. To simplify the management of the certificates, Shamir [1] introduced the concept of identity-based cryptography (IBC) in which the public key of each user is derived directly from his identity, such as an IP address or an e-mail address, and the corresponding private key is generated by a trusted third party called private key generator (PKG). The main practical benefit of IBC lies in the reduction of need for public key certificates. However, if the KGC becomes dishonest, it can impersonate any user using its knowledge of the user's private key. This is due to the key escrow problem inherent in IBC. In addition, private keys must be sent to the users over secure channels, so private key distribution in IBC becomes a very daunting task.

To fill the gap between traditional PKC and IBC, Al-Riyami and Paterson [2] proposed a new paradigm called certificateless public key cryptography (CL-PKC) in Asiacrypt 2003. CL-PKC eliminates the key escrow problem inherent in IBC. At the same time, it preserves the advantage of IBC which is the absence of certificates and their heavy management overhead. In CL-PKC, a trusted third party called key generating center (KGC) is involved in the process of issuing a partial secret key for each user. The user independently generates its public/private key pair and combines the partial secret key from the KGC with its private key to generate the actual decryption key. By way of contrast to the PKG in IBC, the KGC does not have access to the user's decryption key. Therefore, CL-PKC solves the key escrow problem. However, as partial secret keys must be sent to the users over secure channels, CL-PKC suffers from the distribution problem.

In Eurocrypt 2003, Gentry [3] introduced the notion of certificate-based cryptography (CBC). CBC provides an implicit certification mechanism for a traditional PKI and allows for a periodical update of certificate status. As in traditional PKC, each user in CBC generates his own public/private key pair and requests a certificate from a trusted

TABLE 1: Properties of the related public key cryptosystems.

	Do not require trusted third party	Implicit certificates	Key escrow free	Key distribution free
Traditional PKC	×	×	✓	✓
IBC	×	✓	×	×
CL-PKC	×	✓	✓	×
CBC	×	✓	✓	✓

third party called certifier. The certificate will be pushed only to the owner of the public/private key pair and act as a partial decryption key or a partial signing key. This additional functionality provides an efficient implicit certificate mechanism. For example, in the encryption scenario, a receiver needs both his private key and certificate to decrypt a ciphertext sent to him, while the message sender need not be concerned about the certificate revocation problem. The feature of implicit certification allows us to eliminate third-party queries for the certificate status and simply the public key revocation problem so that CBC does not need infrastructures like CRL and OCSP. Therefore, CBC can be used to construct an efficient PKI requiring fewer infrastructures than the traditional one. Although CBC may be inefficient when a certifier has a large number of users, this problem can be overcome by using *subset covers* [3]. Furthermore, there are no key escrow problem (since the certifier does not know the private keys of users) and key distribution problem (since the certificates need not be kept secret) in CBC.

Table 1 summarizes the comparison of the above cryptosystems.

Since its advent, CBC has attracted great interest in the research community and many schemes have been proposed, including many encryption schemes (e.g., [4–10]) and signature schemes (e.g., [11–16]). As an extension of the signcryption [17] in CBC, Li et al. [18] introduced the concept of certificate-based signcryption (CBSC) that provides the functionalities of encryption and signature simultaneously. As far as we know, there exist three CBSC schemes in the literature so far. In [18], Li et al. proposed the first CBSC scheme based on Chen and Malone-Lee’s identity-based signcryption scheme [19]. However, they did not give a formal proof of their security claim. A subsequent paper by Luo et al. [20] proposed the second CBSC scheme alone with a security model of CBSC. Recently, Li et al. [21] proposed a publicly verifiable CBSC scheme that is provably secure in the random oracle model.

Our Motivations and Contributions. In this paper, we focus on the construction of a CBSC scheme that resists both the public key replacement attacks and the insider attacks.

Public key replacement attack was first introduced into CL-PKC by Al-Riyami and Paterson [2]. In this attack, an adversary who can replace a user’s public key with a value of its choice dupes any other third parties to encrypt messages or verify signatures using a false public key. It seems that this attack does not have effect on CBC since a certifier is employed for issuing a certificate for each user. Unfortunately, some previous research works [13, 16, 22] have demonstrated that it does. In CBC, the certifier does issue the certificates.

However, as introduced above, CBC adopts the implicit certificate mechanism so that only the owner of a certificate needs to check the validity of his certificate and others need not be concerned about the status of his certificate. Thus, a malicious user is able to launch the public key replacement attack against an ill-designed certificate-based cryptographic scheme. We observe that Luo et al.’s CBSC scheme [20] is insecure under this attack. The concrete attack can be found in Section 4 of this paper.

Insider security [23] refers to the security against the attacks made by the insider (i.e., the sender or the receiver). It requires that, even if a sender’s private key is compromised, an attacker should not be able to designcrypt the message generated by the sender and, even with a receiver’s private key, an attacker should not be able to forge a valid signcryption as if generated by the same sender. In contrast to outsider security [23] that refers to the security against the attacks made by the outsider (i.e., any third party except the sender and the receiver), insider security can provide the stronger security for signcryption schemes [24, 25]. Therefore, it has been accepted as a necessary security requirement for a signcryption scheme to achieve. However, none of the previous constructions of CBSC [18, 20, 21] has considered insider security. The previous security models of CBSC [20, 21] only cover the case where the CBSC scheme is attacked by the outsiders. Actually, the public key replacement attack presented in Section 4 also shows that Luo et al.’s CBSC scheme [20] fails in providing insider security.

The main contributions of this paper are as follows.

- (1) We extend previous works by proposing an improved security model for CBSC that accurately models both the public key replacement attacks and the insider attacks. We show that Luo et al.’s CBSC scheme [20] is insecure in our security model.
- (2) We develop a new CBSC scheme and formally prove its security in our improved security model. In the random oracle, we prove that the proposed scheme is chosen-ciphertext secure and existentially unforgeable. To the best of our knowledge, it is the first signcryption scheme that achieves security under both the public key replacement attacks and the insider attacks in the certificate-based cryptographic setting. Furthermore, compared with the previous CBSC schemes, our scheme enjoys better performance, especially in the computation efficiency.

Paper Organization. The rest of this paper is organized as follows. In the next section, we briefly review some

preliminaries required in this paper. In Section 3, we present an improved security model of CBSC. In Section 4, we show that Luo et al.'s CBSC scheme is insecure in our security model. The proposed CBSC scheme is described and analyzed in Section 5. Finally, we draw our conclusions in Section 6.

2. Preliminaries

Let k be a security parameter and p a k -bit prime number. Let G be an additive cyclic group of prime order p and G_T a multiplicative cyclic group of the same order, and let P be a generator of G . A bilinear pairing is a map $e : G \times G \rightarrow G_T$ satisfying the following properties.

- (i) Bilinearity: for all $P_1, P_2 \in G$, and all $a, b \in Z_p^*$, we have $e(aP_1, bP_2) = e(P_1, P_2)^{ab}$.
- (ii) Nondegeneracy: $e(P, P) \neq 1$.
- (iii) Computability: for all $P_1, P_2 \in G$, $e(P_1, P_2)$ can be efficiently computed.

The security of our CBSC scheme is based on the following hard problems.

Definition 1. The computational Diffie-Hellman (CDH) problem in G is, given a tuple $(P, aP, bP) \in G^3$ for unknown $a, b \in Z_p^*$, to compute $abP \in G$.

Definition 2 (see [26]). The bilinear Diffie-Hellman (BDH) problem in (G, G_T) is, given a tuple $(P, aP, bP, cP) \in G^4$ for unknown $a, b, c \in Z_p^*$, to compute $e(P, P)^{abc} \in G_T$.

Definition 3 (see [27]). The collusion attack algorithm with q -traitors (q -CAA) problem in G is given a tuple $(P, \alpha P, (\omega_1 + \alpha)^{-1}P, \dots, (\omega_q + \alpha)^{-1}P, \omega_1, \dots, \omega_q) \in G^{q+2} \times (Z_p^*)^q$ for unknown $\alpha \in Z_p^*$, to compute $(\omega^* + \alpha)^{-1}P$ for some value $\omega^* \notin \{\omega_1, \dots, \omega_q\}$.

Definition 4 (see [28]). The modified bilinear Diffie-Hellman inversion for q -values (q -mBDHI) problem in G is given a tuple $(P, \alpha P, (\omega_1 + \alpha)^{-1}P, \dots, (\omega_q + \alpha)^{-1}P, \omega_1, \dots, \omega_q) \in G^{q+2} \times (Z_p^*)^q$ for unknown $\alpha \in Z_p^*$, to compute $e(P, P)^{(\omega^* + \alpha)^{-1}}$ for some value $\omega^* \in Z_p^* - \{\omega_1, \dots, \omega_q\}$.

3. Improved Security Model for CBSC Schemes

In this section, we present an improved security model for CBSC that covers both public key replacement attack and insider security. Below, we first briefly review the definition of CBSC.

Formally, a CBSC scheme is specified by the following five algorithms.

- (i) *Setup*(k): on input a security parameter $k \in Z^+$, this algorithm generates a master key msk and a list of public parameters $params$. This algorithm is performed by a certifier. After the algorithm is performed, the certifier publishes $params$ and keeps msk secret.

- (ii) *UserKeyGen*($params$): on input the public parameters $params$, this algorithm generates a private key and public key pair (SK_U, PK_U) for a user with identity id_U .

- (iii) *CertGen*($params, msk, id_U, PK_U$): on input the public parameters $params$, the master key msk , a user's identity id_U , and public key PK_U , this algorithm generates a certificate $Cert_U$. This algorithm is performed by a certifier. After this algorithm is performed, the certifier sends the certificate $Cert_U$ to the user id_U via an open channel.

- (iv) *Signcrypt*($params, M, id_S, PK_S, SK_S, Cert_S, id_R, PK_R$): on input the public parameters $params$, a sender's identity id_S , public key PK_S , private key SK_S and certificate $Cert_S$, a receiver's identity id_R , and public key PK_R , this algorithm generates a ciphertext σ .

- (v) *Designcrypt*($params, \sigma, id_R, PK_R, SK_R, Cert_R, id_S, PK_S$): on input the public parameters $params$, a ciphertext σ , the receiver's identity id_R , public key PK_R , private key SK_R and certificate $Cert_R$, the sender's identity id_S , and public key PK_S , this algorithm outputs either a plaintext M or a special symbol \perp indicating a designcrypt failure.

As introduced in [3], the adversaries against a certificate-based cryptographic scheme should be divided into two types: Type I and Type II. Type I adversary (denoted by A_I) models an uncertified user while Type II adversary (denoted by A_{II}) models an honest-but-curious certifier who is equipped with the master key. In order to capture public key replacement attack, the Type I adversary A_I in our CBSC security model is allowed to replace any user's public key. Note that the Type II adversary A_{II} should not be allowed to make public key replacement attacks; otherwise, it may trivially break the security of a CBSC scheme using a man-in-the-middle attack.

A CBSC scheme should satisfy both confidentiality (i.e., indistinguishability against adaptive chosen-ciphertext attacks (IND-CBSC-CCA2)) and unforgeability (i.e., existential unforgeability against adaptive chosen-messages attacks (EUF-CBSC-CMA)).

The confidentiality security of a CBSC scheme is defined via the following two games: "IND-CBSC-CCA2 Game-I" and "IND-CBSC-CCA2 Game-II," in which a Type I adversary A_I and a Type II adversary A_{II} interact with a challenger, respectively.

IND-CBSC-CCA2 Game-I. This game is played between A_I and a challenger.

Setup. The challenger runs the algorithm *Setup*(k) to generate msk and $params$. It then returns $params$ to A_I and keeps msk to itself.

Phase 1. In this phase, A_I makes requests to the following oracles adaptively.

- (i) $O^{CreateUser}$: on input an identity id_U , if id_U has already been created; the challenger outputs the current public key PK_U associated with id_U . Otherwise, it performs the algorithm $UserKeyGen(params)$ to generate a private/public key pair (SK_U, PK_U) , inserts (id_U, PK_U, SK_U) into a list, and outputs PK_U . In this case, id_U is said to be created. We assume that other oracles defined below only respond to an identity which has been created.
- (ii) $O^{ReplacePublicKey}$: on input an identity id_U and a value PK'_U , the challenger replaces the current public key of the identity id_U with PK'_U . Note that the current value of a user's public key is used by the challenger in any computations or responses to A_I 's requests. This oracle models the ability of a Type I adversary to convince a legitimate user to use a false public key and thus enables our security model to capture the public key replacement attacks attempted by the Type I adversary A_I .
- (iii) $O^{GenerateCertificate}$: on input an identity id_U , the challenger responds with a certificate $Cert_U$ by running the algorithm $CertGen(params, msk, id_U, PK_U)$.
- (iv) $O^{ExtractPrivateKey}$: on input an identity id_U , the challenger responds with a private key SK_U . Here, A_I is disallowed to query this oracle on any identity for which the public key has been replaced. This restriction is imposed due to the fact that it is unreasonable to expect the challenger to be able to provide a private key of a user for which it does not know the private key.
- (v) $O^{Signcrypt}$: on input a message M , a sender's identity id_S , and a receiver's identity id_R , the challenger responds with $\sigma = Signcrypt(params, M, id_S, PK_S, SK_S, Cert_S, id_R, PK_R)$. Note that it is possible that the challenger is not aware of the sender's private key if the associated public key has been replaced. In this case, we require A_I to provide it. In addition, we do not consider attacks targeting ciphertexts where the identities of the sender and receiver are the same. So, we disallow queries where $id_S = id_R$.
- (vi) $O^{Decrypt}$: on input a ciphertext σ , a sender's identity id_S , and a receiver's identity id_R , the challenger responds with the result of $Decrypt(params, \sigma, id_R, PK_R, SK_R, Cert_R, id_S, PK_S)$. Note that it is possible that the challenger is not aware of the receiver's private key if the associated public key has been replaced. In this case, we require A_I to provide it. Again, we disallow queries where $id_S = id_R$.

Challenge. Once A_I decides that Phase 1 is over, it outputs two equal-length messages (M_0, M_1) and two distinct identities (id_S^*, id_R^*) . The challenger picks a random bit b , computes

$$\sigma^* = Signcrypt(params, M_b, id_S^*, PK_S^*, SK_S^*, Cert_S^*, id_R^*, PK_R^*), \text{ and returns } \sigma^* \text{ as the challenge ciphertext to } A_I.$$

Phase 2. In this phase, A_I continues to issues queries as in Phase 1.

Guess. Finally, A_I outputs a guess $b' \in \{0, 1\}$. We say that A_I wins the game if $b = b'$ and the following conditions are simultaneously satisfied: (1) A_I cannot query $O^{GenerateCertificate}$ on the identity id_R^* at any point; (2) A_I cannot query $O^{ExtractPrivateKey}$ on an identity if the corresponding public key has been replaced; (3) in Phase 2, A_I cannot query $O^{Decrypt}$ on $(\sigma^*, id_S^*, id_R^*)$ unless the public key of the sender id_S^* or that of the receiver id_R^* has been replaced after the challenge was issued. We define A_I 's advantage in this game to be $2|\Pr\{b = b'\} - 1/2|$.

IND-CBSC-CCA2 Game-II. This game is played between A_{II} and a challenger.

Setup. The challenger runs the algorithm $Setup(k)$ to generate msk and $params$. It then returns $params$ and msk to A_{II} .

Phase 1. In this phase, A_{II} adaptively asks a polynomial bounded number of queries as in *IND-CBSC-CCA2 Game-I*. The only restriction is that A_{II} cannot replace public keys of any users. In addition, A_{II} need not make any queries to $O^{GenerateCertificate}$ since it can compute the certificates for any identities by itself with the master key msk .

Challenge. Once A_{II} decides that Phase 1 is over, it outputs two equal-length messages (M_0, M_1) and two distinct identities (id_S^*, id_R^*) . The challenger picks a random bit b , computes $\sigma^* = Signcrypt(params, M_b, id_S^*, PK_S^*, SK_S^*, Cert_S^*, id_R^*, PK_R^*)$, and returns σ^* as the challenge ciphertext to A_{II} .

Phase 2. In this phase, A_{II} continues to issue queries as in Phase 1.

Guess. Finally, A_{II} outputs a guess $b' \in \{0, 1\}$. We say that A_{II} wins the game if $b = b'$ and the following two conditions are both satisfied: (1) A_{II} cannot query $O^{ExtractPrivateKey}$ on the identity id_R^* at any point; (2) A_{II} cannot query $O^{Decrypt}$ on $(\sigma^*, id_S^*, id_R^*)$ in Phase 2. We define A_{II} 's advantage in this game to be $2|\Pr\{b = b'\} - 1/2|$.

Definition 5. A CBSC scheme is said to be IND-CBSC-CCA2 secure if no probabilistic polynomial time (PPT) adversary has nonnegligible advantage in the above two games.

Remark 6. The oracle $O^{ReplacePublicKey}$ defined in the game *IND-CBSC-CCA2 Game-I* models the ability of a Type I adversary to convince a legitimate user to use a false public key. It enables our security model to capture the public key replacement attacks attempted by the Type I adversary A_I .

Remark 7. The adversary in the above definition of message confidentiality is allowed to be challenged on a ciphertext generated using a corrupted sender's private key and

certificate. This condition corresponds to the stringent requirement of insider security for confidentiality of signcryption [23]. This means that our security model ensures that the confidentiality of signcryption is preserved even if a sender's private key is corrupted.

The unforgeability security of a CBSC scheme is defined via the following two games: "EUF-CBSC-CMA Game-I" and "EUF-CBSC-CMA Game-II," in which a Type I adversary A_I and a Type II adversary A_{II} interact with a challenger, respectively.

EUF-CBSC-CMA Game-I. This game is played between A_I and a challenger.

Setup. The challenger runs the algorithm $Setup(k)$ to generate msk and $params$. It then returns $params$ to A_I and keeps msk to itself.

Query. In this phase, A_I can adaptively ask a polynomial bounded number of queries as in the game *IND-CBSC-CCA2 Game-I*.

Forge. Finally, A_I outputs a forgery $(\sigma^*, id_S^*, id_R^*)$. We say that A_I wins the game if the result of $Designcrypt(params, \sigma^*, id_R^*, PK_R^*, SK_R^*, Cert_R^*, id_S^*, PK_S^*)$ is not the \perp symbol and the following conditions are simultaneously satisfied: (1) A_I cannot query $O^{GenerateCertificate}$ on the identity id_S^* at any point; (2) A_I cannot query $O^{ExtractPrivateKey}$ on an identity if the corresponding public key has been replaced; (3) σ^* is not the output of any $O^{Signcryption}$ query on (M^*, id_S^*, id_R^*) , where M^* is a message. We define A_I 's advantage in this game to be the probability that it wins the game.

EUF-CBSC-CMA Game-II. This game is played between A_{II} and a challenger.

Setup. The challenger runs the algorithm $Setup(k)$ to generate msk and $params$. It then returns $params$ and msk to A_{II} .

Query. In this phase, A_{II} can adaptively ask a polynomial bounded number of queries as in the game *IND-CBSC-CCA2 Game-II*.

Forge. Finally, A_{II} outputs a forgery $(\sigma^*, id_S^*, id_R^*)$. We say that A_{II} wins the game if the result of $Designcrypt(params, \sigma^*, id_R^*, PK_R^*, SK_R^*, Cert_R^*, id_S^*, PK_S^*)$ is not the \perp symbol and the following conditions are simultaneously satisfied: (1) A_{II} cannot query $O^{ExtractPrivateKey}$ on the identity id_S^* ; (2) σ^* is not the output of any $O^{Signcryption}$ query on (M^*, id_S^*, id_R^*) , where M^* is a message. We define A_{II} 's advantage in this game to be the probability that it wins the game.

Definition 8. A CBSC scheme is said to be EUF-CBSC-CMA secure if no PPT adversary has nonnegligible advantage in the above two games.

Remark 9. The adversary in the above definition of signature unforgeability may output a ciphertext generated using a

corrupted receiver's private key and certificate. Again, this condition corresponds to the stringent requirement of insider security for unforgeability of signcryption [23]. Hence, our security model also ensures that the unforgeability of signcryption is preserved even if a receiver's private key is corrupted.

4. Cryptanalysis of Luo et al.'s CBSC Scheme

In this section, we give the review and attack of Luo et al.'s CBSC scheme [20].

4.1. Review of Luo et al.'s CBSC Scheme. Luo et al.'s CBSC scheme consists of the following six algorithms.

- (i) *Setup:* given a security parameter k , the certifier performs as follows: generate two cyclic groups G and G_T of prime order p such that there exists a bilinear pairing map $e : G \times G \rightarrow G_T$; select a random element $s \in Z_p^*$ and a random generator $P \in G$, and compute $P_{pub} = sP$; select four hash functions $H_1 : \{0, 1\}^n \times G \rightarrow G$, $H_2 : \{0, 1\}^n \times G \times G \rightarrow G$, $H_3 : G \times G \times \{0, 1\}^n \rightarrow Z_p^*$, and $H_4 : G_T \rightarrow \{0, 1\}^n$; set the public parameters $params = \{p, G, G_T, e, n, P, P_{pub}, H_1, H_2, H_3, H_4\}$ and the master key $msk = s$.
- (ii) *UserKeyGen:* given $params$, a user with identity $id_U \in \{0, 1\}^n$ chooses a random $x_U \in Z_p^*$ as his private key SK_U and then computes his public key $PK_U = x_U P$.
- (iii) *CertGen:* to generate a certificate for the user with identity id_U and public key PK_U , the certifier computes $Q_U = H_1(id_U, PK_U)$ and outputs the certificate $Cert_U = sQ_U$.
- (iv) *Sender Signcrypt:* to send a message $M \in \{0, 1\}^n$ to the receiver id_R , the sender id_S does the following: randomly choose $r \in Z_p^*$ and compute $R = rP$ and $T = H_2(id_S, PK_S, R)$; compute $h = H_3(R, S, M)$ and $V = r^{-1}(Cert_S + SK_S \cdot T + h \cdot P_{pub})$; compute $W = e(PK_S, PK_R)^r$ and then $C = M \oplus H_4(W)$; set the ciphertext $\sigma = (C, R, V)$.
- (v) *Receiver Decrypt:* when receiving a ciphertext $\sigma = (C, R, V)$ from the sender id_S , the receiver id_R does the following: compute $M = C \oplus H_4(W)$ where $W = e(R, SK_R \cdot PK_S)$; forward the message M and signature (R, V) to the algorithm *Receiver Verify*.
- (vi) *Receiver Verify:* to verify the sender id_S 's signature (R, V) on the message M , the receiver id_R does the following: compute $S = H_2(id_S, PK_S, R)$ and $h = H_3(R, S, M)$; check whether $e(R, V) = e(P_{pub}, Q_S)e(PK_S, S)e(P, P_{pub})^h$. If the check holds, output M ; otherwise, output \perp .

4.2. Attack on Luo et al.'s CBSC Scheme. A Type I adversary who is capable of replacing any user's public key can forge a valid signcryption on any message M from id_S to id_R by performing the following steps.

- (1) Replace the sender id_S 's public key with $PK'_S = x'_S P_{pub}$, where x'_S is a random value chosen from Z_p^* .
- (2) Choose a random value $r' \in Z_p^*$ and compute $R' = r' P_{pub}$ and $T' = H_2(id_S, PK'_S, R')$.
- (3) Choose a random message $M \in \{0, 1\}^n$ and compute $V' = r'^{-1}(Q_S + x'_S T' + h' P)$, where $Q_S = H_1(id_S, PK'_S)$ and $h' = H_3(R', T', M)$.
- (4) Randomly choose $C' \in \{0, 1\}^n$ and set $\sigma' = (C', R', V')$ as the signcryption of the message M . Note that if the adversary has corrupted the receiver id_R 's private key SK_R , it can compute $C' = M \oplus H_4(W')$, where $W' = e(R', SK_R, PK'_S)$.

The ciphertext $\sigma' = (C', R', V')$ passes the verification test as shown below:

$$\begin{aligned}
 & e(P_{pub}, Q_S) e(PK'_S, T') e(P, P_{pub})^{h'} \\
 &= e(P_{pub}, Q_S + x'_S T' + h' P) \\
 &= e(r' P_{pub}, r'^{-1} (Q_S + x'_S T' + h' P)) \\
 &= e(R', V').
 \end{aligned} \tag{1}$$

This proves that the forged signcryption is valid.

Note that Luo et al.'s scheme also does not resist insider attacks since the adversary can forge a valid signcryption using the corrupted receiver id_R 's private key in the step (4).

5. Our Proposed CBSC Scheme

5.1. Description of the Scheme. Our CBSC scheme is constructed from the certificate-based encryption scheme proposed by Lu et al. [8]. It consists of the following five algorithms.

- (i) *Setup*(k): given a security parameter k , the certifier performs the following: generate two cyclic groups G and G_T of a k -bit prime order p such that there exists a bilinear pairing map $e : G \times G \rightarrow G_T$; choose two random generators $P, Q \in G$ and compute $g = e(P, Q)$; choose a random element $\alpha \in Z_p^*$ and set $P_{pub} = \alpha P$; select three hash functions $H_1 : \{0, 1\}^* \times G_T \rightarrow Z_p^*$, $H_2 : G_T \times G_T \rightarrow \{0, 1\}^n$ and $H_3 : \{0, 1\}^* \rightarrow Z_p^*$, where n is the bit-length of the message to be signcrypted; set the public parameters $params = \{p, G, G_T, e, n, P, Q, P_{pub}, g, H_1, H_2, H_3\}$ and the master key $msk = \alpha$.
- (ii) *UserKeyGen*($params$): given $params$, a user with identity $id_U \in \{0, 1\}^*$ chooses a random $x_U \in Z_p^*$ as his private key SK_U and then computes his public key $PK_U = g^{x_U}$.
- (iii) *CertGen*($params, msk, id_U, PK_U$): to generate a certificate for a user with identity id_U and public key PK_U , the certifier computes $Cert_U = (H_1(id_U, PK_U) + \alpha)^{-1} Q$. The user id_U can check the

validness of the certificate $Cert_U$ by verifying whether $e(H_1(id_U, PK_U)P + P_{pub}, Cert_U) = g$.

- (iv) *Signcrypt*($params, M, id_S, PK_S, SK_S, Cert_S, id_R, PK_R$): to send a message $M \in \{0, 1\}^n$ to the receiver id_R , the sender id_S does the following: randomly choose $r \in Z_p^*$ and compute $R_1 = g^r$ and $R_2 = (PK_R)^r$; compute $U = r(H_1(id_R, PK_R)P + P_{pub})$ and $C = M \oplus H_2(R_1, R_2)$; compute $V = (h \cdot SK_S + r) \cdot Cert_S$, where $h = H_3(M, U, R_1, R_2, id_S, PK_S, id_R, PK_R)$; set the ciphertext $\sigma = (C, U, V)$.
- (v) *Designcrypt*($params, \sigma, id_R, PK_R, SK_R, Cert_R, id_S, PK_S$): to designcrypt a ciphertext $\sigma = (C, U, V)$ from the sender id_S , the receiver id_R does the following: compute $R_1 = e(U, Cert_R)$ and $R_2 = e(U, Cert_R)^{SK_R}$; compute $M = C \oplus H_2(R_1, R_2)$ and then check whether $e(H_1(id_S, PK_S)P + P_{pub}, V)(PK_S)^{-h} = R_1$, where $h = H_3(M, U, R_1, R_2, id_S, PK_S, id_R, PK_R)$. If the check holds, output M ; otherwise, output \perp .

The consistency of our scheme can be easily verified by the following equalities:

- (1) $e(U, Cert_R) = e(r(H_1(id_R, PK_R)P + P_{pub}), (H_1(id_R, PK_R) + \alpha)^{-1} Q) = e(P, Q)^r = g^r$;
- (2) $e(U, Cert_R)^{SK_R} = (g^r)^{SK_R} = (PK_R)^r$;
- (3)

$$\begin{aligned}
 & e(H_1(id_S, PK_S)P + P_{pub}, V) (PK_S)^{-h} \\
 &= e((H_1(id_S, PK_S) + \alpha)^{-1} P, (h \cdot SK_S + r) \cdot Cert_S) \\
 &\quad \cdot (PK_S)^{-h} \\
 &= e(P, (h \cdot SK_S + r) Q) \cdot (PK_S)^{-h} \\
 &= e(P, Q)^r = R_1.
 \end{aligned} \tag{2}$$

5.2. Security Proof

Theorem 10. *The CBSC scheme above is IND-CBSC-CCA2 secure under the hardness of the q -mBDHI and BDH problems in the random oracle model.*

This theorem can be proved by combining the following two lemmas.

Lemma 11. *If a Type I adversary A_1 has advantage ϵ against our CBSC scheme when asking at most q_{cu} to $O^{CreateUser}$ queries, q_{sc} queries to $O^{Signcryption}$, q_{dsc} queries to $O^{Designcryption}$, and q_i queries to random oracles $H_1 \sim H_3$, then there exists an algorithm B to solve the $(q_1 - 1)$ -mBDHI problem with advantage*

$$\begin{aligned}
 \epsilon' &\geq \frac{\epsilon}{q_1 (q_2 + 2q_3 + 2q_{sc})} \left(1 - q_{sc} \frac{q_2 + 2q_3 + 2q_{sc}}{2^k} \right) \\
 &\quad \times \left(1 - \frac{q_{dsc}}{2^k} \right).
 \end{aligned} \tag{3}$$

Proof. Assume that B is given a random q -mBDHI instance $(P, \alpha P, (\omega_1 + \alpha)^{-1}P, \dots, (\omega_q + \alpha)^{-1}P, \omega_1, \dots, \omega_q)$, where $q = q_1 - 1$. B interacts with A_I as follows.

In the setup phase, B randomly chooses $t \in Z_p^*$ and sets $P_{\text{pub}} = \alpha P$, $Q = tP$, and $g = e(P, Q)$. Furthermore, it randomly chooses a value $\omega^* \in Z_p^*$ such that $\omega^* \notin \{\omega_1, \dots, \omega_q\}$ and an index $\theta \in [1, q_1]$. Then, B starts *IND-CBSC-CCA2 Game-I* by supplying A_I with $params = \{p, G, G_T, e, n, P, Q, P_{\text{pub}}, g, H_1, H_2, H_3\}$, where $H_1 \sim H_3$ are random oracles controlled by B. A_I can make queries on these random oracles at any time during the game. Note that the corresponding master key is $msk = \alpha$ which is unknown to B.

Now, B starts to respond to various queries as follows:

H_1 Queries. We assume that q_1 queries to H_1 are distinct. B maintains a list $H_1\text{List}$ of tuples $\langle id_i, PK_i, h_{1,i}, Cert_i \rangle$. On input $\langle id_i, PK_i \rangle$, B does the following.

- (1) If $\langle id_i, PK_i \rangle$ already appears on $H_1\text{List}$ in a tuple $\langle id_i, PK_i, h_{1,i}, Cert_i \rangle$, then B returns $h_{1,i}$ to A_I .
- (2) Else if the query is on the θ th distinct (id_θ, PK_θ) , then B inserts $\langle id_\theta, PK_\theta, \omega^*, \perp \rangle$ into $H_1\text{List}$ and returns $h_{1,\theta} = \omega^*$ to A_I . Note that the certificate for the identity id_θ is $Cert_\theta = t(\omega^* + \alpha)^{-1}P$ which is unknown to B.
- (3) Else B sets $h_{1,i}$ to be ω_j ($j \in [1, q]$) which has not been used and computes $Cert_i = t(\omega_j + \alpha)^{-1}P$. It then inserts $\langle id_i, PK_i, h_{1,i}, Cert_i \rangle$ into $H_1\text{List}$ and returns $h_{1,i}$.

H_2 Queries. B maintains a list $H_2\text{List}$ of tuples $\langle R_1, R_2, h_2 \rangle$. On input $\langle R_1, R_2 \rangle$, B does the following.

- (1) If $\langle R_1, R_2 \rangle$ already appears on $H_2\text{List}$ in a tuple $\langle R_1, R_2, h_2 \rangle$, B returns h_2 to A_I .
- (2) Otherwise, it returns a random $h_2 \in \{0, 1\}^n$ and inserts $\langle R_1, R_2, h_2 \rangle$ into $H_2\text{List}$.

H_3 Queries. B maintains a list $H_3\text{List}$ of tuples $\langle M, U, R_1, R_2, id_S, PK_S, id_R, PK_R, h_3, C \rangle$. On input $\langle M, U, R_1, R_2, id_S, PK_S, id_R, PK_R \rangle$, B does the following.

- (1) If $\langle M, U, R_1, R_2, id_S, PK_S, id_R, PK_R \rangle$ already appears on $H_3\text{List}$ in a tuple $\langle M, U, R_1, R_2, id_S, PK_S, id_R, PK_R, h_3, C \rangle$, B returns h_3 to A_I .
- (2) Otherwise, it returns a random $h_3 \in Z_p^*$ to A_I . To anticipate possible subsequent queries to $O^{\text{Designcrypt}}$, it additionally simulates the random oracle H_2 on its own to obtain $h_2 = H_2(R_1, R_2)$ and then inserts $\langle M, U, R_1, R_2, id_S, PK_S, id_R, PK_R, h_3, C = M \oplus h_2 \rangle$ into $H_3\text{List}$.

$O^{\text{CreateUser}}$ Queries. B maintains a list KeyList of tuples $\langle id_i, PK_i, SK_i, \text{flag}_i \rangle$ which is initially empty. On input $\langle id_i \rangle$, B does the following.

- (1) If id_i already appears on KeyList in a tuple $\langle id_i, PK_i, SK_i, \text{flag}_i \rangle$, B returns PK_i to A_I directly.
- (2) Otherwise, B randomly chooses $x_i \in Z_p^*$ as the private key SK_i for the identity id_i and computes the corresponding public key as $PK_i = g^{x_i}$. It then inserts $\langle id_i, PK_i, SK_i, 0 \rangle$ into KeyList and returns PK_i to A_I .

$O^{\text{ReplacePublicKey}}$ Queries. On input $\langle id_i, PK'_i \rangle$, B searches id_i in KeyList to find a tuple $\langle id_i, PK_i, SK_i, \text{flag}_i \rangle$ and updates the tuple with $\langle id_i, PK'_i, SK_i, 1 \rangle$.

$O^{\text{ExtractPrivateKey}}$ Queries. On input $\langle id_i \rangle$, B searches id_i in KeyList to find a tuple $\langle id_i, PK_i, SK_i, \text{flag}_i \rangle$. If $\text{flag}_i = 0$, it returns SK_i to A_I ; otherwise, it rejects this query.

$O^{\text{GenerateCertificate}}$ Queries. On input $\langle id_i \rangle$, B does the following.

- (1) If $\langle id_i, PK_i \rangle = \langle id_\theta, PK_\theta \rangle$, then B aborts.
- (2) Otherwise, B searches id_i in $H_1\text{List}$ to find a tuple $\langle id_i, PK_i, h_{1,i}, Cert_i \rangle$ and then returns $Cert_i$ to A_I . If $H_1\text{List}$ does not contain such a tuple, B queries H_1 on $\langle id_i, PK_i \rangle$ first.

$O^{\text{Signcrypt}}$ Queries. On input $\langle M, id_S, id_R \rangle$, B performs as follows.

- (1) If $\langle id_S, PK_S \rangle \neq \langle id_\theta, PK_\theta \rangle$, B can answer the query according to the specification of the algorithm *Signcrypt* since it knows the sender id_S 's private key and certificate.
- (2) Otherwise, B randomly chooses $r, h_3 \in Z_p^*$, $h_2 \in \{0, 1\}^n$ and sets $U = r(H_1(id_\theta, PK_\theta)P + P_{\text{pub}}) - h_3SK_\theta(H_1(id_R, PK_R)P + P_{\text{pub}})$, $V = rCert_R$, $C = M \oplus h_2$, $R_1 = e(U, Cert_R)$, and $R_2 = e(U, Cert_R)^{SK_R}$, respectively. It is easy to verify that $e(H_1(id_\theta, PK_\theta)P + P_{\text{pub}}, V) \cdot (PK_\theta)^{-h_3} = e(U, Cert_R)$. Then, B inserts $\langle R_1, R_2, h_2 \rangle$ and $\langle M, U, R_1, R_2, id_\theta, PK_\theta, id_R, PK_R, h_3, C \rangle$ into $H_2\text{List}$ and $H_3\text{List}$ respectively, and returns the ciphertext $\sigma = (C, U, V)$ to A_I . Note that B fails if $H_2\text{List}$ or $H_3\text{List}$ is already defined in the corresponding value, but this only happens with probability smaller than $(q_2 + 2q_3 + 2q_{sc})/2^k$.

$O^{\text{Designcrypt}}$ Queries. On input $\langle \sigma = (C, U, V), id_S, id_R \rangle$, B does the following.

- (1) If $\langle id_R, PK_R \rangle \neq \langle id_\theta, PK_\theta \rangle$, B can answer the query according to the specification of the algorithm *Designcrypt* since it knows the receiver id_R 's private key and certificate.
- (2) Otherwise, B searches in $H_3\text{List}$ for all tuples of the form $\langle M, U, R_1, R_2, id_S, PK_S, id_\theta, PK_\theta, h_3, C \rangle$. If no such tuple is found, then σ is rejected. Otherwise,

each one of them is further examined. For a tuple $\langle M, U, R_1, R_2, id_S, PK_S, id_\theta, PK_\theta, h_3, C \rangle$, B first checks whether $e(H_1(id_S, PK_S)P + P_{pub}, V) \cdot (PK_S)^{-h_3} = R_1$. If the tuple passes the verification, then B returns M in this tuple to A_I . If no such tuple is found, σ is rejected. Note that a valid ciphertext is rejected with probability smaller than $q_{dsc}/2^k$ across the whole game.

In the challenge phase, A_I outputs $(M_0, M_1, id_S^*, id_R^*)$, on which it wants to be challenged. If $(id_R^*, PK_R^*) \neq (id_\theta, PK_\theta)$, then B aborts. Otherwise, B randomly chooses $C^* \in \{0, 1\}^n$, $r^* \in Z_p^*$, and $V^* \in G$, computes $U^* = r^*P$, and returns $\sigma^* = (C^*, U^*, V^*)$ to A_I as the challenge ciphertext. Observe that the decryption of C^* is $C^* \oplus H_2(e(U^*, Cert_\theta), e(U^*, Cert_\theta)^{SK_\theta})$.

In the guess phase, A_I outputs a bit which is ignored by B. Note that A_I cannot recognize that σ^* is not a valid ciphertext unless it queries H_2 on $(e(U^*, Cert_\theta), e(U^*, Cert_\theta)^{SK_\theta})$ or H_3 on $(M_b, U^*, (e(U^*, Cert_\theta), e(U^*, Cert_\theta)^{SK_\theta}), id_S^*, PK_S^*, id_\theta, PK_\theta)$, where $b \in \{0, 1\}$. Standard arguments can show that a successful A_I is very likely to query H_2 on $(e(U^*, Cert_\theta), e(U^*, Cert_\theta)^{SK_\theta})$ or H_3 on $(M_b, U^*, (e(U^*, Cert_\theta), e(U^*, Cert_\theta)^{SK_\theta}), id_S^*, PK_S^*, id_\theta, PK_\theta)$ if the simulation is indistinguishable from a real attack environment. To produce a result, B picks a random tuple $\langle R_1, R_2, h_2 \rangle$ or $\langle M, U, R_1, R_2, id_S, PK_S, id_R, PK_R, h_3, C \rangle$ from H_2List or H_3List . With probability $1/(q_2 + 2q_3 + 2q_{sc})$ (as H_2List, H_3List contain at most $q_2 + q_3 + q_{sc}, q_3 + q_{sc}$ tuples, resp.), the chosen tuple will contain the value $R_1 = e(U^*, Cert_\theta)$. Because $e(U^*, Cert_\theta) = e(r^*P, t(\omega^* + \alpha)^{-1}P) = e(P, P)^{tr^*(\omega^* + \alpha)^{-1}}$, B returns $T = R_1^{(tr^*)^{-1}}$ as the solution to the given q -mBDHI problem.

We now derive B's advantage in solving the q -mBDHI problem. From the above construction, the simulation fails if any of the following events occurs: (1) E_1 : in the challenge phase, B aborts because $(id_R^*, PK_R^*) \neq (id_\theta, PK_\theta)$; (2) E_2 : A_I makes an $O^{GenerateCertificate}$ query on (id_θ, PK_θ) ; (3) E_3 : B aborts in answer one of A_I 's $O^{Signcrypt}$ queries because of a collision on H_2 or H_3 ; (4) E_4 : B rejects a valid ciphertext at some point of the game.

We clearly have that $\Pr[\neg E_1] = 1/q_1$ and $\neg E_1$ implies $\neg E_2$. We also already observed that $\Pr[E_3] \leq (q_2 + 2q_3 + 2q_{sc})/2^k$ and $\Pr[E_4] \leq q_{dsc}/2^k$. Thus, we have that

$$\Pr[\neg E_1 \wedge \neg E_2 \wedge \neg E_3 \wedge \neg E_4] \geq \frac{1}{q_1} \left(1 - q_{sc} \frac{q_2 + 2q_3 + 2q_{sc}}{2^k}\right) \times \left(1 - \frac{q_{dsc}}{2^k}\right). \quad (4)$$

Since B selects the correct tuple from H_2List or H_3List with probability $1/(q_2 + 2q_3 + 2q_{sc})$, we obtain the announced bound on B's advantage in solving the q -mBDHI problem. \square

Lemma 12. *If a Type II adversary A_{II} has advantage ε against our CBSC scheme when asking at most q_{cu} queries to $O^{CreateUser}$,*

q_{sc} queries to $O^{Signcrypt}$, q_{dsc} queries to $O^{Decrypt}$, and q_i queries to random oracles $H_1 \sim H_3$, then there exists an algorithm B to solve the BDH problem with advantage

$$\varepsilon' \geq \frac{\varepsilon}{q_{cu}(q_2 + 2q_3 + 2q_{sc})} \left(1 - q_{sc} \frac{q_2 + 2q_3 + 2q_{sc}}{2^k}\right) \times \left(1 - \frac{q_{dsc}}{2^k}\right). \quad (5)$$

Proof. Assume that B is given a BDH instance (P, aP, bP, cP) , where a, b, c are three random elements from Z_p^* . B interacts with A_{II} as follows.

In the setup phase, B randomly chooses $\alpha \in Z_p^*$, sets $Q = \alpha P$, and computes $P_{pub} = \alpha P$ and $g = e(P, Q)$. Furthermore, it randomly chooses an index θ with $1 \leq \theta \leq q_{cu}$. Then, B starts *IND-CBSC-CCA2 Game-II* by supplying A_{II} with $msk = \alpha$ and $params = \{p, G, G_T, e, n, P, Q, P_{pub}, g, H_1, H_2, H_3\}$, where $H_1 \sim H_3$ are random oracles controlled by B. A_{II} can make queries on these random oracles at any time during the game.

Now, B starts to respond various queries as follows.

H_1 Queries. B maintains a list H_1List of tuples $\langle id_i, PK_i, h_{1,i} \rangle$. On input (id_i, PK_i) , B does the following: if (id_i, PK_i) already appears on H_1List in a tuple $\langle id_i, PK_i, h_{1,i} \rangle$, then B returns $h_{1,i}$ to A_{II} ; otherwise, it returns a random $h_{1,i} \in Z_p^*$ and inserts $\langle id_i, PK_i, h_{1,i} \rangle$ into H_1List .

H_2 Queries. B responds as in the proof of Lemma 11.

H_3 Queries. B responds as in the proof of Lemma 11.

$O^{CreateUser}$ Queries. B maintains a list $KeyList$ of tuples $\langle id_i, PK_i, SK_i \rangle$. On input (id_i) , B does the following: (1) if id_i already appears on $KeyList$ in a tuple $\langle id_i, PK_i, SK_i \rangle$, B returns PK_i to A_{II} . (2) Else if $id_i = id_\theta$, B returns $PK_\theta = e(bP, Q) = e(bP, \alpha P)$ to A_{II} and inserts $\langle id_\theta, PK_\theta, \perp \rangle$ into $KeyList$. Note that the private key for the identity id_θ is b which is unknown to B. (3) Else B randomly chooses $x_i \in Z_p^*$ as the private key SK_i for the identity id_i and computes the corresponding public key as $PK_i = g^{x_i}$. It then inserts $\langle id_i, PK_i, SK_i \rangle$ into $KeyList$ and returns PK_i to A_{II} .

$O^{ExtractPrivateKey}$ Queries. On receiving such a query on id_i , B does the following: if $id_i = id_\theta$, then B aborts; otherwise, B searches id_i in $KeyList$ to find the tuple $\langle id_i, PK_i, SK_i \rangle$ and returns SK_i to A_{II} .

$O^{Signcrypt}$ Queries. On input (M, id_S, id_R) , B does the following: if $id_S \neq id_\theta$, B can answer the query according to the specification of the *Signcrypt* algorithm since it knows the sender id_S 's private key and certificate. Otherwise, B randomly chooses $r, h_3 \in Z_p^*$, $h_2 \in \{0, 1\}^n$ and computes $U = r(H_1(id_\theta, PK_\theta)P + P_{pub}) - h_3(H_1(id_R, PK_R)bP + \alpha P)$, $V = rCert_R$, $C = M \oplus h_2$, $R_1 = e(U, Cert_R)$, and $R_2 = e(U, Cert_R)^{SK_R}$, respectively. It is easy to verify that $e(H_1(id_\theta, PK_\theta)P + P_{pub}, V) \cdot (PK_\theta)^{-h_3} = e(U, Cert_R)$. It then inserts $\langle R_1, R_2, h_2 \rangle$ and $\langle M, U, R_1, R_2, id_\theta, PK_\theta, id_R, PK_R, h_3, C \rangle$

into H_2List and H_3List respectively, and returns the ciphertext $\sigma = (C, U, V)$ to A_{II} . Note that B fails if H_2List or H_3List is already defined in the corresponding value, but this only happens with probability smaller than $(q_2 + 2q_3 + 2q_{sc})/2^k$.

$O^{Designcrypt}$ Queries. B responds as in the proof of Lemma 11.

In the challenge phase, A_{II} outputs $(M_0, M_1, id_S^*, id_R^*)$, on which it wants to be challenged. If $id_R^* \neq id_\theta$, then B aborts. Otherwise, B randomly chooses $C^* \in \{0, 1\}^n$, $V^* \in G$, computes $U^* = (H_1(id_\theta, PK_\theta) + \alpha)cP$, and returns $\sigma^* = (C^*, U^*, V^*)$ to A_{II} as the challenge ciphertext. Observe that the decryption of C^* is $C^* \oplus H_2(e(U^*, Cert_\theta), e(U^*, Cert_\theta)^{SK_\theta})$.

In the guess phase, A_{II} outputs a bit, which is ignored by B. Note that A_{II} cannot recognize that σ^* is not a valid ciphertext unless it queries H_2 on $(e(U^*, Cert_\theta), e(U^*, Cert_\theta)^{SK_\theta})$ or H_3 on $(M_\beta, U^*, e(U^*, Cert_\theta), e(U^*, Cert_\theta)^{SK_\theta}, id_S^*, PK_S^*, id_\theta, PK_\theta)$, where $\beta \in \{0, 1\}$. Standard arguments can show that a successful A_{II} is very likely to query H_2 on $(e(U^*, Cert_\theta), e(U^*, Cert_\theta)^{SK_\theta})$ or H_3 on $(M_\beta, U^*, e(U^*, Cert_\theta), e(U^*, Cert_\theta)^{SK_\theta}, id_S^*, PK_S^*, id_\theta, PK_\theta)$ if the simulation is indistinguishable from a real attack environment. To produce a result, B picks a random tuple $\langle R_1, R_2, h_2 \rangle$ or $\langle M, U, R_1, R_2, id_S, PK_S, id_R, PK_R, h_3, C \rangle$ from H_2List or H_3List . With probability $1/(q_2 + 2q_3 + 2q_{sc})$ (as H_2List, H_3List contain at most $q_2 + q_3 + q_{sc}$, $q_3 + q_{sc}$ tuples, resp.), the chosen tuple will contain the right element $R_2 = e(U^*, Cert_\theta)^{SK_\theta} = e(P, P)^{abc}$. B then returns R_2 as the solution to the given BDH problem.

We now derive B's advantage in solving the BDH problem. From the above construction, the simulation fails if any of the following events occurs: (1) E_1 : in the challenge phase, B aborts because $id_R^* \neq id_\theta$; (2) E_2 : A_{II} makes an $O^{ExtractPrivateKey}$ query on id_θ ; (3) E_3 : B aborts in answer A_{II} 's $O^{Signcrypt}$ query because of a collision on H_2 or H_3 ; (4) E_4 : B rejects a valid ciphertext at some point of the game.

We clearly have that $\Pr[\neg E_1] = 1/q_{cu}$ and $\neg E_1$ implies $\neg E_2$. We also already observed that $\Pr[E_3] \leq (q_2 + 2q_3 + 2q_{sc})/2^k$ and $\Pr[E_4] \leq q_{dsc}/2^k$. Thus, we have that

$$\begin{aligned} & \Pr[\neg E_1 \wedge \neg E_2 \wedge \neg E_3 \wedge \neg E_4] \\ & \geq \frac{1}{q_{cu}} \left(1 - q_{sc} \frac{q_2 + 2q_3 + 2q_{sc}}{2^k}\right) \left(1 - \frac{q_{dsc}}{2^k}\right). \end{aligned} \quad (6)$$

Since B selects the correct tuple from H_2List or H_3List with probability $1/(q_2 + 2q_3 + 2q_{sc})$, we obtain the announced bound on B's advantage in solving the BDH problem. \square

Theorem 13. *The CBSC scheme above is EUF-CBSC-CMA secure under the hardness of the q-CAA and CDH problems in the random oracle model.*

This theorem can be proved by combining the following two lemmas.

Lemma 14. *If a Type I adversary A_I asks at most q_{cu} queries to $O^{CreateUser}$, q_{sc} queries to $O^{Signcrypt}$, q_{dsc} queries to $O^{Designcrypt}$, and q_i queries to random oracles $H_1 \sim H_3$ and produces a valid forgery with probability $\varepsilon \geq 10(q_{sc} + 1)(q_{sc} + q_3)/2^k$, then there exists an algorithm B to solve the $(q_1 - 1)$ -CAA problem with advantage $\varepsilon' \geq 1/(9q_1)$.*

Proof. Assume that B is given a q -CAA instance $(P, \alpha P, (\omega_1 + \alpha)^{-1}P, \dots, (\omega_q + \alpha)^{-1}P, \omega_1, \dots, \omega_q)$, where $q = q_1 - 1$. B interacts with A_I as follows.

In the setup phase, B randomly chooses $t \in Z_p^*$, sets $P_{pub} = \alpha P$, and computes $Q = tP$ and $g = e(P, Q)$. Furthermore, it randomly chooses a value $\omega^* \in Z_p^*$ such that $\omega^* \notin \{\omega_1, \dots, \omega_q\}$ and an index $\theta \in [1, q_1]$. Then, B starts *EUFCBSC-CMA Game-I* by supplying A_I with *params* = $\{p, G, G_T, e, n, P, Q, P_{pub}, g, H_1, H_2, H_3\}$, where $H_1 \sim H_3$ are random oracles controlled by B. Note that the corresponding master key is $msk = \alpha$ which is unknown to B.

In the query phase, B responds to various oracle queries as in the proof of Lemma 11.

Finally, in the forge phase A_I outputs a valid forgery $(\sigma^* = (C^*, U^*, V^*), id_S^*, id_R^*)$ with probability $\varepsilon \geq 10(q_{sc} + 1)(q_{sc} + q_3)/2^k$ [29]. If $(id_S^*, PK_S^*) \neq (id_\theta, PK_\theta)$, B aborts. Otherwise, having the knowledge of SK_R^* and $Cert_R^*$, B runs the algorithm *Designcrypt(params, $\sigma^*, id_S^*, PK_S^*, SK_R^*, Cert_R^*, id_\theta, PK_\theta$)* to obtain the message M^* and then simulates the random oracle H_3 on its own to obtain $h_3^* = H_3(M^*, U^*, e(U^*, Cert_R^*), e(U^*, Cert_R^*)^{SK_R^*}, id_\theta, PK_\theta, id_R^*, PK_R^*)$. Using the oracle replay technique [29], B replays A_I with the same random tape but with the different hash value $h_3^{*'} (\neq h_3^*)$ to generate one more valid ciphertext $\sigma^{*'} = (C^*, U^*, V^{*'})$ such that $V^{*'} \neq V^*$. Since $\sigma^* = (C^*, U^*, V^*)$ and $\sigma^{*'} = (C^*, U^*, V^{*'})$ are both valid ciphertexts for the same message M^* and the randomness r^* , we obtain the following relations:

$$\begin{aligned} V^* - V^{*'} &= (h_3^* SK_\theta + r^*) Cert_\theta - (h_3^{*'} SK_\theta + r^*) Cert_\theta \\ &= (h_3^* - h_3^{*'}) SK_\theta Cert_\theta. \end{aligned} \quad (7)$$

Because $Cert_\theta = t(\omega^* + \alpha)^{-1}P$, B can compute $(\omega^* + \alpha)^{-1}P = [t(h_3^* - h_3^{*'})SK_\theta]^{-1}(V^* - V^{*'})$ as the solution to the given q -CAA problem.

We now derive B's advantage in solving the q -CAA problem. From the above construction, the simulation fails after A_I outputs a valid forgery if any of the following events occurs: (1) E_1 : in the forge phase, B aborts because $(id_S^*, PK_S^*) \neq (id_\theta, PK_\theta)$; (2) E_2 : B fails in using the oracle replay technique to generate one more valid ciphertext.

Clearly, $\Pr[\neg E_1] = 1/q_1$. Moreover, from the forking lemma [29], we know that $\Pr[\neg E_2] \geq 1/9$. Thus, we have that if A_I produces a forgery, then B will succeed in solving the q -CAA problem with probability $\varepsilon' = \Pr[\neg E_1 \wedge \neg E_2] \geq 1/(9q_1)$. \square

Lemma 15. *If a Type II adversary A_{II} asks at most q_{cu} queries to $O^{CreateUser}$, q_{sc} queries to $O^{Signcrypt}$, q_{dsc} queries to $O^{Designcrypt}$, and q_i queries to random oracles $H_1 \sim H_3$ and*

TABLE 2: Performance of the CBSC schemes.

Schemes	Signcryption cost	Designcryption cost	Ciphertext overhead
Ours	$2e + 3m + 3h$	$2p + 2e + 1m + 3h$	$2 G $
[18]	$1p + 1e + 4m + 3h$	$3p + 1e + 1m + 3h$	$2 G $
[20]	$1p + 5m + 4h$	$4p + 2m + 3h$	$3 G + id $
[21]	$2p + 4e + 3m + 3h$	$3p + 4e + 3h$	$2 G + 2 Z_p $

produces a valid forgery with probability $\epsilon \geq 10(q_{sc} + 1)(q_{sc} + q_3)/2^k$, then there exists an algorithm B to solve the CDH problem with advantage $\epsilon' \geq 1/(9q_{cu})$.

Proof. Assume that B is given a random CDH instance (P, aP, bP) where a, b are two random elements from Z_p^* . B interacts with A_{II} as follows.

In the setup phase, B randomly chooses $\alpha \in Z_p^*$, sets $Q = aP$, and computes $P_{pub} = \alpha P$ and $g = e(P, Q)$. Furthermore, it randomly chooses an index θ with $1 \leq \theta \leq q_{cu}$. Then, B starts *EUF-CBSC-CMA Game-II* by supplying A_{II} with $msk = \alpha$ and $params = \{p, G, G_T, e, n, P, Q, P_{pub}, g, H_1, H_2, H_3\}$, where $H_1 \sim H_3$ are random oracles controlled by B .

In the query phase, B responds to various oracle queries as in the proof of Lemma 12.

Finally, in the forge phase A_I outputs a valid forgery $(\sigma^* = (C^*, U^*, V^*), id_S^*, id_R^*)$ with probability $\epsilon \geq 10(q_{sc} + 1)(q_{sc} + q_3)/2^k$ [29]. If $id_S^* \neq id_\theta$, then B aborts. Otherwise, having the knowledge of SK_R^* and $Cert_R^*$, B runs the algorithm *Designcrypt*($params, \sigma^*, id_R^*, PK_R^*, SK_R^*, Cert_R^*, id_\theta, PK_\theta$) to obtain the message M^* and then simulates the random oracle H_3 on its own to obtain $h_3^* = H_3(M^*, U^*, e(U^*, Cert_R^*), e(U^*, Cert_R^*)^{SK_R^*}, id_\theta, PK_\theta, id_R^*, PK_R^*)$. Using the oracle replay technique [29], B replays A_{II} with the same random tape but with the different hash value $h_3^{*'} (\neq h_3^*)$ to generate one more valid ciphertext $\sigma^{*'} = (C^*, U^*, V^{*'})$ such that $V^{*'} \neq V^*$. Since $\sigma^* = (C^*, U^*, V^*)$ and $\sigma^{*'} = (C^*, U^*, V^{*'})$ are both valid ciphertexts for the same message M^* and randomness r^* , we obtain the following relations:

$$\begin{aligned} V^* - V^{*'} &= (h_3^* SK_\theta + r^*) Cert_\theta - (h_3^{*'} SK_\theta + r^*) Cert_\theta \\ &= (h_3^* - h_3^{*'}) SK_\theta (H_1(id_\theta, PK_\theta) + \alpha)^{-1} Q. \end{aligned} \tag{8}$$

Then, we have the following relations:

$$\begin{aligned} e(H_1(id_\theta, PK_\theta)P + \alpha P, V^* - V^{*'}) \\ = e(P, (h_3^* - h_3^{*'}) SK_\theta Q). \end{aligned} \tag{9}$$

Because $Q = aP$ and $SK_\theta = b$, B can compute $abP = SK_\theta Q = (h_3^* - h_3^{*'})^{-1} (H_1(id_\theta, PK_\theta) + \alpha)(V^* - V^{*'})$ as the solution to the given CDH problem.

We now derive B 's advantage in solving the CDH problem. From the above construction, the simulation fails if any of the following events occurs: (1) E_1 : in the forge phase, B aborts because $id_S^* \neq id_\theta$; (2) E_2 : B fails in using the oracle replay technique to generate one more valid ciphertext. Clearly, $\Pr[\neg E_1] = 1/q_{cu}$. From the forking lemma [29], we

TABLE 3: Timings needed to perform atomic operations and representation of group elements in bits.

Curves	Relative timings (1 unit = 1 scalar multiplication in G)			Representation sizes (bits)	
	m	e	p	$ G $	$ G_T $
MNT/80	1	36	150	171	1026
SS/80	1	4	20	512	1024

know that $\Pr[\neg E_2] \geq 1/9$. Thus, we have that if A_{II} produces a valid forgery, then B will succeed in solving the CDH problem with probability $\epsilon' = \Pr[\neg E_1 \wedge \neg E_2] \geq 1/(9q_{cu})$. \square

5.3. Performance. To evaluate the performance of our new CBSC scheme, we compare our scheme with the previous CBSC schemes in terms of the computational cost and the communicational cost.

In the computational cost comparison, we consider four major operations: pairing, exponentiation in G_T , scalar multiplication in G , and hash. Among these operations, the pairing is considered as the heaviest time-consuming one in spite of the recent advances in the implementation technique. For simplicity, we denote these operations by p, e, m , and h , respectively. In the communicational cost comparison, ciphertext overhead represents the difference (in bits) between the ciphertext length and the message length, $|id|$ denotes the bit-length of user's identity, and $|G|$ and $|Z_p|$ denote the bit-length of an element in G and Z_p , respectively. Without considering precomputation, the performances of the compared CBSC schemes are listed in Table 2.

The efficiency of a pairing-based cryptosystem always depends on the chosen curve. Boyen [30] computes estimated relative timings for all atomic asymmetric operations (exponentiations and pairings) and representation sizes for group elements when instantiated in supersingular curves with 80-bit security (SS/80) and MNT curves with 80-bit security (MNT/80). In Table 3, we recall the data from [30].

To make a much clearer comparison, Table 4 gives the concrete values of the computational cost and the communicational cost for the compared CBE schemes according to the data in Table 3. As the hash operation is much more efficient than the multiplication in the group G , the costs of the hash operations are ignored.

In our proposed CBSC scheme, the *Signcrypt* algorithm does not require computing any time-consuming pairings. It only needs to compute two exponentiations in G_T , three scalar multiplications in G and three hashes in each signcryption operation. The *Designcrypt* algorithm needs to compute two pairings, two exponentiations in G_T , one

TABLE 4: Performance comparison of the CBSC schemes.

Schemes	Signcryption cost	Designcryption cost	Ciphertext overhead
MNT/80			
Ours	75	373	342
[18]	187	487	342
[20]	155	602	513 + id
[21]	447	594	2390
SS/80			
Ours	11	49	1024
[18]	28	65	1024
[20]	25	82	1536 + id
[21]	59	76	3072

scalar multiplication in G , and three hashes to designcrypt a ciphertext. From Tables 2 and 4, we can see that our scheme is more efficient than the previous CBSC scheme, especially in the computational efficiency. Actually, the computational performance of our scheme can be further optimized when $H_1(id_U, PK_U)P + P_{pub}$ can be precomputed. Such a precomputation enables us to additionally reduce one scalar multiplication computation in G and one hash computation in both the *Signcrypt* algorithm and the *Designcrypt* algorithm. In addition and most importantly, it is believed that our scheme is the first signcryption scheme in the certificate-based cryptographic setting that achieves security against both the public key replacement attacks and the insider attacks.

6. Conclusions

In this paper, we have introduced an improved security model of CBSC that captures both public key replacement attack and insider security. Our cryptanalysis has shown that Luo et al.'s CBSC scheme [20] is insecure in our security model. We have proposed a new CBSC scheme that resists both the key replacement attacks and the insider attacks. Compared with the previous CBSC schemes in the literature, the proposed scheme enjoys better performance, especially in the computation efficiency. However, a limitation of our schemes is that its security can only be achieved in the random oracle model [31]. Therefore, it would be interesting to construct a secure CBSC scheme without random oracles.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the anonymous referees for their helpful comments. This work is supported by the National Natural Science Foundation of China (Grant no. 61272542).

References

- [1] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the Advances in Cryptology (CRYPTO '84)*, pp. 47–53, 1984.
- [2] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proceedings of the 9th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT '03)*, pp. 452–473, 2003.
- [3] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '03)*, pp. 272–293, 2003.
- [4] S. S. Al-Riyami and K. G. Paterson, "CBE from CL-PKE: a generic construction and efficient schemes," in *Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC '05)*, pp. 398–415, January 2005.
- [5] P. Morillo and C. Ràfols, "Certificate-based encryption without random oracles," Tech. Rep. 2006/12, Cryptology ePrint Archive, <http://eprint.iacr.org/2006/012.pdf>.
- [6] D. Galindo, P. Morillo, and C. Ràfols, "Improved certificate-based encryption in the standard model," *Journal of Systems and Software*, vol. 81, no. 7, pp. 1218–1226, 2008.
- [7] J. K. Liu and J. Zhou, "Efficient certificate-based encryption in the standard model," in *Proceedings of the 6th International Conference on Security and Cryptography for Networks*, pp. 144–155, 2008.
- [8] Y. Lu, J. Li, and J. Xiao, "Constructing efficient certificate-based encryption with paring," *Journal of Computers*, vol. 4, no. 1, pp. 19–26, 2009.
- [9] Z. Shao, "Enhanced certificate-based encryption from pairings," *Computers and Electrical Engineering*, vol. 37, no. 2, pp. 136–146, 2011.
- [10] Y. Lu and J. Li, "Constructing certificate-based encryption secure against key replacement attacks," *ICIC Express Letters B: Applications*, vol. 3, no. 1, pp. 195–200, 2012.
- [11] B. G. Kang, J. H. Park, and S. G. Hahn, "A certificate-based signature scheme," in *Proceedings of the Cryptographers' Track at the RSA Conference (CT-RSA '04)*, pp. 99–111, 2004.
- [12] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen, "Certificate based (linkable) ring signature," in *Proceedings of the 3rd Information Security Practice and Experience Conference*, pp. 79–92, 2007.
- [13] J. Li, X. Huang, Y. Mu, W. Susilo, and Q. Wu, "Certificate-based signature: security model and efficient construction," in *Proceedings of the 4th European PKI Workshop*, pp. 110–125, 2007.
- [14] J. K. Liu, J. Baek, W. Susilo, and J. Zhou, "Certificate based signature schemes without pairings or random oracles," in *Proceedings of the 11th International conference on Information Security*, pp. 285–297, 2008.
- [15] W. Wu, Y. Mu, W. Susilo, and X. Huang, "Certificate-based signatures revisited," *Journal of Universal Computer Science*, vol. 15, no. 8, pp. 1659–1684, 2009.
- [16] J. Li, X. Huang, Y. Mu, W. Susilo, and Q. Wu, "Constructions of certificate-based signature secure against key replacement attacks," *Journal of Computer Security*, vol. 18, no. 3, pp. 421–449, 2010.
- [17] Y. Zheng, "Digital signcryption or how to achieve $\text{cost}(\text{signature} \& \text{encryption}) \ll \text{cost}(\text{signature}) + \text{cost}(\text{encryption})$," in *Proceedings of the 17th Annual International Cryptology Conference (CRYPTO '97)*, pp. 165–179, 1997.

- [18] F. Li, X. Xin, and Y. Hu, "Efficient certificate-based signcryption scheme from bilinear pairings," *International Journal of Computers and Applications*, vol. 30, no. 2, pp. 129–133, 2008.
- [19] L. Chen and J. Malone-Lee, "Improved identity-based signcryption," in *Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC '05)*, pp. 362–379, January 2005.
- [20] M. Luo, Y. Wen, and H. Zhao, "A certificate-based signcryption scheme," in *Proceedings of the International Conference on Computer Science and Information Technology (ICCSIT '08)*, pp. 17–23, September 2008.
- [21] J. Li, X. Huang, M. Hong, and Y. Zhang, "Certificate-based signcryption with enhanced security features," *Computers and Mathematics with Applications*, 2012.
- [22] J. H. Park and D. H. Lee, "On the security of status certificate-based encryption scheme," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 1, pp. 303–304, 2007.
- [23] J. An, Y. Dodis, and T. Rabin, "On the security of joint signature and encryption," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '02)*, pp. 83–107, 2002.
- [24] J. Baek, R. Steinfeld, and Y. Zheng, "Formal proofs for the security of signcryption," *Journal of Cryptology*, vol. 20, no. 2, pp. 203–235, 2007.
- [25] A. W. Dent, "Hybrid signcryption schemes with insider security," in *Proceedings of the 10th Australasian Conference on Information Security and Privacy (ACISP '05)*, pp. 253–266, July 2005.
- [26] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.
- [27] S. Mitsunari, R. Sakai, and M. Kasahara, "A new traitor tracing," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E85-A, no. 2, pp. 481–484, 2002.
- [28] S. D. Selvi, S. S. Vivek, D. Shukla, and P. R. Chandrasekaran, "Efficient and provably secure certificateless multi-receiver signcryption," in *Proceedings of the 2nd International Provable Security (ProvSec '08)*, pp. 52–67, 2008.
- [29] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, vol. 13, no. 3, pp. 361–396, 2000.
- [30] X. Boyen, "The BB_1 identity-based cryptosystem: a standard for encryption and key encapsulation," IEEE 1363.3, 2006, http://grouper.ieee.org/groups/1363/IBC/submissions/Boyen-bb1_ieee.pdf.
- [31] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 62–73, November 1993.

Research Article

Average Gait Differential Image Based Human Recognition

Jinyan Chen¹ and Jiansheng Liu²

¹ School of Computer Software, Tianjin University, Tianjin 300072, China

² College of Science, Jiangxi University of Science and Technology, Ganzhou 330200, China

Correspondence should be addressed to Jinyan Chen; chenjinyan@tju.edu.cn

Received 4 March 2014; Accepted 25 March 2014; Published 6 May 2014

Academic Editor: Fei Yu

Copyright © 2014 J. Chen and J. Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The difference between adjacent frames of human walking contains useful information for human gait identification. Based on the previous idea a silhouettes difference based human gait recognition method named as average gait differential image (AGDI) is proposed in this paper. The AGDI is generated by the accumulation of the silhouettes difference between adjacent frames. The advantage of this method lies in that as a feature image it can preserve both the kinetic and static information of walking. Comparing to gait energy image (GEI), AGDI is more fit to representation the variation of silhouettes during walking. Two-dimensional principal component analysis (2DPCA) is used to extract features from the AGDI. Experiments on CASIA dataset show that AGDI has better identification and verification performance than GEI. Comparing to PCA, 2DPCA is a more efficient and less memory storage consumption feature extraction method in gait based recognition.

1. Introduction

With the development of information and Internet technology, it is very necessary to authenticate and authorize human securely. The rapid growth of e-commerce also needs a reliable identification method to ensure safety transaction. As a promising authentication method, biometrics is attracting more and more attention. Biometrics overcomes the inherent flaws and limitations of conventional identification technology and brings a highly secure identification and authentication method. Traditional biometrical resources include fingerprint, face, and iris, which have been widely used for authentication. However, these biometrical features have the following disadvantage. (1) These features cannot be taken in a relative long distance. (2) User's cooperation is required to get good results. As a new biometrics method, gait based human identification overcomes the above limitation and is attracting more and more researchers.

Human gait is the manner of one walking, which was firstly studied in medical field. Doctors analyzed the human gait to find out whether patients had health problems [1, 2]. Later researchers found that just like fingerprint and iris, almost everyone has his distinctive walking style [3, 4]. So it was believed that gait could also be used as a biological feature

to recognize the person. Although suffering from clothing, shoes, view angle, or environmental context, human gait is still a promising identification method.

Human walking can be considered as an images sequence; however, most of the current model-free gait based identification methods extract features from image sequence without considering its contained spatiotemporal information. The method proposed in this paper focuses on the difference among the images sequence while constructing the feature image. The procedure can be described as follows. The silhouettes were normalized to the same height and aligned by the centroid. Then the difference between two adjacent silhouettes was accumulated to get the average gait differential image (AGDI) which is used as the feature image of one walking. Two-dimensional principal component analysis is used to extract feature from AGDI.

2. Related Work and Our Contribution

Usually recognition based on human gait includes several different approaches like walking, running, and jumping. In this paper we would like to restrict the recognition to walking. Currently human gait recognition can be divided into two categories: model-based methods and motion-based ones.

Model-based approaches aim to describe human movement using a mathematical model. Cunado et al. [5] used Hough transform to extract the positions of arms, legs, and torso and then use articulated pendulum to match those moving body parts. Yoo et al. [6] divided the body into head, neck, waist, leg, and arm by image segmentation and then got the moving curves of these body parts, respectively. Lee and Grimson [7] used 7 ellipses to model the human body and applied the ellipses' movement features to identify human. Yam et al. [8, 9] used dynamically coupled oscillator to describe and analyze the walking and running style of a person. Tafazzoli and Safabakhsh [10] constructed movements model based on anatomical proportions; then, Fourier transform was used to analyze human walking.

Model-free methods focused on the statistics information derived from the human gait. Cheng et al. [11] took HMM and manifold to analyze the relationship between the human and their gait images. Chen et al. [12] used parallel HMM to describe the features of human gait. Kale et al. [13] used "frieze" patterns to get features from image sequence and use them to identify a human. Murase and Sakai [14] speeded up the comparison of human gait by parametric eigenspace representation. Little and Boyd [15] derived scale-independent scalar features from optical flow information of walking figures to recognize individuals. Wang et al. [16] extracted feature by unwrapping the outer contour of silhouette and use PCA to reduce the dimension of the feature. Lee et al. [17] adopted product of Fourier coefficients as a distance measure between contours to recognize gait. Hu [18] combined the enhanced Gabor (EG) representation of the gait energy image and the regularized local tensor discriminate analysis (RLTDA) method in human identification. Hong et al. [19] proposed probabilistic framework to identify a human. Wang et al. [20] proposed spatiotemporal information analysis to get the features of human walking. Collins et al. [21] extract key frames from the image sequence and compare the key frames similarity by normalized correlation. Sarkar et al. [22] estimated the similarity between the gallery image sequence and the probe image sequence by directly computing the correlation between the frame pairs. Chen [23] proposes image correlation based human identification method.

Our method is similar to gait energy image (GEI) proposed by Yu et al. [24], Han and Bhanu [25], and frame difference energy image (FDEI) proposed by Chen et al. [26]. The major difference lies in the approach to generate the feature image. GEI is obtained by directly adding up every normalized silhouette. FDEI is obtained by taking the difference from every adjacent two frames and then combined with the "denoised" GEI. In this paper the difference between every two frames will be accumulated to generate average gait differential image. We also enhanced the feature extraction method by using 2DPCA, which has been used in the application of face recognition [27].

In comparison with the works of state of the art, the contributions of this paper are as follows.

Gait Representation Method. We propose a new gait feature representation which is called average gait differential image.

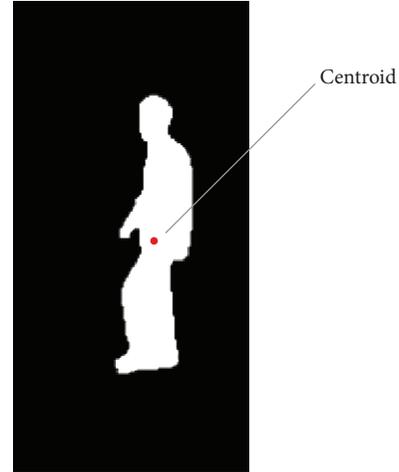


FIGURE 1: The centroid of a silhouette.

Comparing to GEI, our method has the advantage of better performance.

Feature Extraction Method. Two-dimensional principal component analysis (2DPCA) is used to extract features from AGDI, which can be more efficient and save more storage comparing to the widely used one-dimensional principal component analysis (PCA).

3. Average Gait Differential Image (AGDI) Representation

3.1. The Construction of AGDI. The construction of average differential image can be expressed in the following steps.

Silhouette Segmentation. Gauss model is used to get the background model from the original images sequence. To eliminate the effect of noise, every image is blurred by Gauss filter. The method proposed by Wang et al. [16] is used to extract walking object from the original images.

Normalization. To exclude the distance effect, every silhouette is normalized to the same height using bicubic interpolation.

Alignment and Subtraction. We define the centroid (x_c, y_c) of a silhouette as follows:

$$\begin{aligned} x_c &= \frac{1}{n} \sum_{i=1}^n x_i, \\ y_c &= \frac{1}{n} \sum_{i=1}^n y_i, \end{aligned} \quad (1)$$

where n is the number of pixels in the silhouette. Figure 1 shows the centroid of a silhouette.

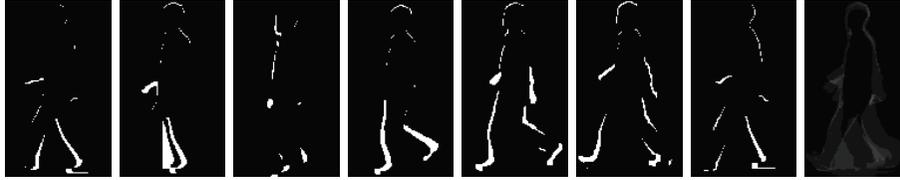


FIGURE 2: Differential images and average gait differential image.

Suppose that I_j and I_{j+1} are two adjacent images aligned by the centroid; the gait differential image D_j can then be defined as follows:

$$D_j(x, y) = \begin{cases} 0 & \text{if } I_j(x, y) = I_{j+1}(x, y) \\ 1 & \text{if } I_j(x, y) \neq I_{j+1}(x, y) \end{cases} \quad (2)$$

where j is the frame number in the image sequence and x and y are values in the 2D image coordinate.

Get the Average Gait Differential Image. By overlapping all the differential images of one human gait cycle, we can get the following average gait differential image:

$$G(x, y) = \frac{1}{N-1} \sum_{j=1}^{N-1} D_j(x, y), \quad (3)$$

where N is the number of frames in the complete gait cycle(s) of a silhouette sequence. Figure 2 show the differential images in a gait cycle and the average gait differential image, respectively.

3.2. Feature Extraction. Although in the previous section we have compressed the human gait features into one image, the dimensionality of the average gait differential image is still very large. The most commonly used dimensional reduction method is principal component analysis (PCA). In traditional PCA method, every two-dimensional image must be transformed into one-dimensional vector, leading to a covariance matrix with large size. This large matrix will use massive memory storage and is difficult to be evaluated accurately.

To reduce memory storage and speed up the calculation, this paper adopts the two-dimensional principal component analysis (2DPCA) to reduce the dimensionality, which was first proposed by Yang et al. [27] in the recombination of human face. Our final target is to project average gait differential image G , a $m \times n$ random matrix, onto a m -dimension projected vector Y which is called the projected feature vector of image G by the following linear transformation [27]:

$$Y = GW, \quad (4)$$

where W denotes a n -dimensional unitary column vector. To preserve the features of G , W should make Y have the

maximum scatter. We define S_y as the scatter of Y [27] as follows:

$$\begin{aligned} S_y &= E(Y - E(Y))(Y - E(Y))^T \\ &= E(GW - E(GW))(GW - E(GW))^T \\ &= E(G - E(G))WW^T(G - E(G))^T. \end{aligned} \quad (5)$$

The trace of S_y can be expressed as

$$\text{tr}(S_y) = W^T (E(G - E(G))^T (G - E(G))) W. \quad (6)$$

Here, we can define the image covariance matrix as

$$C_t = E(G - E(G))^T (G - E(G)). \quad (7)$$

In this paper, average gait differential image for each individual $(1, 2, \dots, M)$ is expressed as G_1, G_2, \dots, G_M , and then C_t can be calculated by

$$C_t = \sum_{i=1}^M (G_i - \bar{G})^T (G_i - \bar{G}). \quad (8)$$

Our target is to find a series of W_{opt} in formula (6) to make $\text{tr}(S_y)$ have the maximum value. According to [13], the optimal projection axis W_{opt} is the unitary orthogonal eigenvector of C_t corresponding to the largest eigenvalue. We define the first d unitary orthogonal eigenvector as W_1, W_2, \dots, W_d ; that is,

$$\begin{aligned} \{W_1, \dots, W_d\} &= \arg \max (W^T C_t W) \\ W_i W_j &= 0, \quad i \neq j, i, j = 1, \dots, d, \\ W_i W_j &= 1, \quad i = j, j = 1, \dots, d. \end{aligned} \quad (9)$$

The first d optimal projection vectors, W_1, \dots, W_d , are used to extract features from the average different images. That is to say, given an average gait differential image X , let

$$Y_k = GW_k, \quad k = 1, 2, \dots, d. \quad (10)$$

Then we get a series of projected feature vectors, $Y_1 \dots Y_d$, which are different from those scalar counterparts obtained from PCA. By using 2DPCA, the original $m \times n$ image is projected to a $m \times d$ ($d \leq n$) feature matrix Y as

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_d \end{bmatrix}. \quad (11)$$

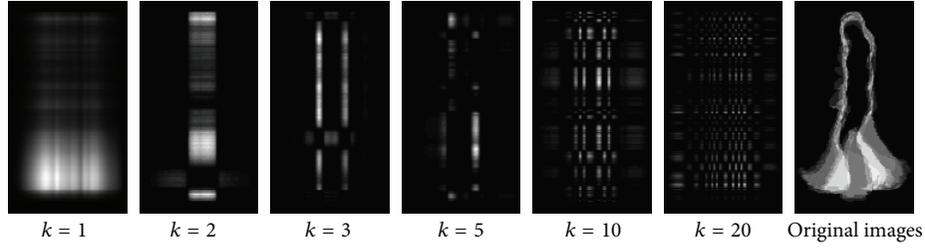


FIGURE 3: The reconstructed subimages ($k = 1, 2, 3, 5, 10, 20$) and the original images.

The distance between two feature matrixes is defined as

$$d(Y(i), Y(j)) = \sum_{k=1}^d \|Y_k^{(i)} - Y_k^{(j)}\|, \quad (12)$$

where $\|Y_k^{(i)} - Y_k^{(j)}\|$ means the Euclidean distance between two vectors.

3.3. Identification and Verification. Following the pattern proposed by Sarkar et al. [22], we evaluate performance for both identification and verification scenarios.

In the scenario of identification, every images sequence in the gallery (training set) is transformed to a $m \times d$ feature matrix $Y(i)$ by the method described in Section 3.2. Given a probe silhouette sequence, its transformed feature matrix is defined as P . This probe P is assigned to person k by using the nearest neighbor method:

$$d(Y(k), P) = \min_i d(Y(i) - P). \quad (13)$$

In the scenario of verification, the similarity between two feature matrixes is defined as the negative of distance; that is,

$$\text{Sim}(Y(i), Y(j)) = - \sum_{k=1}^d \|Y_k^{(i)} - Y_k^{(j)}\|. \quad (14)$$

In this paper the similarity between a probe, P_j , and $Y(i)$ in the gallery is defined as z -normed similarity [28]:

$$\text{Sim}(P_j, Y(i)) = \frac{\text{Sim}(P_j, Y(i)) - \text{Mean}_i \text{Sim}(P_j, Y(i))}{\text{s.d.}_i \text{Sim}(P_j, Y(i))}, \quad (15)$$

where s.d. is standard deviation.

FAR (false acceptance rate), FRR (false rejection rate), and EER (equal error rate) are used to evaluate the performance of verification [22].

3.4. DPCA-Based Average Gait Differential Image Reconstruction. In PCA the principal components and eigenvectors can be combined to reconstruct the original matrix. Similarly, 2DPCA can also be used to reconstruct an average gait differential silhouette.

Suppose that the eigenvectors corresponding to the largest d eigenvalues of C_t are W_1, \dots, W_d ; that is,

$$Y_k = G W_k, \quad k = 1, \dots, d, \quad (16)$$

$$[Y_1 \cdots Y_d] = X [W_1 \cdots W_d].$$

According to formula (9), $W_1 \cdots W_d$ are normal orthogonal vectors so the new reconstructed image \bar{G} can be expressed as

$$\begin{aligned} \bar{G} &= G [W_1 \cdots W_d] [W_1 \cdots W_d]^T = [Y_1 \cdots Y_d] [W_1 \cdots W_d]^T \\ &= \sum_{k=1}^d Y_k W_k^T. \end{aligned} \quad (17)$$

4. Experiments and Analysis

4.1. Data and Parameters. CASIA gait database (Dataset B) [29], one of the largest gait databases in gait-research field, is used in the following experiment. Dataset B consists of 124 subjects (93 males and 31 females) captured from 11 view angles (ranging from 0 to 180° degree with view angle interval of 18). For every person there are six normal walking sequences (named normal-01...normal-06) conducted from every view angle. Every walking sequence contains 3–8 gait cycles (about 40–100 frames). The video frame size is 320×240 pixels, and the frame rate is 25 fps. We use all the 124 objects in Dataset B to carry out our experiments.

In all the following experiments, 2DPCA method was used to get features from the images and 20 eigenvectors corresponding to the first 20 eigenvalues are used to produce features ($d = 20$). The size of original image is 240×320 except for special declaration.

For each person, from every view angle, we select the 39 frames from sequence normal-01 as the training data (gallery) and 13 frames (except for special declaration) from sequence normal-02 as the test data (probe).

For every view angle, each time we leave one training image sequence out and use the remainder as the training set. In the scenario of identification we calculate the distance

between the probe corresponding to the leave out training image sequence and the 124 classes (including the leave out image sequence). In the scenario of verification, we calculate the similarity between the probe corresponding to the leave out training image sequence and the 124 classes (including the leave out image sequence).

4.2. The Reconstruction of Subimage. Formula (17) indicates that we can reconstruct the subimage from the W_k and Y_k . Figure 3 shows the result of the reconstruction. For the consideration of illustration we normalize the brightness of every image into the range of 0–255.

As showed in Figure 3, the first and the second ($k = 1$, $k = 2$ in formula (17)) subimages corresponding to large eigenvectors of C_t contain the most energy of the original images. With the increase of k , the subimage contains more detailed information.

We also demonstrate the eigenvalue calculated by 2DPCA. Figure 4 shows the magnitude of the eigenvalues that quickly converges to zero.

4.3. Performance Evaluation

4.3.1. Comparison of AGDI and GEI. We compare the performance of our AGDI base method with that of gait energy image based method (In this paper, we use real template for GEI method [25]). Table 1 shows the rank 1 and rank 5 identification rates comparing with GEI.

To compare the performance of verification, we also evaluate the FAR (false acceptance rate) and FRR (false rejection rate) for AGDI and GEI. The ROC (receiver operating characteristic) curves under view angles 0° , 90° , and 180° are shown in Figures 5(a)–5(c). The comparison of EERs (equal error rate) is shown in Figure 5(d).

From Table 1 and Figure 5, we can see that almost under every view angle AGDI has better performance comparing to GEI (except that it is comparable under 0° view angle).

As also can be seen in Table 2, the best performance was obtained from the walking sequence taken from 0° , 90° , and 180° , while the worst was obtained from the walking sequence taken from 36° and 54° . This is probably due to the least visual deformation in the former degrees but more in the latter ones.

4.3.2. The Effect of Images Amount. From the definition of AGDI (formula (3)) we can see that the AGDI image is the average value of differential images. It should be expected that the use of more images as sample would contribute to a more precise result. To demonstrate this effect, a test was conducted by selectively choosing 13, 26, and 39 (approximately corresponding to 1, 2, and 3 gait cycles) images from 90 degree in sequences normal 01–02 as test dataset probe. Figure 5 shows the experimental result.

Indeed in Figure 5 the performance of 26 and 39 images is much better than that of 13.

4.3.3. Comparison of 2DPCA and PCA. We also design an experiment to compare the performance of 2DPCA and PCA, which were applied in the step of feature extraction,

TABLE 1: Comparison of identification performance of AGDI and GEI.

View angle	Rank 1 performance		Rank 5 performance	
	AGDI	GEI	AGDI	GEI
0°	72%	68%	88%	89%
18°	54%	37%	73%	60%
36°	35%	22%	51%	44%
54°	44%	26%	55%	45%
72°	66%	44%	86%	66%
90°	81%	77%	93%	90%
108°	78%	62%	92%	85%
126°	46%	36%	73%	53%
144°	49%	34%	72%	52%
162°	59%	35%	75%	48%
180°	88%	84%	94%	93%

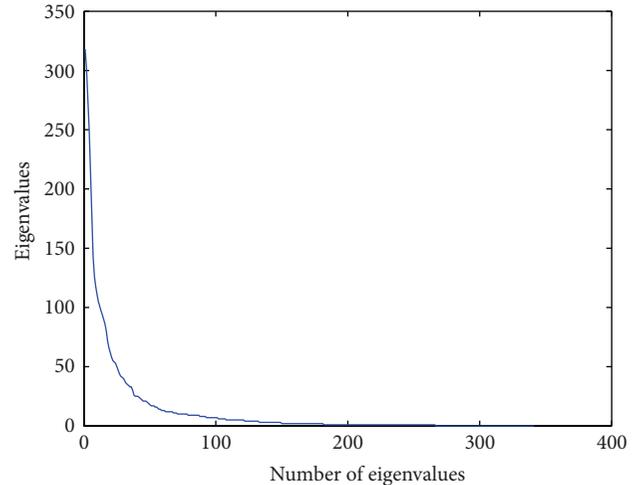


FIGURE 4: The magnitude of eigenvalue.

respectively. The data set view angle is 90° and every frame is resized to 120×160 .

As illustrated in Figure 7, the performance of 2DPCA, achieving the maximum at about 25 dimensions, is much better than PCA.

The key step for both PCA and 2DPCA is to get the eigenvalue and eigenvector from the covariance matrix C_t . For PCA method, every line of the covariance matrix corresponds to an image, as does the whole covariance matrix for the 2DPCA. That is, if the image size is $m \times n$, for PCA, the covariance matrix will be an $(m \times n) \times (m \times n)$ matrix, while for 2DPCA it is just an $m \times m$ matrix. We resize the silhouette to different sizes and compare the CPU time of PCA and 2DPCA for the step of feature extraction.

From Table 2 we can see that 2DPCA is more efficient than PCA, especially when the image is large.

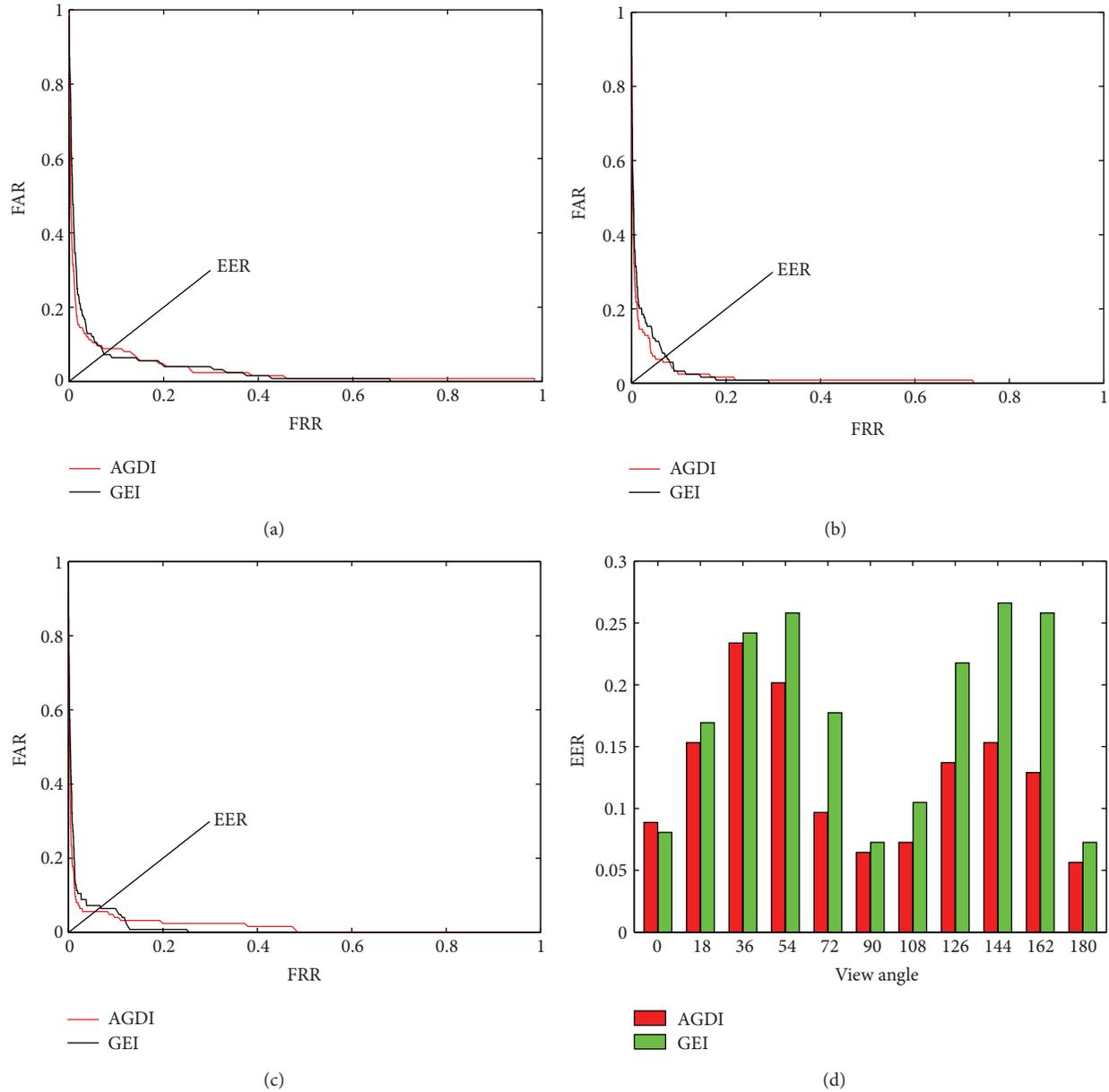


FIGURE 5: (a)–(c) The comparison of ROC curves of AGDI and GEI with view angles 0° , 90° , and 180° , respectively. (d) The comparison of EERs of AGDI and GEI with view angles 0° – 180° .

TABLE 2: Comparison of CPU time (ms) for PCA and 2DPCA feature extraction (CPU: Intel Core i3 2.30 GHz; RAM: 4 GB).

Feature extraction method	Image size					
	32×24	64×48	96×72	128×96	160×120	192×144
2DPCA	17 ms	47 ms	105 ms	167 ms	257 ms	431 ms
PCA	117 ms	318 ms	1273 ms	4288 ms	8896 ms	17876 ms

5. Conclusions

An average gait differential image based human recognition method is proposed in this paper (Figure 6). The Kernel idea of AGDI is to apply the average of differential image as the feature image and use the two-dimensional principal

component analysis to extract features. Experiments on CASIA dataset show the following. (1) Comparing to GEI, AGDI method achieves better identification and verification performance. (2) Comparing to PCA, 2DPCA is more efficient and needs lower memory storage. (3) The 0, 90, and 180 degrees silhouettes are more fit to AGDI base recognition.

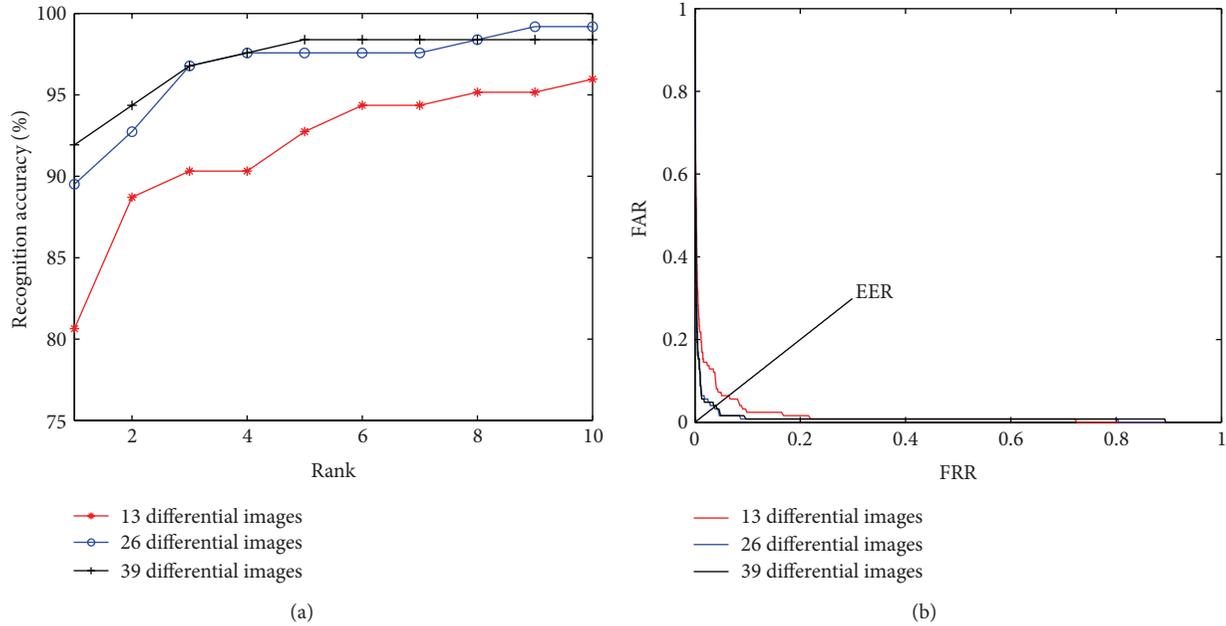


FIGURE 6: (a) Recognition accuracy of different probe sizes. (b) The comparison of ROC curves of different probe sizes.

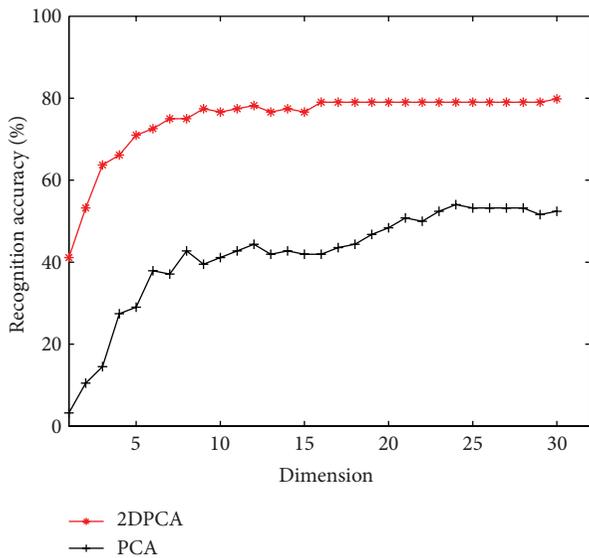


FIGURE 7: The rank 1 recognition accuracy comparison of 2DPCA and PCA.

Conflict of Interests

The authors declare that they have no financial or personal relationships with other people or organizations that can inappropriately influence their work. There are no professional or other personal interests of any nature or kind in any product, service, and/or company that could be construed as influencing the position presented in, or the review of, the paper.

Acknowledgments

This work is funded by Ph.D. Programs Foundation of Ministry of Education of China (no. 20100032120011). Dataset used in this paper is provided by Institute of Automation, Chinese Academy of Sciences [29].

References

- [1] M. W. Whittle, "Clinical gait analysis: a review," *Human Movement Science*, vol. 15, no. 3, pp. 369–387, 1996.
- [2] S. Lobet, C. Detrembleur, F. Massaad, and C. Hermans, "Three-dimensional gait analysis can shed new light on walking in patients with haemophilia," *The Scientific World Journal*, vol. 2013, Article ID 284358, 7 pages, 2013.
- [3] S. V. Stevenage, M. S. Nixon, and K. Vince, "Visual analysis of gait as a cue to identity," *Applied Cognitive Psychology*, vol. 13, no. 6, pp. 513–526, 1999.
- [4] C. Ben Abdelkader, R. Cutler, and L. Davis, "Stride and cadence as a biometric in automatic person identification and verification," in *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '02)*, 2002.
- [5] D. Cunado, M. S. Nixon, and J. N. Carter, "Automatic extraction and description of human gait models for recognition purposes," *Computer Vision and Image Understanding*, vol. 90, no. 1, pp. 1–41, 2003.
- [6] J. H. Yoo, D. Hwang, and M. S. Nixon, "Gender classification in human gait using support vector machine," in *Advanced Concepts for Intelligent Vision Systems: 7th International Conference, ACIVS 2005, Antwerp, Belgium, September 20-23, 2005. Proceedings*, vol. 3708 of *Lecture Notes in Computer Science*, pp. 138–145, Springer, 2005.
- [7] L. Lee and W. E. L. Grimson, "Gait analysis for recognition and classification," in *Proceedings of 5th IEEE International*

- Conference on Automatic Face Gesture Recognition*, pp. 155–162, 2002.
- [8] C. Y. C. Yam, M. S. Nixon, and J. N. Carter, “Automated person recognition by walking and running via model-based approaches,” *Pattern Recognition*, vol. 37, no. 5, pp. 1057–1072, 2004.
- [9] C. Yam, M. S. Nixon, and J. N. Carter, “Gait recognition by walking and running: a model-based approach,” in *Proceedings of the 5th Asian Conference on Computer Vision*, Melbourne, Australia, 2002.
- [10] F. Tafazzoli and R. Safabakhsh, “Model-based human gait recognition using leg and arm movements,” *Engineering Applications of Artificial Intelligence*, vol. 23, no. 8, pp. 1237–1246, 2010.
- [11] M.-H. Cheng, M.-F. Ho, and C.-L. Huang, “Gait analysis for human identification through manifold learning and HMM,” *Pattern Recognition*, vol. 41, no. 8, pp. 2541–2553, 2008.
- [12] C. Chen, J. Liang, H. Zhao, H. Hu, and J. Tian, “Factorial HMM and parallel HMM for gait recognition,” *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 39, no. 1, pp. 114–123, 2009.
- [13] A. Kale, A. Roy-Chowdhury, and R. Chellappa, “Gait-based human identification from a monocular video sequence,” in *Handbook on Pattern Recognition and Computer Vision*, C. H. Cheng and P. S. P. Wang, Eds., World Scientific Publishing Company, 3rd edition, 2005.
- [14] H. Murase and R. Sakai, “Moving object recognition in eigenspace representation: gait analysis and lip reading,” *Pattern Recognition Letters*, vol. 17, no. 2, pp. 155–162, 1996.
- [15] J. J. Little and J. E. Boyd, “Recognizing people by their gait: the shape of motion,” *Videre: Journal of Computer Vision Research*, vol. 1, no. 2, 1998.
- [16] L. Wang, T. Tan, H. Ning, and W. Hu, “Silhouette analysis-based gait recognition for human identification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1505–1518, 2003.
- [17] C. P. Lee, A. W. C. Tan, and S. C. Tan, “Gait recognition via optimally interpolated deformable contours,” *Pattern Recognition Letters*, vol. 34, no. 6, pp. 663–669, 2013.
- [18] H. Hu, “Enhanced gabor feature based classification using a regularized locally tensor discriminant model for multiview gait recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp. 1274–1286, 2013.
- [19] S. Hong, H. Lee, and E. Kim, “Probabilistic gait modelling and recognition,” *Computer Vision*, vol. 7, no. 1, pp. 56–70, 2013.
- [20] C. Wang, J. Zhang, L. Wang, J. Pu, and X. Yuan, “Human identification using temporal information preserving gait template,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2164–2176, 2012.
- [21] R. T. Collins, R. Gross, and S. Jianbo, “Silhouette-based human identification from body shape and gait,” in *Proceedings of 5th IEEE International Conference on Automatic Face Gesture Recognition*, 2002.
- [22] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, “The humanID gait challenge problem: data sets, performance, and analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 162–177, 2005.
- [23] J. Chen, “Gait correlation analysis based human identification,” *The Scientific World Journal*, vol. 2014, Article ID 168275, 8 pages, 2014.
- [24] S. Yu, T. Tan, K. Huang, K. Jia, and X. Wu, “A study on gait-based gender classification,” *IEEE Transactions on Image Processing*, vol. 18, no. 8, pp. 1905–1910, 2009.
- [25] J. Han and B. Bhanu, “Individual recognition using gait energy image,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 316–322, 2006.
- [26] C. Chen, J. Liang, H. Zhao, H. Hu, and J. Tian, “Frame difference energy image for gait recognition with incomplete silhouettes,” *Pattern Recognition Letters*, vol. 30, no. 11, pp. 977–984, 2009.
- [27] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, “Two-dimensional PCA: a new approach to appearance-based face representation and recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.
- [28] P. J. Phillips, P. Grother, R. Micheals, D. M. Blackburn, E. Tabassi, and M. Bone, “Face recognition vendor test 2002,” in *Proceedings of the IEEE International Workshop on Face Recognition*, 2003.
- [29] “CASIA Gait Database,” 2009, <http://www.sinobiometrics.com/>.

Research Article

A Secure and Fair Joint E-Lottery Protocol

Chin-Ling Chen,¹ Yuan-Hao Liao,¹ and Woei-Jiunn Tsaur²

¹ Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan

² Department of Information Management, Da-Yeh University, Changhua 51591, Taiwan

Correspondence should be addressed to Woei-Jiunn Tsaur; wjtsaur@mail.dyu.edu.tw

Received 23 January 2014; Accepted 3 March 2014; Published 4 May 2014

Academic Editors: M. Ivanovic and F. Yu

Copyright © 2014 Chin-Ling Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The attractive huge prize causes people to adore lotteries. Due to the very small probability of winning prizes, the players can enhance their probability of winning by using the method of joint purchase. In spite of many lottery schemes having been proposed, most e-lottery schemes focus on the players' privacy or computation overhead rather than support a joint purchase protocol on the Internet. In this paper, we use the multisignature and verifiable random function to construct a secure and fair joint e-lottery scheme. The players can check the lottery integrity, and the winning numbers can be verified publicly.

1. Introduction

Gambling has the property of nonpredictability and attractive prizes. Players have the chance to obtain a huge prize but of course they cannot predict who the winner will be. Hence, gambling is very fascinating for many people, and the lottery is one kind of popular gambling [1–3]. The players must select their favorite numbers and pay money to purchase lottery tickets. After the deadline of the purchasing phase, the lottery organization (LO) randomly generates the winning numbers. If no one wins the lottery, the prize money will accumulate for the next round. The attractive huge prizes are an extremely powerful factor causing people to purchase lottery tickets and the main reason remains popular among players.

In past years, many lottery schemes were proposed. In 2006, Chow et al. [4] proposed practical electronic lotteries with an offline trusted third party (TTP); their scheme can satisfy all of the identified requirements without the presence of TTP for generating the winning numbers; the result of this generation is publicly verifiable.

Next, Lee and Chang [5] proposed an electronic t -out-of- n lottery on the Internet in 2009. The scheme is based on the Chinese Remainder Theorem that allows lottery players to simultaneously select t out of n numbers in a ticket without iterative selection. The drawback of this scheme is that the computation overhead of players in purchasing lotteries is too heavy.

In the same year, Lee et al. [6] proposed noniterative privacy preservation for online lotteries. This scheme not only allows players to choose t -out-of- n numbers in lotteries without iterative selection but also preserves the privacy of players' choices. Nevertheless, the computation overhead in purchasing lotteries is still heavy for the player who accesses the Internet with mobile or wireless devices.

In an overview of the above schemes, we find that the majority of the schemes focus on the players' privacy or computation overhead but cannot support a joint purchase protocol on the Internet.

Due to the probability of winning prize being very small [7], the players can employ two strategies of purchase to enhance the probability of winning prizes as follows.

- (i) The player invites other players to collect more cash, and then the player purchases the sequential numbers to increase the probability of obtaining a prize.
- (ii) The player pays a small amount of money to purchase the lotteries in cooperation with other players.

To the best of our knowledge, there exist only two websites, called "e-Lottery Syndicates" [8] and "Myleto" [9], which provide a trading platform (TP) for purchases and a proxy purchase service. The difference between the above websites is that the former provides individual purchases while the latter provides joint purchases. Since our scheme

focuses on joint purchases, we chose “Myleto” to discuss joint purchases.

The process for joint purchase in “Myleto” enables the players to bet their favorite numbers by using the “Myleto,” and then “Myleto” counts the preferred numbers of players to generate the popular numbers after the deadline of purchase phase. Then, “Myleto” takes the popular numbers to purchase the lottery for the trusted lottery organization (LO). Finally, LO generates the winning numbers and publishes them on the bulletin board. Then “Myleto” distributes the different prizes to the winning players according to the numbers they bet.

Even if the solution for the joint purchase lottery exists, according to our observations some drawbacks remain.

(1) From the user’s viewpoint, the risks are as follows:

- (i) if the joint purchase players win the first prize, the person receiving the award has a chance to abscond with the funds;
- (ii) the player’s lottery purchase evidence depends on the picture at the time of purchase and the credit card transaction receipt. However, the former lacks credibility because it is easy to fake, and the latter lacks immediacy since the credit card transaction receipt adopts a monthly settlement;
- (iii) if the player’s purchase information is lost or the TP refuses to give out the prize, the player cannot proffer strong evidence to prove the winner is himself/herself.

(2) From the TP’s viewpoint, the risk is as follows:

- (i) if a malicious player forges a picture and a credit card transaction receipt to claim the prize, the TP will find it hard to recognize whether the prize claim evidence is true or false.

At present, we have seen that the current TP of joint purchase exhibits some drawbacks, so determining how to implement a fair and secure joint purchase e-lottery protocol is still an open issue.

Hence, we propose a fair and secure joint e-lottery protocol to guarantee the rights and interests of the players and TP. Simultaneously, our proposed protocol also supports individual purchases.

The proposed scheme must be able to achieve the following requirements [4–6] such that the proposed scheme can be applied in actual practice.

- (1) *Public Verification.* All the valid lottery tickets and the winning numbers must be verified via a verifiable random function.
- (2) *Fairness.* No one can predict the winning result before the winning numbers are published.
- (3) *Security.* No one can forge a winning lottery or impersonate a lottery winner to claim the prize.

- (4) *Correctness.* The players can verify the public information of the bulletin board by themselves.
- (5) *Anonymity.* Including lottery agents, no one can identify the participants by the lottery ticket.
- (6) *Convenience.* The legitimate players should be able to purchase lottery via Internet.
- (7) *Without Preregistration.* Players need not register at any lottery agent or drawing center in advance, as registration in advance is unnecessary; this requirement should conform to an electronic lottery to make it more realistic.
- (8) *No Online Trusted Third Party (TTP).* An electronic lottery is said to be impractical if the security of the entire mechanism depends on an online trusted third party.
- (9) *Participants’ Legality.* The scenario of the joint e-lottery scheme should ensure the participants’ legality via a multisignature.
- (10) *Support Joint and Individual E-Lottery Service.* The protocol must support joint and individual e-lottery service, respectively.

The remainder of this paper is organized to describe and analyze our joint e-lottery scheme as follows. Section 2 introduces related cryptographic techniques used in our scheme. Section 3 presents our proposed protocol, and the security requirements are analyzed in Section 4. Our conclusions are presented in the final section.

2. Preliminaries

In this section, we introduce three cryptographic techniques used in our scheme: a verifiable random function, an identity-based signature scheme, and an efficient identity-based RSA multisignature scheme.

2.1. Verifiable Random Function. A verifiable random function (VRF) was first proposed by Micali et al. [10]. Essentially, it is a pseudorandom function [11] providing noninteractively verifiable proof of the output’s correctness. Therefore, the above properties of VRF are suitable for our scheme.

On the basis of the notation in [12], a set of functions $F_{(\cdot)}(\cdot) : \{0, 1\}^k \rightarrow \{0, 1\}^{l(k)}$ is a verifiable function; suppose there exist polynomial-time algorithms $\text{Gen}(\cdot)$, $\text{Eval}(\cdot, \cdot)$, $\text{Prove}(\cdot, \cdot)$, $\text{Verify}(\cdot, \cdot, \cdot)$ such that

- (1) $\text{Gen}(k)$ is a probabilistic algorithm to generate a secret key SK that is generated by a random function and the corresponding public key PK that enables public verification;
- (2) $\text{Eval}(\text{SK}, x)$ is an algorithm that computes the VRF’s output $y = F_{\text{SK}}(x)$;

- (3) Prove(SK, x) is an algorithm that computes the proof π that $y = F_{SK}(x)$;
- (4) Verify(PK, x, y, π) is an algorithm that verifies $y = F_{SK}(x)$;
- (5) the VRF should satisfy the following properties.
- (6) uniqueness:

$$\text{Verify}(\text{PK}, x, y_1, \pi_1) = \text{Verify}(\text{PK}, x, y_2, \pi_2), \quad (1)$$

where $y_1 = y_2$;

- (7) computability: Eval(SK, x) = $F_{SK}(x)$ is efficiently computable;
- (8) provability: $(y, \pi) = (\text{Eval}(\text{SK}, x), \text{Prove}(\text{SK}, x))$ and Verify(PK, x, y, π);
- (9) pseudorandomness: the probability that an attacker can input any bit of $F_{SK}(x)$ for x his/her choice is negligible even if she/he has seen the values of many $F_{SK}(x')$ given $x' \neq x$.

2.2. Review of Shamir's Identity-Based Signature Scheme. In 1985, in order to simplify the public key authentication problem, Shamir [13] first offered the concept of an identity-based (ID-based) cryptosystem. In this system, each signer needs to register with a private key generator (PKG) and identify himself/herself before accessing the network resource. Once the registration is completed, the PKG will use the signer's identity to generate the secret key. The signer's identity may include the signer's name, email, and address. The advantage of this scheme is that there is no need for a public key directory in the system. The communicating parties only need to know the "identity" of his/her communication partner and the public key of the PKG is able to verify the signature or send an encrypted message.

We first introduce the notations used to explain how Shamir's scheme was constructed:

- (p_X, q_X) : a pair of large prime numbers;
- N_X : a large number, where $N_X = p_X \cdot q_X$, $\varphi(N_X) = (p_X - 1)(q_X - 1)$, and $\varphi(\cdot)$ is Euler's totient function;
- (e_X, d_X) : X 's public and private key, respectively, where $e_X d_X = 1 \pmod{\varphi(N_X)}$;
- $H(\cdot)$: a one way hash function;
- m : a message;
- $A \stackrel{?}{=} B$: comparing whether or not A is equal to B .

2.2.1. Private Key Generator (PKG) Keys. The private key generator (PKG) chooses its public and private key pair as follows.

Step 1. Run the probabilistic polynomial algorithm K_{RSA} to generate two random large primes, p_{PKG} and q_{PKG} .

Step 2. Choose a random public key e_{PKG} such that $\text{gcd}(e_{\text{PKG}}, \phi(N_{\text{PKG}})) = 1$ and compute the private key $d_{\text{PKG}} = e_{\text{PKG}}^{-1} \pmod{\phi(N_{\text{PKG}})}$.

2.2.2. Signer Secret Key Generation. In this algorithm, the signer gets a copy of his/her secret key from the PKG through a two-step process.

Step 1. A signer submits his/her identity to the PKG.

Step 2. The PKG uses its private key d_{PKG} to sign the signer's identity i by generating the secret key g such that $g = i^{d_{\text{PKG}}} \pmod{N_{\text{PKG}}}$.

2.2.3. Message Signing. To sign a message m , the signer with the secret key g and the corresponding public key e_{PKG} of the PKG signs a message m by generating a signature pair $\sigma = (T, S)$ as follows.

Step 1. Select a random number r and compute

$$T = r^{e_{\text{PKG}}} \pmod{N_{\text{PKG}}}. \quad (2)$$

Step 2. For the same random number r , compute

$$S = g \cdot r^{H(T,m)} \pmod{N_{\text{PKG}}}. \quad (3)$$

$\sigma = (T, S)$ is the complete signature of the message m .

2.2.4. Message Verification. The identity-based signature $\sigma = (T, S)$ of a signer with identity i is valid if and only if the following equality holds:

$$S^e = i \cdot T^{H(T,m)} \pmod{N_{\text{PKG}}}. \quad (4)$$

2.3. Review of Harn's Efficient Identity-Based RSA Multisignatures Scheme. In the 2008, Harn and Ren [14] first proposed a digital signature of a message generated by multiple signers with multiple private keys based on Shamir's identity-based signature (IBS) scheme. This was a first efficient identity-based RSA multisignatures scheme with both fixed length and verification time. Harn and Ren's scheme is secure against forgeries under chosen-message attack, against multisigner collusion attack, and adaptive chosen-identity attack.

2.3.1. Private Key Generator (PKG) Keys. The PKG chooses its public and private key pairs as follows.

Step 1. Runs the probabilistic polynomial algorithm K_{RSA} to generate two random large primes, p_{PKG} and q_{PKG} .

Step 2. Choose a random public key e_{PKG} such that $\text{gcd}(e_{\text{PKG}}, \phi(N_{\text{PKG}})) = 1$ and compute the private key $d_{\text{PKG}} = e_{\text{PKG}}^{-1} \pmod{\phi(N_{\text{PKG}})}$.

2.3.2. Signer Secret Key Generation. In this algorithm, the signer gets a copy of his/her secret key from the PKG through a two-step process.

Step 1. A signer submits his/her identity to the PKG.

Step 2. The PKG uses its private key d_{PKG} to sign the message digest of the identity to generate the secret key g_j , such that

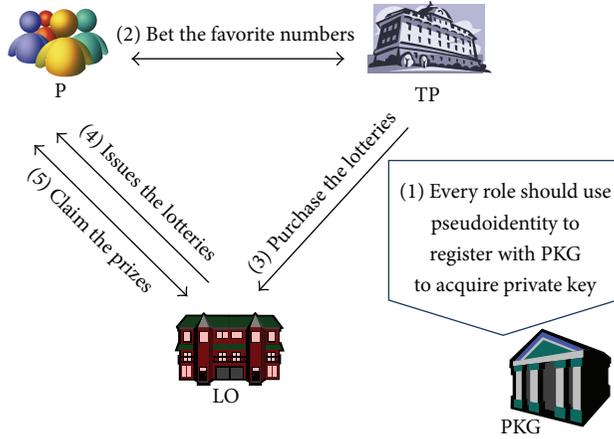


FIGURE 1: The structure of our scheme.

$g_j = i_j^{d_{\text{PKG}}} \bmod N_{\text{PKG}}$. No one will be able to distinguish between the identity and its message digest i_j .

2.3.3. *Message Signing.* To generate an identity-based multisignature, each signer carries out the followings steps.

Step 1. Choose a random integer r_j and compute $t_j = r_j^{e_{\text{PKG}}} \bmod N_{\text{PKG}}$.

Step 2. Broadcast t_j to other signers.

Step 3. Upon receiving of $t_j, j = 1, 2, \dots, l$, each signer computes

$$T = \prod_{j=1}^l t_j \bmod N_{\text{PKG}} \quad (5)$$

$$s_j = g_j \cdot r_j^{H(T,m)} \bmod N_{\text{PKG}}.$$

Step 4. Broadcast s_j to all signers.

Step 5. After receiving of $t_j, j = 1, 2, \dots, l$, the multisignatures component S can be computed as

$$S = \prod_{j=1}^l s_j \bmod N_{\text{PKG}}. \quad (6)$$

The multisignature for message m is $\sigma = (T, S)$.

2.3.4. *Multisignature Verification.* To verify a multisignature $\sigma = (T, S)$ of a message m of signers whose identities are i_1, i_2, \dots, i_l , anyone can verify the correctness as follows:

$$S^{e_{\text{PKG}}} \stackrel{?}{=} (i_1, i_2, \dots, i_l) \cdot T^{H(T,m)} \bmod N_{\text{PKG}}. \quad (7)$$

3. The Proposed Joint E-Lottery Protocol

The structure of our scheme is illustrated in Figure 1.

There are four participants involved in the proposed e-lottery scheme.

- (1) *Private Key Generator (PKG).* The off-line trusted third party which generates private keys to all participants.
- (2) *Player (P).* The player is a participator in the lottery gamble.
- (3) *Trading Platform (TP).* The trading platform is a website to provide players for joining the e-lottery game.
- (4) *Lottery Originator (LO).* The LO issues the lotteries, generates the winning numbers to sell lotteries to gain revenue, and gives out the prizes.

Step 1. P, TP, LO \leftrightarrow PKG: all participants must register to PKG to acquire their private key with his/her pseudonymity.

Step 2. P \leftrightarrow TP: the players bet their favorite numbers to the TP.

Step 3. TP \rightarrow LO: the TP gathers the statistics on the betting numbers to generate the majority of popular numbers and then purchases the popular numbers with the LO.

Step 4. LO \rightarrow P: the LO issues lotteries to the players.

Step 5. P \leftrightarrow LO: after the winning numbers are announced, the winning players use their winning lotteries and private keys to claim the prizes won.

The following notations are used in our protocol:

number _{j} : the favorite numbers of the j th player;

chain _{j} : the published hash chain set of the valid random seed β_j generated by player, which is involved in generating the winning number, where chain₀ is the initial vector; the chain₀ = 0, chain₁ = $H(\text{chain}_0, \beta_1)$, chain₂ = $H(\text{chain}_1, \beta_2)$, and chain _{j} = $H(\text{chain}_{j-1}, \beta_j)$;

C_i : the i th ciphertext;

σ_m : the identity-based signature of message m ;

m_{req} : the request message;

i_j : the message digest of the j th player's identity;

PL: the purchased list, where PL = (i_j, β_j) ;

$H(\cdot)$: one way hash function [15];

r_X : the random number is selected by X ;

β_j : the hash value, where $\beta_j = H(r_X)$;

$K_{X,Y}$: the session key between the X and Y which is constructed by IETF [16, 17];

$E(K_{X,Y}, (m))$: an encryption function which uses the session key $K_{X,Y}$ to encrypt the message m ;

$D(K_{X,Y}, (C))$: a decryption function which uses the session key $K_{X,Y}$ to decrypt the ciphertext C .

3.1. Constructing the Session Key Model. Diffie and Hellman proposed a key agreement protocol [18] in 1978. The RFC 2631 was drawn up for the key agreement protocol in 1999 by the IETF. Therefore, we use the RFC 2631 protocol to construct the session keys. The session keys are used in our protocol with three situations. First, when the purchase is individual, the player must share the session key to protect his/her favorite numbers. Second, the TP and LO are jointed to sign the multisignature; they must share a common secret key to encrypt the signature. Third, the LO issues the lotteries to players, and the winning players send the claim prize message to the LO; they must also share a session key to encrypt or decrypt the messages.

3.2. The Initialization Phase. In this phase, the PKG performs the keys generating function to generate the public and private keys. On the other hand, the LO performs the VRF to generate the related functions and then publishes it.

Step 1. The PKG selects a random number k and then performs $\text{Gen}(k)$ to generate the public e_{PKG} and private d_{PKG} .

Step 2. The LO performs the VRF to generate the related functions that include $\text{Eval}(\cdot, \cdot)$, $\text{Prove}(\cdot, \cdot)$, and $\text{Verify}(\cdot, \cdot, \cdot)$.

3.3. The Registration Phase. In this phase, all of the roles submit their identities to the PKG to become legal participants. Notably, the players must submit their identities (including the players' name, email, and addresses) and a random number to PKG and then PKG signs the message digest of the identity by its d_{PKG} and d_{PKG} .

The PKG computes the participants' private keys with its d_{PKG} as in the following equations:

$$d_j = t_j^{d_{\text{PKG}}} \bmod N_{\text{PKG}} \quad \text{for } j = 1, 2, \dots, n.$$

$$d_{\text{TP}} = \text{ID}_{\text{TP}}^{d_{\text{PKG}}} \bmod N_{\text{PKG}}, \tag{8}$$

$$d_{\text{LO}} = \text{ID}_{\text{LO}}^{d_{\text{PKG}}} \bmod N_{\text{PKG}}.$$

After that, the PKG publishes ID_{TP} and ID_{LO} on the bulletin board.

3.4. The Players Bet for Lottery Numbers Phase. In this phase, the players can bet their favorite numbers via the TP and then the TP publishes the purchase information on the bulletin board. When this phase is finished, the TP will send bulletin board information to the LO. According to the received information, the LO publishes the winning numbers. Moreover, players, TP, and LO can use the published bulletin board information to check whether or not the following three information items are correct.

- (1) The players' purchased lotteries are included in the hash chain.
- (2) The players' bet numbers are valid or not.
- (3) The players are legal or not.

TABLE 1: The information of bulletin board.

Initial condition			
	chain ₀ = 0		
	chain ₁ = H(chain ₀ , β ₁)		
	chain ₂ = H(chain ₁ , β ₂)		
	⋮		
	chain _j = H(chain _{j-1} , β _j)		
Identity information	Hash chain value	Hash value of the random number r _{P_j}	Selected favorite number
(i ₁ , σ _{i₁})	chain ₁	β ₁	number ₁
(i ₂ , σ _{i₂})	chain ₂	β ₂	number ₂
⋮	⋮	⋮	⋮
(i _j , σ _{i_j})	chain _j	β _j	number _j

The individual purchase is also included in the hash chain and the purchased information (including identity information, hash chain value, and hash value of the random number) is also published on the bulletin board, except for the selected favorite numbers.

The players bet for lottery numbers phase is illustrated in Figure 2, and the bulletin board information is illustrated in Table 1.

If anyone questions the players' legality then they can use the signature of the players' identity of the bulletin board to verify the legality of the players by

$$s_{P_j}^{e_{\text{PKG}}} \stackrel{?}{=} i_j \cdot t_{P_j}^{H(t_{P_j}, i_j)} \bmod N_{\text{PKG}}. \tag{9}$$

Step 1. If the purchase is individual then the player must compute session key $K_{P_j\text{-TP}}$ (refer to Section 3.1), using it to protect the individual's favorite number number_j as follows:

$$C_1 = E\left(K_{P_j\text{-TP}}, (\text{number}_j)\right). \tag{10}$$

The individual and joint purchases are both required to process (11)–(15).

Then, the j th player selects a random number r_{P_j} to compute

$$\beta_j = H(r_{P_j}), \tag{11}$$

$$t_{P_j} = t_{P_j}^{e_{\text{PKG}}} \bmod N_{\text{PKG}}.$$

The P_j uses his/her private key d_j to compute

$$s_{P_j} = d_j \cdot r_{P_j}^{H(t_{P_j}, i_j, \text{number}_j, \beta_j)} \bmod N_{\text{PKG}}. \tag{12}$$

Here, we denote the signature as follows:

$$\sigma_{i_j, \text{number}_j, \beta_j} = (s_{P_j}, t_{P_j}). \tag{13}$$

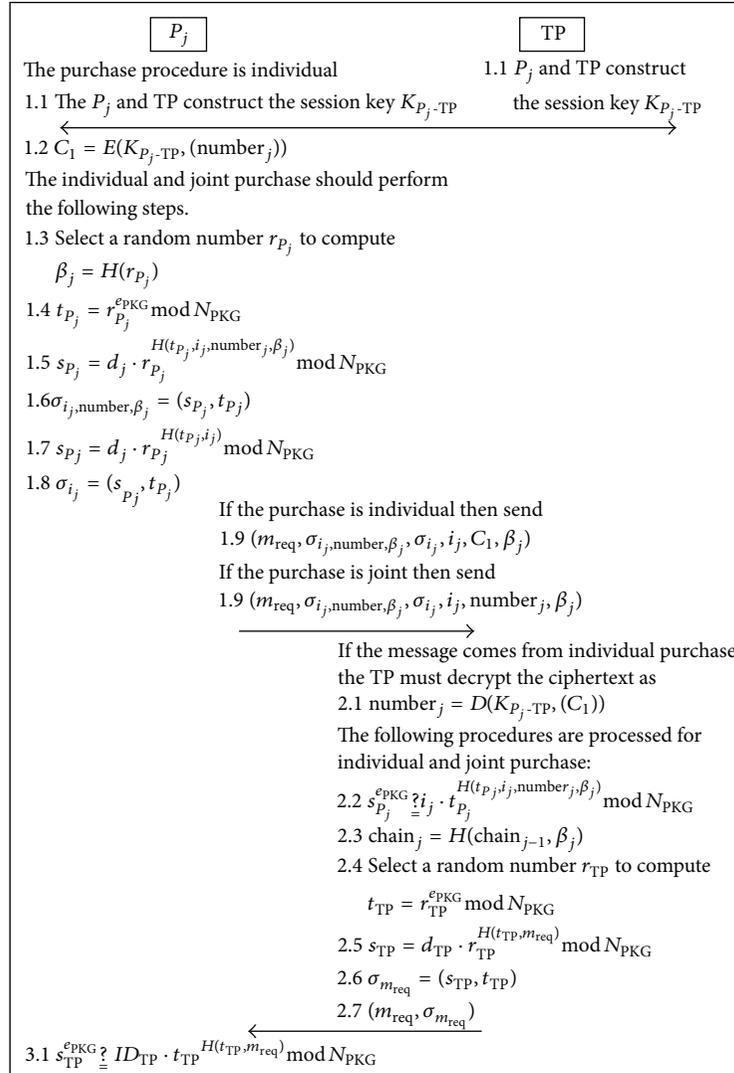


FIGURE 2: Overview of the players bet for lottery number phase.

Finally, the P_j uses his/her private key d_j to sign his/her identity i_j as follows:

$$s_{P_j} = d_j \cdot r_{P_j}^{H(t_{P_j}, i_j)} \bmod N_{PKG}. \quad (14)$$

We denote the signature as follows:

$$\sigma_{i_j} = (s_{P_j}, t_{P_j}). \quad (15)$$

The difference between the multisignature σ_{i_j} and $\sigma_{i_j, \text{number}_j, \beta_j}$ is that the former is published on the bulletin board and all participants can use it to verify the player's legality, while the latter is used to achieve the message nonrepudiation for the TP.

Afterward, if the purchase is individual then the request message m_{req} , signature $\sigma_{i_j, \text{number}_j, \beta_j}$, and related parameters (i_j, C_1, β_j) are sent to the TP.

If the purchase is joint, the request message m_{req} , signature $\sigma_{i_j, \text{number}_j, \beta_j}$, and related parameters $(i_j, \text{number}_j, \beta_j)$ are sent to the TP.

Step 2. After receiving the message, if it comes from individual purchase then the TP must decrypt the ciphertext C_1 to obtain number_j and then the following procedures are processed for individual and joint purchases.

First, the TP checks the validity of signature as follows:

$$s_{P_j} \stackrel{?}{=} i_j \cdot t_{P_j}^{H(t_{P_j}, i_j, \text{number}_j, \beta_j)} \bmod N_{PKG}. \quad (16)$$

The TP links β_j into the hash chain as follows:

$$\text{chain}_j = H(\text{chain}_{j-1}, \beta_j). \quad (17)$$

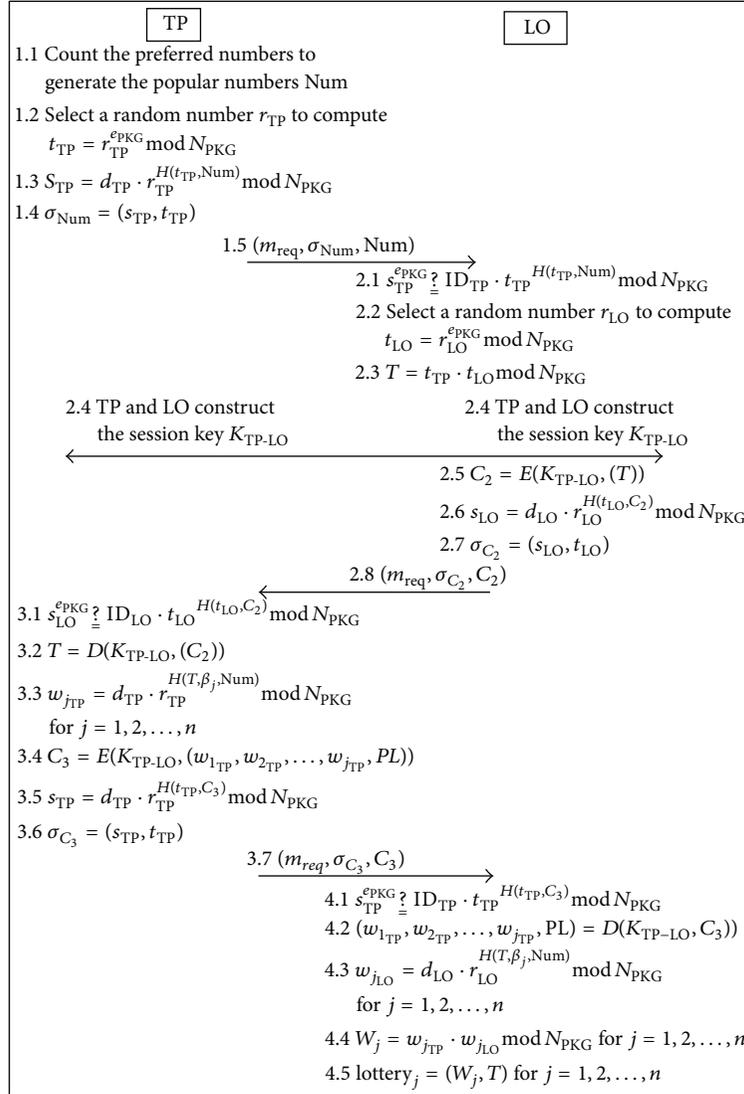


FIGURE 3: Overview of the purchase phase.

Next, the TP selects a random number r_{TP} to compute

$$\begin{aligned} t_{TP} &= r_{TP}^{e_{PKG}} \bmod N_{PKG} \\ s_{TP} &= d_{TP} \cdot r_{TP}^{H(t_{TP}, m_{req})} \bmod N_{PKG}. \end{aligned} \quad (18)$$

Here, we denote the signature of m_{req} as follows:

$$\sigma_{m_{req}} = (s_{TP}, t_{TP}). \quad (19)$$

Finally, TP sends signature $(\sigma_{m_{req}}, m_{req})$ to the P_j .

Step 3. After receiving $(\sigma_{m_{req}}, m_{req})$, P_j checks the validity of signature as follows:

$$s_{TP}^{e_{PKG}} \stackrel{?}{=} ID_{TP} \cdot t_{TP}^{H(t_{TP}, m_{req})} \bmod N_{PKG}. \quad (20)$$

If the signature is invalid, then the P_j terminates the transaction.

3.5. The Purchase Phase. After the purchase deadline, the TP gathers the statistics on numbers to generate the popular numbers. Subsequently, the TP sends the purchase message that includes purchase list and the partial signature to the LO. Note that the lottery's numbers of individual purchase are determined by individual buyers rather than through counting by TP; individual purchase is the same as joint purchase.

- (1) The individual purchase is included in the purchase list.
- (2) The TP also computes the partial signature for individual purchase.

The overview of the purchase phase is illustrated in Figure 3.

Step 1. After the purchase deadline, the TP counts the preferred numbers of all of the players to generate the popular

numbers Num; then the TP selects a random number r_{TP} to compute the partial signature t_{TP} and s_{TP} as follows:

$$\begin{aligned} t_{TP} &= r_{TP}^{e_{PKG}} \bmod N_{PKG} \\ s_{TP} &= d_{TP} \cdot r_{TP}^{H(t_{TP}, \text{Num})} \bmod N_{PKG}. \end{aligned} \quad (21)$$

Here, we denote the signature of Num as follows:

$$\sigma_{\text{Num}} = (s_{TP}, t_{TP}). \quad (22)$$

Finally, the TP sends the request message m_{req} , signature σ_{Num} , and the popular numbers Num to the LO.

Step 2. Before receiving the message $(m_{\text{req}}, \sigma_{\text{Num}}, \text{Num})$, the LO checks the signature validity as follows:

$$s_{TP}^{e_{PKG}} \stackrel{?}{=} \text{ID}_{TP} \cdot t_{TP}^{H(t_{TP}, \text{Num})} \bmod N_{PKG}. \quad (23)$$

Subsequently, the LO selects random number r_{LO} to compute

$$t_{LO} = r_{LO}^{e_{PKG}} \bmod N_{PKG}. \quad (24)$$

LO uses the partial signature of the TP and LO to compute T as follows:

$$T = t_{TP} \cdot t_{LO} \bmod N_{PKG}. \quad (25)$$

The TP and LO construct the session key K_{TP-LO} and then encrypt the partial signature as follows:

$$C_2 = E(K_{TP-LO}, (T)). \quad (26)$$

The LO uses the private key d_{LO} to compute the partial signature of C_2 as follows:

$$s_{LO} = d_{LO} \cdot r_{LO}^{H(t_{LO}, C_2)} \bmod N_{PKG}. \quad (27)$$

Here, we denote the signature of C_2 as

$$\sigma_{C_2} = (s_{LO}, t_{LO}). \quad (28)$$

Finally, the LO sends $(m_{\text{req}}, \sigma_{C_2}, C_2)$ to TP.

Step 3. After receiving $(\sigma_{C_2}, m_{\text{req}})$, the TP checks the validity of signature as follows:

$$s_{LO}^{e_{PKG}} \stackrel{?}{=} \text{ID}_{LO} \cdot t_{LO}^{H(t_{LO}, C_2)} \bmod N_{PKG}. \quad (29)$$

The TP uses the session key K_{TP-LO} to decrypt the cipher text as follows:

$$T = D(K_{TP-LO}, (C_2)). \quad (30)$$

According to the purchased list PL, the TP uses its private key d_{TP} to compute the partial multisignatures $w_{j_{TP}}$ of the player's lottery as in (31) as follows:

$$w_{j_{TP}} = d_{TP} \cdot r_{TP}^{H(T, \beta_j, \text{Num})} \bmod N_{PKG}, \quad \text{for } j = 1, 2, \dots, n. \quad (31)$$

To protect the message, the TP uses the session key K_{TP-LO} to encrypt parameters $(w_{1_{TP}}, w_{2_{TP}}, \dots, w_{j_{TP}}, \text{PL})$ as follows:

$$C_3 = E(K_{TP-LO}, (w_{1_{TP}}, w_{2_{TP}}, \dots, w_{j_{TP}}, \text{PL})). \quad (32)$$

The TP uses its private key d_{TP} to compute the partial signature of C_3 as

$$s_{TP} = d_{TP} \cdot r_{TP}^{H(t_{TP}, C_3)} \bmod N_{PKG}. \quad (33)$$

Here, we denote the signature of C_3 as follows:

$$\sigma_{C_3} = (s_{TP}, t_{TP}). \quad (34)$$

Afterward, the TP sends the request message m_{req} , signature σ_{C_3} , and cipher message C_3 to the LO.

Step 4. Once receiving the message $(m_{\text{req}}, \sigma_{C_3}, C_3)$, the LO checks the signature validity as follows:

$$s_{TP}^{e_{PKG}} \stackrel{?}{=} \text{ID}_{TP} \cdot t_{TP}^{H(t_{TP}, C_3)} \bmod N_{PKG}. \quad (35)$$

The LO uses its private key d_{LO} to decrypt the cipher text as follows:

$$(w_{1_{TP}}, w_{2_{TP}}, \dots, w_{j_{TP}}, \text{PL}) = D(K_{TP-LO}, (C_3)). \quad (36)$$

According to the purchase list PL, the LO uses its private key d_{LO} to compute the partial multisignatures $w_{j_{LO}}$ of all players as follows:

$$w_{j_{LO}} = d_{LO} \cdot r_{LO}^{H(T, \beta_j, \text{Num})} \bmod N_{PKG}, \quad \text{for } j = 1, 2, \dots, n \quad (37)$$

and then LO uses the partial multisignatures of $w_{j_{TP}}$ and $w_{j_{LO}}$ to compute

$$W_j = w_{j_{TP}} \cdot w_{j_{LO}} \bmod N_{PKG}, \quad \text{for } j = 1, 2, \dots, n. \quad (38)$$

We denote the lottery lottery_{*j*} as

$$\text{lottery}_j = (W_j, T), \quad \text{for } j = 1, 2, \dots, n. \quad (39)$$

3.6. The Lottery Issue Phase. Upon receiving the purchase message, the LO issues the lotteries to all players (including the joint purchase and individual purchase) and then the players can apply the multisignature to verify the validity of the lottery. The lottery issue phase is illustrated in Figure 4.

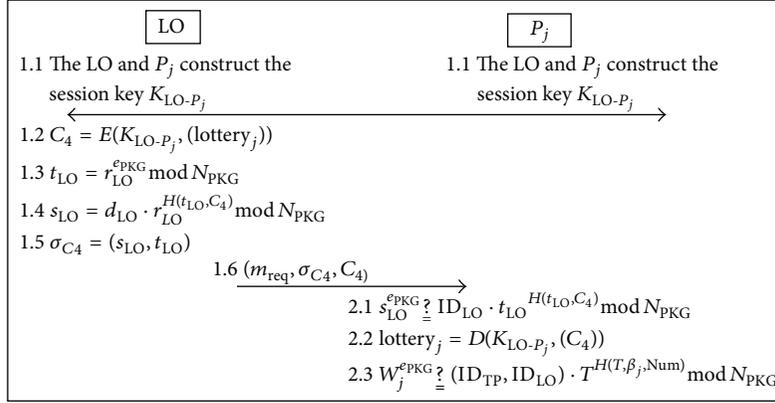


FIGURE 4: Overview of the lottery issue phase.

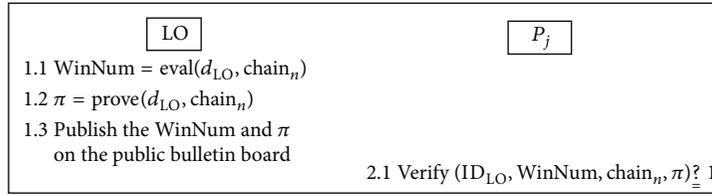


FIGURE 5: Overview of the winning numbers generation and verification phase.

Step 1. The LO and P_j construct the session key K_{LO-P_j} and then encrypt the lottery $_j$ as follows:

$$C_4 = E(K_{LO-P_j}, (\text{lottery}_j)). \quad (40)$$

Next, the LO selects random number r_{LO} to compute

$$t_{LO} = r_{LO}^{e_{PKG}} \bmod N_{PKG} \quad (41)$$

and uses its private key d_{LO} to compute the partial signature as follows:

$$s_{LO} = d_{LO} \cdot r_{LO}^{H(t_{LO}, C_4)} \bmod N_{PKG}. \quad (42)$$

Here, we denote the signature of C_4 as follows:

$$\sigma_{C_4} = (s_{LO}, t_{LO}). \quad (43)$$

Afterward, the LO sends the request message m_{req} , signature σ_{C_4} , and ciphertext C_4 to P_j .

Step 2. When receiving the message (m_{req}, σ_4, C_4) , P_j checks the validity of signature as follows:

$$s_{LO} \stackrel{e_{PKG}}{=} ID_{LO} \cdot t_{LO}^{H(t_{LO}, C_4)} \bmod N_{PKG} \quad (44)$$

and then uses the session key K_{LO-P_j} to decrypt ciphertext C_4 as follows:

$$\text{lottery}_j = D(K_{LO-P_j}, (C_4)). \quad (45)$$

Finally, P_j checks the validity of signature as follows:

$$W_j \stackrel{e_{PKG}}{=} (ID_{TP}, ID_{LO}) \cdot T^{H(T, \beta_j, Num)} \bmod N_{PKG}. \quad (46)$$

3.7. The Winning Numbers Generation and Verification Phase. After the lottery purchase deadline, the LO uses the function of winning numbers generation with the value of final hash chain to generate the winning numbers and then publishes it on the bulletin board. The overview of the winning numbers generation and verification phase is illustrated in Figure 5.

Simultaneously, if the players question whether or not the LO is honest, they can use the public verification function to verify the correctness of the winning numbers.

Step 1. The LO uses its private key d_{LO} and the value of final hash chain chain_n to calculate

$$\begin{aligned} \text{WinNum} &= \text{Eval}(d_{LO}, \text{chain}_n) \\ \pi &= \text{Prove}(d_{LO}, \text{chain}_n). \end{aligned} \quad (47)$$

Finally, the LO publishes the WinNum and π on the bulletin board.

Step 2. After the winning numbers are published, any player can check the correctness of the winning numbers via the public verification function as follows:

$$\text{Verify}(ID_{LO}, \text{WinNum}, \text{chain}_n, \pi) \stackrel{?}{=} 1. \quad (48)$$

Scenario 2. Suppose that any player suspects the correctness of winning lotteries.

Proof. The one suspecting can use the related parameters (including random number of the winning player selected r_{P_j} , the winning numbers WinNum, and the winning lottery (W_j, T)) and (56) to verify the correctness of winning lotteries. The verification equation is as in (56), where $\beta'_j = H(r_{P_j})$.

The derivation of the verification is shown as follows:

$$\begin{aligned}
 & (ID_{TP} \cdot ID_{LO}) \cdot T^{H(T, \beta'_j, \text{WinNum})} \pmod{N_{\text{PKG}}} \\
 &= (d_{TP}^{e_{\text{PKG}}} \cdot d_{LO}^{e_{\text{PKG}}}) \cdot (t_{TP} \cdot t_{LO})^{H(T, \beta'_j, \text{WinNum})} \pmod{N_{\text{PKG}}} \\
 &= (d_{TP}^{e_{\text{PKG}}} \cdot d_{LO}^{e_{\text{PKG}}}) \cdot (r_{TP}^{e_{\text{PKG}}} \cdot r_{LO}^{e_{\text{PKG}}})^{H(T, \beta'_j, \text{WinNum})} \pmod{N_{\text{PKG}}} \\
 &= (d_{TP}^{e_{\text{PKG}}} \cdot d_{LO}^{e_{\text{PKG}}}) \cdot \left(r_{TP}^{e_{\text{PKG}} \cdot H(T, \beta'_j, \text{WinNum})} \right. \\
 &\quad \left. \cdot r_{LO}^{e_{\text{PKG}} \cdot H(T, \beta'_j, \text{WinNum})} \right) \pmod{N_{\text{PKG}}} \\
 &= \left((d_{TP} \cdot d_{LO}) \cdot \left(r_{TP}^{H(T, \beta'_j, \text{WinNum})} \right. \right. \\
 &\quad \left. \left. \cdot r_{LO}^{H(T, \beta'_j, \text{WinNum})} \right) \right)^{e_{\text{PKG}}} \pmod{N_{\text{PKG}}} \\
 &= \left(\left(d_{TP} \cdot r_{TP}^{H(T, \beta'_j, \text{WinNum})} \right) \right. \\
 &\quad \left. \cdot \left(d_{LO} \cdot r_{LO}^{H(T, \beta'_j, \text{WinNum})} \right) \right)^{e_{\text{PKG}}} \pmod{N_{\text{PKG}}} \\
 &= (w_{j_{TP}} \cdot w_{j_{LO}})^{e_{\text{PKG}}} \pmod{N_{\text{PKG}}} \\
 &= W_j^{e_{\text{PKG}}} \pmod{N_{\text{PKG}}}. \tag{57}
 \end{aligned}$$

Because the multisignature of the winning numbers is valid, the winning lottery is correct. \square

4.2. Fairness. No one can predict the winning result before the LO publishes the winning numbers.

Scenario 3. If a player wants to predict or bias the winning result, he or she will fail.

Proof. Since each purchasing behavior is random and occasional, the final value of hash chain chain_n is contributed by all of the lotteries. Hence, no one can learn the final value of the hash chain chain_n . \square

4.3. Security. No one can forge winning lotteries or impersonate lottery winners to claim their prize.

Scenario 4. If *Eve* tries to forge a winning lottery to claim the prize, she will fail.

Proof. In reviewing the purchase phase, the TP and LO used their private keys d_{TP} and d_{LO} to sign the lotteries. On the other hand, if *Eve* wants to fake the winning lottery, she must forge their private keys, respectively. In fact, she must solve the factorization problem in RSA cryptosystems [19]. \square

Scenario 5. If *Eve* tries to forge a winning player, she will fail.

Proof. In the prize claim phase, the lottery winner must submit his/her digest i_j , random number r_{P_j} and β_j (where $\beta_j = H(r_{P_j})$) to proof his/her identity. If *Eve* uses the fake random number r'_{P_j} to claim the prize, then LO can perceive the attempt via the following equation:

$$\beta_j \stackrel{?}{=} H(r'_{P_j}). \tag{58}$$

On the other hand, if *Eve* wants to impersonate a winning player, she must find the r_{P_j} . In fact, based on the secure one way hash function, it is computationally infeasible to obtain r_{P_j} from β_j .

4.4. Correctness. The players can verify the public information via the bulletin board by themselves.

Scenario 6. The one suspecting questions

- (1) the correctness of the player who bet numbers number j ,
- (2) the correctness of the value of final hash chain chain_n ,
- (3) the correctness of popular numbers Num.

Proof. The one suspecting can use the published bulletin board information to verify the number j , Num, and chain_n as

- (1) the players can check whether the bet numbers are equal to the public information number j ;
- (2) they can recalculate all bet numbers of players to determine whether the popular numbers Num is equal to the recalculated value;
- (3) finally, the players can verify the validity of the value of final hash chain chain_n by using the public function hash chain as follows:

$$\begin{aligned}
 & \text{Initial condition } \text{chain}_0 = 0 \\
 & \text{chain}_1 = H(\text{chain}_0, \beta_1) \\
 & \text{chain}_2 = H(\text{chain}_1, \beta_2) \\
 & \vdots \\
 & \text{chain}_n = H(\text{chain}_{n-1}, \beta_n), \tag{59}
 \end{aligned}$$

where n is the number of the sold lottery tickets so far. \square

4.5. *Anonymity.* Including the TP and LO, no one can identify the player from the lottery.

Scenario 7. If *Eve* tries to distinguish between messages digest i_j and real identity of player, she will fail.

Proof. In the registration phase, the players submit their personal information to the PKG and then PKG generates a message digest with personal information as $i_j = H(\text{players' personal information})$.

The message digest is a well-known cryptographic assumption: the secure one way hash function has properties such that given a message m , it is easy to compute $H(m)$. On the other hand, it is computationally infeasible to obtain m from $H(m)$. And given $H(m)$, it is infeasible to find m' to let $H(m) = H(m')$. Hence *Eve* cannot find the real identity of the player from i_j . \square

4.6. *Convenience.* Players are able to purchase lottery tickets if they can access the Internet. Clearly, the proposed joint e-lottery mechanism can achieve this requirement as indicated in the players betting for lottery numbers phase.

4.7. *Without Preregistration.* Players need not register at any lottery organizations in advance. In our scheme, the players need not register at any lottery organizations except for the PKG. In fact, if the players want to join other ID-based applications, the players still need to register to PKG for any PKI applications.

4.8. *No Online Trusted Third Party.* The proposed joint e-lottery mechanism does not require an online TTP.

In our scheme, no online TTP is used to participate in all of the transaction scenarios. Therefore, this requirement is completed in our scheme.

4.9. *Participants' Legality.* The scenario of the proposed joint e-lottery mechanism should ensure participants' legality.

Scenario 8. Suppose that players suspect the legality of the TP and LO.

Proof. In the lottery issuing phase, upon the players receiving lotteries from the LO, players can use the multisignature of lotteries to confirm the legality of TP and LO by (46).

If the equation holds, the participants' legality can be authenticated.

That is, only the legitimate private key is able to sign the valid signature. From another viewpoint, the PKG uses its private key d_{PKG} to generate d_{LO} in the registration phase; if anyone attempts to forge d_{LO} he/she must solve the RSA public-key cryptosystem to acquire the private key d_{PKG} . In fact, it is an integer factorization problem [19]. \square

Scenario 9. Suppose that the players, TP, or LO suspect the legality of player.

Proof. Anyonesuspecting can authenticate the player's legality by verifying the signature $\sigma_{i_j} = (s_{P_j}, t_{P_j})$ by (9).

If the equation holds, the j th player's legality can be authenticated; the derivation of the verification is shown as follows:

$$\begin{aligned}
 & i_j \cdot t_{P_j}^{H(t_{P_j}, i_j)} \pmod{N_{PKG}} \\
 &= d_j^{e_{PKG}} \cdot t_{P_j}^{H(t_{P_j}, i_j)} \pmod{N_{PKG}} \\
 &= d_j^{e_{PKG}} \cdot (r_{P_j}^{e_{PKG}})^{H(t_{P_j}, i_j)} \pmod{N_{PKG}} \tag{60} \\
 &= (d_j \cdot r_{P_j})^{e_{PKG} \cdot H(t_{P_j}, i_j)} \pmod{N_{PKG}} \\
 &= s_{P_j}^{e_{PKG}} \pmod{N_{PKG}}.
 \end{aligned}$$

\square

From the above derivation of the verification, only the legitimate private key d_j is able to sign the valid signature. On the other hand, the player is only able to sign the valid signature if he/she registers with the PKG as a legal participant and acquires the private key d_j .

4.10. *Support Joint and Individual E-Lottery Service.* The protocol can support joint and individual e-lottery service, respectively. In our proposed scheme, we propose two purchase models to satisfy the requirements. Hence, two purchase models have the same rights and protections making our proposed scheme more practical and attractive.

4.11. *Discussions.* Our scheme focuses on proposing a secure and fair joint e-lottery, despite requiring more communication, more data transfer, and a higher computational complexity. We compare the functional properties between related works and ours in Table 2.

Table 2 shows that our scheme achieves the two new functional properties in comparison with related works [4–6]: participant's legality and supporting joint and individual e-lottery service.

In addition, we compare mechanisms with the existed lottery websites [8, 9] and ours in Table 3. Basically, [8, 9] only support a lottery agent. So, the player should register with the TP; this differs from ours.

Table 3 shows that our scheme adopted the ID-based multisignature to verify the legality of all participants while existing lottery websites lack effective mechanisms to achieve this requirement. On the other hand, the existing websites do not have remedial measures to prevent malicious behaviors by the lottery agent or players; for instance, the lottery agent refuses to give out the prize, a malicious player forges a picture to claim a prize, or the purchased lottery of a player is lost when the lottery agent's database crashes. Our scheme uses the ID-based multisignature to provide nonrepudiation evidence to prevent the above situations.

5. Conclusions

In this paper, we present a novel joint e-lottery protocol using the multisignature and verifiable random function. Having

TABLE 2: Comparisons between related works and ours.

	Chow et al.'s [4]	Lee and Chang's [5]	Lee et al.'s [6]	Ours
Security	Yes	Yes	Yes	Yes
Correctness	Yes	Yes	Yes	Yes
Anonymity	Yes	Yes	Yes	Yes
Random generation	Yes	Yes	Yes	Yes
Public verification	Yes	Yes	Yes	Yes
Fairness	Yes	Yes	Yes	Yes
Convenience	Yes	Yes	Yes	Yes
No online trusted third party	Yes	Yes	Yes	Yes
No pre-registration required	Yes	Yes	Yes	Yes
Participants legality	No	No	No	Yes
Support joint and individual e-lottery service	Support individual only	Support individual only	Support individual only	Yes

TABLE 3: Comparisons with the existing e-lottery websites.

	E-Lottery Syndicates [8]	Myleto [9]	Ours
Support joint and individual e-lottery service	Support individual only	Support joint only	Support joint and individual
Player should register with the TP	Yes	Yes	No
Allows players to verify legality of lottery agent	Absence of verification mechanisms	Absence of verification mechanisms	Adopts ID-based multi-signature
Allows players and lottery agent to verify the legality of other players	Absence of verification mechanisms	Absence of verification mechanisms	Adopts ID-based signature
The lottery agent refuses to give out the prize	No remedial measures	No remedial measures	Players hold the TP's signature to arbitral request
If a malicious player forges a picture to claim prize	Not easy to identify the legal lottery	Not easy to identify the legal lottery	Prompt identification by digital signature
Non-repudiation evidence	Depend on the scanned copy of the lottery shown on the screen	Depend on the scanned copy of the lottery shown on the screen	PKI digital signature

been proved, the new mechanism can achieve the requirements of general electronic lotteries. The players can increase the probability of winning prizes by using the proposed secure and fair joint e-lottery scheme. Notably, anyone can verify the correctness of winning lotteries and participants' legality simultaneously by verifying the multisignature; this functionality increases the convenience and security when a new participant joins the system. In the future, we are going to integrate the cash flow concept into our system.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the National Science Council, Taiwan, under contract nos. NSC 101-2221-E-324-005-MY2, 101-2221-E-212-006-MY3, and 102-2219-E-212-001.

References

- [1] Mega millions, <http://www.megamillions.com/>.
- [2] 649Lotter, <http://www.649lotter.com/>.
- [3] California State Lottery, <http://www.calottery.com/default.htm>.
- [4] S. S. M. Chow, L. C. K. Hui, S. M. Yiu, and K. P. Chow, "Practical electronic lotteries with offline TTP," *Computer Communications*, vol. 29, no. 15, pp. 2830–2840, 2006.
- [5] J.-S. Lee and C.-C. Chang, "Design of electronic t-out-of-n lotteries on the Internet," *Computer Standards and Interfaces*, vol. 31, no. 2, pp. 395–400, 2009.
- [6] J.-S. Lee, C.-S. Chan, and C.-C. Chang, "Non-iterative privacy preservation for online lotteries," *IET Information Security*, vol. 3, no. 4, pp. 139–147, 2009.
- [7] J. Haigh, "The statistics of lotteries," in *Handbook of Sports and Lottery Markets*, pp. 481–502, 2008.
- [8] e-Lottery Syndicates, <http://www.e-lottery-syndicates.com/>.
- [9] Myleto, <http://www.myleto.cc/>.
- [10] S. Micali, M. Rabin, and S. Vadhan, "Verifiable random functions," in *Proceedings of the IEEE 40th Annual Conference on Foundations of Computer Science*, pp. 120–130, October 1999.

- [11] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions (extended abstract)," in *Proceedings of the IEEE Annual Symposium on Foundations of Computer Science*, pp. 464–479, 1984.
- [12] A. Lysyanskaya, "Unique signatures and verifiable random functions from the DH-DDH separation," in *Proceedings of the Advances in Cryptology (CRYPTO '02)*, vol. 2442 of *Lecture Notes in Computer Science*, pp. 597–612, 2002.
- [13] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the Advances in Cryptology (CRYPTO '85)*, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, 1985.
- [14] L. Harn and J. Ren, "Efficient identity-based RSA multisignatures," *Computers and Security*, vol. 27, no. 1-2, pp. 12–15, 2008.
- [15] P. Sarkar, "Domain extender for collision resistant hash functions: improving upon Merkle-Damgård iteration," *Discrete Applied Mathematics*, vol. 157, no. 5, pp. 1086–1097, 2009.
- [16] C.-L. Chen and M.-H. Liu, "A traceable E-cash transfer system against blackmail via subliminal channel," *Electronic Commerce Research and Applications*, vol. 8, no. 6, pp. 327–333, 2009.
- [17] Internet Engineering Task Force (IETF) Working Group, "RFC 2631 Diffie-Hellman Key Agreement Method," June 1999.
- [18] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.
- [19] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

Research Article

Password-Only Authenticated Three-Party Key Exchange with Provable Security in the Standard Model

Junghyun Nam,¹ Kim-Kwang Raymond Choo,² Junghwan Kim,¹ Hyun-Kyu Kang,¹
Jinsoo Kim,¹ Juryon Paik,³ and Dongho Won³

¹ Department of Computer Engineering, Konkuk University, 268 Chungwondaero, Chungju, Chungcheongbukdo 380-701, Republic of Korea

² Information Assurance Research Group, Advanced Computing Research Centre, University of South Australia, Mawson Lakes, SA 5095, Australia

³ Department of Computer Engineering, Sungkyunkwan University, 2066 Seoburo, Suwon, Gyeonggido 440-746, Republic of Korea

Correspondence should be addressed to Dongho Won; dhwon@security.re.kr

Received 23 January 2014; Accepted 27 February 2014; Published 14 April 2014

Academic Editors: T. Cao, M. Ivanovic, and F. Yu

Copyright © 2014 Junghyun Nam et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Protocols for password-only authenticated key exchange (PAKE) in the three-party setting allow two clients registered with the same authentication server to derive a common secret key from their individual password shared with the server. Existing three-party PAKE protocols were proven secure under the assumption of the existence of random oracles or in a model that does not consider insider attacks. Therefore, these protocols may turn out to be insecure when the random oracle is instantiated with a particular hash function or an insider attack is mounted against the partner client. The contribution of this paper is to present the first three-party PAKE protocol whose security is proven without any idealized assumptions in a model that captures insider attacks. The proof model we use is a variant of the indistinguishability-based model of Bellare, Pointcheval, and Rogaway (2000), which is one of the most widely accepted models for security analysis of password-based key exchange protocols. We demonstrated that our protocol achieves not only the typical indistinguishability-based security of session keys but also the password security against undetectable online dictionary attacks.

1. Introduction

Authenticated key exchange is one of the most fundamental problems in cryptography and network security. In 1992, Bellare and Merritt [1] introduced encrypted key exchange (or EKE) protocols, which allow

arbitrary two parties, who share only a low-entropy password, to establish a common high-entropy secret key (called a session key) over an insecure public network.

Since the work of Bellare and Merritt [1], password-only authenticated key exchange (PAKE) protocols have attracted much greater attention mainly due to the persistent popularity of passwords as a practical (and cheap) authentication method [2]. Since the publication of the first EKE protocol (with only heuristic security arguments), many provably

secure PAKE protocols have been published. Recent examples include the protocol of Katz and Vaikuntanathan [3], which enjoys both round optimality and provable security in the standard model (i.e., without random oracles and ideal ciphers).

A major challenge in designing PAKE protocols is to protect passwords from a *dictionary attack*, in which an adversary enumerates all possible passwords while testing each one against known password verifiers in order to determine the correct one. The design of two-party PAKE protocols secure against dictionary attacks has been extensively studied over the past two decades and is now fairly well understood. However, three-party PAKE protocols have received far less attention and preventing dictionary attacks is more challenging in the three-party setting. Unlike the two-party setting that assumes the same password is shared

between the two parties, the three-party setting assumes that the two parties (commonly known as *clients*) wishing to establish a session key do not share the same password but hold their individual password shared only with a trusted server. This implies that in the three-party setting, a malicious client can attempt to mount an insider dictionary attack against its partner client. Indeed, many published three-party PAKE protocols were subsequently found to be vulnerable to an insider online/offline dictionary attack (e.g., [4–10]).

It is widely regarded that the design of key exchange protocols (including PAKE protocols) is notoriously hard, and conducting security analysis for such protocols is time-consuming and error-prone [11–13]. The many flaws discovered in published protocols have promoted the use of formal models and rigorous security proofs [14–16]. In the provable security paradigm for protocol analysis, a deductive reasoning process is adopted whereby emphasis is placed on a proven reduction from the problem of breaking the protocol to another problem believed to be computationally hard. A complete mathematical proof with respect to cryptographic definitions provides a strong assurance that a protocol is behaving as desired. It is now standard practice for protocol designers to provide security proofs in a well-defined formal model in order to assure protocol implementers about the security properties of protocols.

Over the past decade, we have seen a number of PAKE protocols proposed in the three-party setting [4–8, 17–29]. Many of these published protocols either did not have a proof of security [5, 6, 17, 22–25] or were subsequently found to be flawed [4–10, 12, 23, 24, 27, 30–36]. There are only a handful of provably secure three-party PAKE protocols [4, 7, 8, 21] whose claimed security properties have not been invalidated. However, there are limitations in the security proof of these protocols. For example, the protocols of [7, 8] are proven secure in the random oracle model. Although a proof of security in the random oracle model is definitely better than having no proof, it may not guarantee security in the real world (currently an open question). The protocols of [4, 21] are proven secure in a restricted model where the adversary is not allowed to corrupt protocol participants. Note that a protocol proven secure in such a restricted model cannot guarantee its security against attacks by malicious clients including insider online/offline dictionary attacks. (Readers who are unfamiliar with formal security models are referred to Section 2.1.) Although Yang and Cao [37] proposed a new three-party key exchange protocol that was proven secure in the standard model, the protocol is based on the ElGamal encryption scheme and thus requires a server’s public key as well as clients’ passwords to be preestablished before the protocol is ever executed. We refer the readers to [33, 38–44] for other recently published protocols designed to work in a “hybrid” setting where a cryptographic key is required in addition to passwords.

To the best of our knowledge, there is no published three-party PAKE protocol whose security is proven secure in the standard model that allows an adversary to corrupt protocol participants. In this work, we present the first three-party PAKE protocol that achieves provable security in the standard model against an active adversary with the corruption

capability. We prove the security of session keys for our protocol in the widely accepted indistinguishability-based model of Bellare, Pointcheval, and Rogaway [14]—this model is, probably, one of the most popular proof models in the provable security paradigm for key exchange protocols. However, the indistinguishability-based security of session keys proven in the Bellare-Pointcheval-Rogaway model (and several other standard models) does not imply the security of passwords against undetectable online dictionary attacks, in which each guess on the password is checked undetectably via an online transaction with the server (see Section 2.3 for more details). We address this problem by providing a separate proof of security for the protocol against undetectable online dictionary (UDOD) attacks. This second proof is compact and elegant and does not rely upon idealized assumptions about the cryptographic primitives. Table 1 compares our protocol against other provably secure three-party PAKE protocols in terms of security proofs.

The remainder of this paper is structured as follows. Section 2 describes a formal proof model along with the associated definitions of security. Section 3 presents our proposed three-party PAKE protocol. In Section 4, we prove that the proposed protocol achieves not only the typical indistinguishability-based security of session keys but also the password security against undetectable online dictionary attacks. We conclude the paper in Section 5.

2. Formal Setting

In this section, we

- (1) first describe a security model adapted from the Bellare-Pointcheval-Rogaway 2000 model [14],
- (2) define a typical indistinguishability-based security of session keys, which we call the *SK security*,
- (3) provide a simple and intuitive definition of security against undetectable online dictionary attacks.

2.1. The Security Model

Protocol Participants. Let \mathcal{C} be the set of all clients registered with the trusted authentication server S . Clients $C, C' \in \mathcal{C}$ who are both registered with S may run a three-party PAKE protocol P at any point in time to establish a session key. Let $\mathcal{U} = \mathcal{C} \cup \{S\}$. A party $U \in \mathcal{U}$ may have several instances involved in distinct, possibly concurrent, executions of protocol P . We use Π_U^i to denote the i th instance of party U . A client instance Π_C^i is said to *accept* when it successfully computes its session key sk_C^i in a protocol execution.

Long-Term Keys. Each client $C \in \mathcal{C}$ chooses a password pw_C from a fixed dictionary PW and shares it with the server S via a secure channel. Accordingly, S holds all the passwords $\{pw_C \mid C \in \mathcal{C}\}$. Each password pw_C is used as the long-term secret key of C and S .

TABLE 1: Security proof comparison.

Protocol	Idealized assumption	Adversary capability	Resistance to UDOD attacks [†]
Our protocol	None	Not restricted	Proven
GPAKE [21]	None	Restricted from corrupting parties	No [4]
NGPAKE [4]	None		Not proven
Lin and Hwang [7]	Random oracles	Not restricted	Not proven
Wu et al. [8]	Random oracles	Not restricted	Not proven

[†]Resistance to undetectable online dictionary attacks.

Partnership. The notion of partnership is a key element in defining the security of the protocol. Two instances are *partners* if both participate in a protocol execution and establish a (shared) session key. We define the partnership relations between instances using the notions of session identifiers and partner identifiers (see [45] on the role and the possible construct of session and partner identifiers as a form of partnering mechanism that enables the right session key to be identified in concurrent protocol executions.). A session identifier (sid) is a unique identifier of a protocol session and is defined as a function of the messages transmitted in the protocol session. We use sid_U^i to denote the sid of instance Π_U^i . A partner identifier (pid) is the set of participants of a specific protocol session. Instances should receive as input a pid before they can run the protocol. By pid_U^i , we denote the pid given to instance Π_U^i . Notice that pid_C^i consists of three participants: server S , client C , and another client C' with whom Π_C^i believes it runs the protocol. We say that any two instances Π_C^i and $\Pi_{C'}^j$ are *partners* if (1) both Π_C^i and $\Pi_{C'}^j$ have accepted, (2) $\text{sid}_C^i = \text{sid}_{C'}^j$, and (3) $\text{pid}_C^i = \text{pid}_{C'}^j$.

Adversary. In the model, the probabilistic polynomial-time (PPT) adversary, \mathcal{A} , controls all the communications that take place between parties via a predefined set of oracle queries. For example, the adversary can ask participants to reveal session keys and passwords using *Reveal* and *Corrupt* queries as described below.

- (i) *Execute*($\Pi_C^i, \Pi_{C'}^j, \Pi_S^k$). This query models passive eavesdropping of a protocol execution. It prompts an honest execution of the protocol between the instances Π_C^i , $\Pi_{C'}^j$ and Π_S^k . The transcript of the protocol execution is returned as the output of the query.
- (ii) *Send*(Π_U^i, m). This query models active attacks against the protocol. It sends message m to instance Π_U^i and returns the message that Π_U^i sends out in response to m . A query of the form *Send*($\Pi_C^i, \text{start} : (C, C', S)$) prompts Π_C^i to initiate the protocol with $\text{pid}_C^i = (C, C', S)$.
- (iii) *Reveal*(Π_C^i). This query returns the session key sk_C^i . This query captures the notion of known key security (and it is often reasonable to assume that the adversary will be able to obtain session keys from any session different from the one under attack). Any

client, Π_C^i , upon receiving such a query and if it has accepted and holds some session key, will send this session key back to \mathcal{A} . However, the adversary is not allowed to ask this query if it has already made a *Test* query to the instance Π_C^i or its partner instance (see below for explanation of the *Test* oracle).

- (iv) *Corrupt*(U). This query captures not only the notion of forward secrecy but also unknown key share attacks and insider attacks. The query provides the adversary with U 's password pw_U . Notice that a *Corrupt* query does not result in the release of the session keys since the adversary already has the ability to obtain session keys through *Reveal* queries. If $U = S$ (i.e., the server is corrupted), all clients' passwords stored by the server will be returned.
- (v) *Test*(Π_C^i). This query is the only oracle query that does not correspond to any of the adversary's abilities. If Π_C^i has accepted with some session key and is being asked a *Test*(Π_C^i) query, then depending on a randomly chosen bit b , the adversary is given either the actual session key (when $b = 1$) or a session key drawn randomly from the session key distribution (when $b = 0$). The adversary can access the *Test* oracle as many times as necessary. All the queries to the oracle are answered using the same value of the hidden bit b . Namely, the keys returned by the *Test* oracle are either all real or all random. But, we require that for each different set of partners, the adversary should access the *Test* oracle only once.

We represent the number of queries used by an adversary as an ordered sequence of five nonnegative integers, $Q = (q_{\text{ex}}, q_{\text{se}}, q_{\text{re}}, q_{\text{co}}, q_{\text{te}})$, where the five elements refer to the numbers of queries that the adversary made, respectively, to its *Execute*, *Send*, *Reveal*, *Corrupt*, and *Test* oracles. We call this usage of queries by an adversary the *query complexity* of the adversary.

2.2. Session Key (SK) Security. We now define the basic security, called the SK security, of a 3-party PAKE protocol. As usual, we define the SK security via the notion of *freshness*. Intuitively, a fresh instance is one that holds a session key which should not be known to the adversary \mathcal{A} , and an unfresh instance is one whose session key (or some information about the key) can be known by trivial means. The formal definition of freshness is explained in Definition 1.

Definition 1. An instance Π_C^i is fresh if none of the following occurs: (1) \mathcal{A} queries $\text{Reveal}(\Pi_C^i)$ or $\text{Reveal}(\Pi_{C'}^j)$, where $\Pi_{C'}^j$ is the partner of Π_C^i and (2) \mathcal{A} queries $\text{Corrupt}(U)$, for some $U \in \text{pid}_C^i$, before Π_C^i or its partner $\Pi_{C'}^j$ accepts.

The SK security of a 3-party PAKE protocol P is defined in the context of the following two-stage experiment.

Stage 1. \mathcal{A} makes any oracle queries at will except that:

- (i) \mathcal{A} is not allowed to ask the $\text{Test}(\Pi_C^i)$ query if the instance Π_C^i is unfresh.
- (ii) \mathcal{A} is not allowed to ask the $\text{Reveal}(\Pi_C^i)$ query if it has already made a Test query to Π_C^i or $\Pi_{C'}^j$, where $\Pi_{C'}^j$ is the partner of Π_C^i .

Stage 2. Once \mathcal{A} decides that Phase 1 is over, it outputs a bit b' as a guess on the hidden bit b chosen by the Test oracle. \mathcal{A} is said to succeed if $b = b'$.

Let Succ be the event that \mathcal{A} succeeds in this experiment. Then we define the advantage of \mathcal{A} in breaking the SK security of protocol P as

$$\begin{aligned} \text{Adv}_P^{\text{sk}}(\mathcal{A}) &= 2 \cdot \Pr[\text{Succ}] - 1, \\ \text{Adv}_P^{\text{sk}}(t, Q) &= \max_{\mathcal{A}} \{ \text{Adv}_P^{\text{sk}}(\mathcal{A}) \}, \end{aligned} \quad (1)$$

where the maximum is over all PPT adversaries \mathcal{A} with time complexity at most t and query complexity at most Q .

Definition 2. A 3-party PAKE protocol P is SK-secure if, for any PPT adversary \mathcal{A} asking at most q_{se} Send queries, $\text{Adv}_P^{\text{sk}}(\mathcal{A})$ is only negligibly larger than $c \cdot q_{\text{se}}/|\text{PW}|$, where c is a very small constant (usually around 2 or 4) when compared with $|\text{PW}|$.

2.3. Modelling Undetectable Online Dictionary Attacks. The SK security does not imply security against undetectable online dictionary attacks. In other words, a 3-party PAKE protocol that is not secure against an undetectable online dictionary attack may be rendered SK-secure. To see this, suppose that a 3-party PAKE protocol P is susceptible to undetectable online dictionary attacks whereby an attacker A can find out the password of any registered client B . Then, we can construct an adversary \mathcal{A} who attacks protocol P with advantage 1 as follows.

Corruption. If A is a registered client, \mathcal{A} queries $\text{Corrupt}(A)$ to obtain the password pw_A . Otherwise, \mathcal{A} skips this step.

Undetectable Online Dictionary Attacks. Next, \mathcal{A} runs the protocol P in the same way as A conducts its undetectable online dictionary attacks against client B . Note that \mathcal{A} can perfectly simulate A 's attack by using the disclosed password pw_A and by asking oracle queries appropriately. At the end of this step, \mathcal{A} will obtain the password pw_B as a result of the attacks.

Impersonation. \mathcal{A} then initiates a new protocol session by querying $\text{Send}(\Pi_C^i, \text{start}: (B, C, S))$, where Π_C^i is an unused instance of an uncorrupted client C . \mathcal{A} runs this session as per the protocol specification, but simulating by itself all the actions of B (by using pw_B). At the end of the session, the instance Π_C^i will accept with its session key sk_C^i .

Test. The instance Π_C^i is fresh as (1) no Reveal query has been made on Π_C^i or its partner (which does not exist) and (2) no Corrupt query has been made against any of B, C , and S . Thus, \mathcal{A} may ask the $\text{Test}(\Pi_C^i)$ query. Since \mathcal{A} can compute the session key sk_C^i by itself, it follows that $\Pr_{P, \mathcal{A}}[\text{Succ}] = 1$ and thus $\text{Adv}_P^{\text{sk}}(\mathcal{A}) = 1$.

Since verifying the correctness of a password guess may require more than one Send queries to be asked, \mathcal{A} may have to ask Send queries as many times as $d \cdot |\text{PW}|$, for some integer $d \geq 1$, to correctly determine the password pw_A . Then, even if $\text{Adv}_P^{\text{sk}}(\mathcal{A}) = 1$, the following holds for some $c \geq 1$:

$$\text{Adv}_P^{\text{sk}}(\mathcal{A}) \leq \frac{cd|\text{PW}|}{|\text{PW}|}, \quad (2)$$

and the protocol P is rendered SK-secure by Definition 2.

This result is not surprising since we call a protocol SK-secure if mounting an online dictionary attack by asking Send queries is the best an adversary can do. However, we want to be able to distinguish undetectable online dictionary attacks from detectable online dictionary attacks, and ensure that the best an adversary can do is to mount a detectable online dictionary attack. The following new definitions together provide a simple and intuitive way of capturing security against undetectable online dictionary attacks.

Definition 3. The $\text{Send}(\Pi_S^k, m)$ query models an *online dictionary attack* if both the following are true at the time of the termination of instance Π_S^k : (1) m was not output by a previous Send query asked to an instance of C by which, Π_S^k believes, m was sent and (2) the adversary \mathcal{A} queried neither $\text{Corrupt}(S)$ nor $\text{Corrupt}(C)$.

In Definition 3, the first condition implies that a straightforward delivery of a message between instances is not considered as an online dictionary attack while the second condition implies that, when C' is the (assumed) peer of client C , the adversary \mathcal{A} can corrupt the peer client C' to mount an (insider) online dictionary attack. Note that our definition of an online dictionary attack does not impose any restriction on asking Reveal queries.

Consider the two-stage experiment described in the previous section. Let Undet be the event that in the experiment, a server instance terminates normally when an online dictionary attack was mounted against the instance. We say that the adversary \mathcal{A} succeeds in mounting an undetectable online dictionary attack if the event Undet occurs.

Definition 4. A 3-party PAKE protocol P is secure against an undetectable online dictionary attack if, for any PPT adversary \mathcal{A} asking at most q_{se} Send queries, $\Pr_{P, \mathcal{A}}[\text{Undet}]$ is only negligibly larger than $c \cdot q_{se} / |\text{PW}|$, where c is as in Definition 2.

3. Our Proposed Protocol

As we have earlier claimed, our proposed protocol presented in this section is the first three-party PAKE protocol proven secure in the standard model against an active adversary who has the corruption ability.

3.1. Preliminaries. We begin by reviewing some cryptographic primitives which underlie the security of our protocol.

Decisional Diffie-Hellman (DDH) Assumption. Let \mathbb{G} be a cyclic (multiplicative) group of prime order q . Since the order of \mathbb{G} is prime, all the elements of \mathbb{G} , except 1, are generators of \mathbb{G} . Let g be a random fixed generator of \mathbb{G} and let x, y, z be randomly chosen elements in \mathbb{Z}_q where $z \neq xy$. Informally stated, the DDH problem for \mathbb{G} is to distinguish between the distributions of (g^x, g^y, g^{xy}) and (g^x, g^y, g^z) , and the DDH assumption is said to hold in \mathbb{G} if it is computationally infeasible to solve the DDH problem for \mathbb{G} . More formally, we define the advantage of \mathcal{D} in solving the DDH problem for \mathbb{G} as $\text{Adv}_{\mathbb{G}}^{\text{ddh}}(\mathcal{D}) = |\Pr[\mathcal{D}(\mathbb{G}, g, g^x, g^y, g^{xy}) = 1] - \Pr[\mathcal{D}(\mathbb{G}, g, g^x, g^y, g^z) = 1]|$. We say that the DDH assumption holds in \mathbb{G} if $\text{Adv}_{\mathbb{G}}^{\text{ddh}}(\mathcal{D})$ is negligible for all PPT algorithms \mathcal{D} . We denote by $\text{Adv}_{\mathbb{G}}^{\text{ddh}}(t)$ the maximum value of $\text{Adv}_{\mathbb{G}}^{\text{ddh}}(\mathcal{D})$ over all algorithms \mathcal{D} running in time at most t .

Message Authentication Codes. Let $\Sigma = (\text{Gen}, \text{Mac}, \text{Ver})$ be a message authentication code (MAC) scheme. The key generation algorithm Gen takes as input a security parameter 1^ℓ and outputs a key k chosen uniformly at random from $\{0, 1\}^\ell$. The MAC generation algorithm Mac takes as input a key k and a message m and outputs a MAC (also known as a tag) σ . The MAC verification algorithm Ver takes as input a key k , a message m , and a MAC σ and outputs 1 if σ is valid for m under k or outputs 0 if σ is invalid. Let $\text{Adv}_{\Sigma}^{\text{euf-cma}}(\mathcal{A})$ be the probability that an adversary \mathcal{A} succeeds in breaking the existential unforgeability of Σ under adaptive chosen message attacks. We say that the MAC scheme Σ is secure if $\text{Adv}_{\Sigma}^{\text{euf-cma}}(\mathcal{A})$ is negligible for every PPT adversary \mathcal{A} . We use $\text{Adv}_{\Sigma}^{\text{euf-cma}}(t, q_{\text{mac}}, q_{\text{ver}})$ to denote the maximum value of $\text{Adv}_{\Sigma}^{\text{euf-cma}}(\mathcal{A})$ over all PPT adversaries \mathcal{A} running in time at most t and asking at most q_{mac} and q_{ver} queries to its MAC generation and verification oracles, respectively.

Two-Party PAKE Protocols. Let 2PAKE be a two-party PAKE protocol that outputs session keys distributed in $\{0, 1\}^\ell$. We assume that 2PAKE is SK-secure against an adversary who is given access to all the oracles: Send, Execute, Reveal, Corrupt, and Test. Let $\text{Adv}_{2\text{PAKE}}^{\text{sk}}(\mathcal{A})$ be the

advantage of an adversary \mathcal{A} in breaking the SK security of 2PAKE. We require that, for all PPT adversaries \mathcal{A} making at most q_{se} Send queries, $\text{Adv}_{2\text{PAKE}}^{\text{sk}}(\mathcal{A})$ is only negligibly larger than $q_{se} / |\text{PW}|$. We denote by $\text{ADV}_{2\text{PAKE}}^{\text{sk}}(t, Q)$ the maximum value of $\text{Adv}_{2\text{PAKE}}^{\text{sk}}(\mathcal{A})$ over all PPT adversaries \mathcal{A} with time complexity at most t and query complexity at most Q .

3.2. Protocol Description. Let A and B be two clients who wish to establish a session key, and let S be a trusted server with which A and B have secretly shared their respective passwords pw_A and pw_B . Our protocol proceeds as follows.

Step 1. A and S establish a shared secret key k_{AS} by running the two-party protocol 2PAKE. Likewise, B and S establish a shared secret key k_{BS} .

Step 2. A (resp., B and S) selects a nonce n_A (resp., n_B and n_S) at random from \mathbb{Z}_q and sends $A \parallel n_A$ (resp., $B \parallel n_B$ and $S \parallel n_S$) to the other two parties. All the parties (A , B , and S) define their session identifiers as $\text{sid}_A = \text{sid}_B = \text{sid}_S = A \parallel n_A \parallel B \parallel n_B \parallel S \parallel n_S$.

Step 3. A chooses a random $x \in \mathbb{Z}_q$, computes $X = g^x$ and $\sigma_{AS} = \text{Mac}_{k_{AS}}(A \parallel X \parallel \text{sid}_A)$, and sends $A \parallel X \parallel \sigma_{AS}$ to S . Meanwhile, B chooses a random $y \in \mathbb{Z}_q$, computes $Y = g^y$ and $\sigma_{BS} = \text{Mac}_{k_{BS}}(B \parallel Y \parallel \text{sid}_B)$, and sends $B \parallel Y \parallel \sigma_{BS}$ to S .

Step 4. S checks that $\text{Ver}_{k_{AS}}(A \parallel X \parallel \text{sid}_S, \sigma_{AS}) = 1$ and $\text{Ver}_{k_{BS}}(B \parallel Y \parallel \text{sid}_S, \sigma_{BS}) = 1$. If either of these is untrue, S aborts the protocol. Otherwise, S computes $\sigma_{SA} = \text{Mac}_{k_{AS}}(S \parallel Y \parallel \text{sid}_S)$ and $\sigma_{SB} = \text{Mac}_{k_{BS}}(S \parallel X \parallel \text{sid}_S)$ and sends $S \parallel Y \parallel \sigma_{SA}$ and $S \parallel X \parallel \sigma_{SB}$ to A and B , respectively.

Step 5. A computes the session key $\text{sk} = Y^x$ if $\text{Ver}_{k_{AS}}(S \parallel Y \parallel \text{sid}_A, \sigma_{SA}) = 1$, while B computes the session key $\text{sk} = X^y$ if $\text{Ver}_{k_{BS}}(S \parallel X \parallel \text{sid}_B, \sigma_{SB}) = 1$. A and B abort the protocol if their verification fails.

The operation of this protocol is illustrated in Figure 1. Steps 1 and 2 of the protocol are independent and can be performed in parallel. The session-key computation in the protocol is the same as in the Diffie-Hellman key exchange protocol (i.e., $\text{sk} = g^{xy}$). Hence, it is straightforward to verify the correctness of the protocol.

4. Security Proofs

In this section we prove that our three-party PAKE protocol is SK-secure and is resistant to undetectable online dictionary attacks. The proofs of both properties rely on neither random oracles nor ideal ciphers. Therefore, if 2PAKE is instantiated with a protocol proven secure in the standard model (e.g., [3, 49]), our three-party PAKE protocol would also be provably secure in the standard model. Hereafter, we denote our protocol by 3PAKEsm (“sm” for “standard model”).

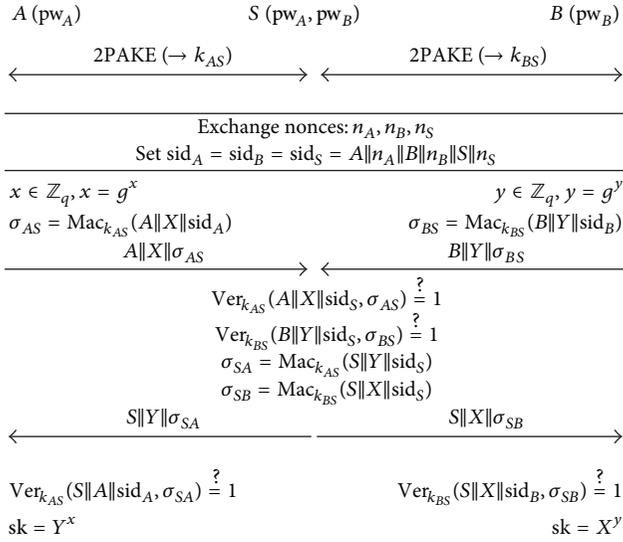


FIGURE 1: The proposed three-party PAKE protocol.

4.1. Proof of SK Security. We first claim that, if the underlying two-party protocol 2PAKE is SK-secure, then the 3PAKEsm protocol is SK-secure as well under the DDH assumption in \mathbb{G} and the security of the MAC scheme Σ .

Theorem 5. Let $Q = (q_{ex}, q_{se}, q_{re}, q_{co}, q_{te})$. For any adversary with query complexity at most Q and time complexity at most t , its advantage in attacking protocol 3PAKEsm is bounded by

$$\begin{aligned} \text{Adv}_{3\text{PAKEsm}}^{\text{sk}}(t, Q) &\leq 2 \cdot \text{Adv}_{2\text{PAKE}}^{\text{sk}}(t', Q') + \frac{(q_{se} + q_{ex})^2}{|\mathbb{G}|} \\ &\quad + 2 \cdot q_{se} \cdot \text{Adv}_{\Sigma}^{\text{uf-cma}}(t', 4, 4) \\ &\quad + 4 \cdot \text{Adv}_{\mathbb{G}}^{\text{ddh}}(t'), \end{aligned} \quad (3)$$

where $Q' = (2q_{ex}, q_{se}, 0, q_{co}, 2q_{ex} + q_{se})$ and t' is the maximum time required to perform an entire experiment involving an adversary who attacks protocol 3PAKEsm with time complexity t .

Proof. Assume an adversary \mathcal{A} attacking protocol 3PAKEsm with time complexity t and query complexity $Q = (q_{ex}, q_{se}, q_{re}, q_{co}, q_{te})$. We prove Theorem 5 by introducing a sequence of experiments $\mathbf{Expr}_0, \dots, \mathbf{Expr}_5$ and bounding the difference in \mathcal{A} 's advantage between two consecutive experiments. \mathbf{Expr}_0 is the original experiment (described in Section 2.2) in which \mathcal{A} attacks the actual protocol, and \mathbf{Expr}_5 is the experiment in which the advantage of \mathcal{A} is 0. Let Succ_i be the event that \mathcal{A} correctly guesses the hidden bit b (chosen by the Test oracle) in experiment \mathbf{Expr}_i . By definition, we get $\text{Adv}_{3\text{PAKEsm}}^{\text{sk}}(\mathcal{A}) = 2 \cdot \Pr[\text{Succ}_0] - 1$.

Before providing details of the proof, we first define the notion of an *uncorrupted* instance. \square

Definition 6. We say an instance Π_U^i is clean if no one in pid_U^i has been asked a Corrupt query. Otherwise, we say it is unclean.

Experiment Expr₁. We modify the experiment so that each different MAC key is chosen uniformly at random from $\{0, 1\}^\ell$ for all clean instances. The difference in \mathcal{A} 's success probability between \mathbf{Expr}_0 and \mathbf{Expr}_1 is bounded by

Claim 1.

$$|\Pr[\text{Succ}_1] - \Pr[\text{Succ}_0]| \leq \text{Adv}_{2\text{PAKE}}^{\text{sk}}(t', Q'). \quad (4)$$

Proof. Assume that the advantage of \mathcal{A} in attacking protocol 3PAKEsm is different between two experiments \mathbf{Expr}_0 and \mathbf{Expr}_1 . Then we prove the claim by constructing, from \mathcal{A} , an adversary $\mathcal{A}_{2\text{PAKE}}$ attacking protocol 2PAKE with time complexity t' and query complexity Q' .

$\mathcal{A}_{2\text{PAKE}}$ begins by choosing a bit b uniformly at random. $\mathcal{A}_{2\text{PAKE}}$ then invokes \mathcal{A} as a subroutine and answers the oracle queries of \mathcal{A} on its own as follows.

Execute Queries. $\mathcal{A}_{2\text{PAKE}}$ answers Execute queries of \mathcal{A} by making Execute and Test queries to its own oracles. Specifically, $\mathcal{A}_{2\text{PAKE}}$ handles each Execute($\Pi_A^i, \Pi_B^j, \Pi_S^k$) query as follows.

- (i) If anyone in $\{A, B, S\}$ has been corrupted, then $\mathcal{A}_{2\text{PAKE}}$ answers the Execute query as in experiment \mathbf{Expr}_0 .
- (ii) Otherwise, $\mathcal{A}_{2\text{PAKE}}$ first makes two queries Execute(Π_A^i, Π_S^k) and Execute(Π_B^j, Π_S^k). Let $T_{2\text{PAKE}}$ and $T'_{2\text{PAKE}}$ be two transcripts returned in response to the Execute queries. Next, $\mathcal{A}_{2\text{PAKE}}$ makes the queries Test(Π_A^i) and Test(Π_B^j) and receives in return two keys \bar{k}_{AS} and \bar{k}_{BS} (either real or random). $\mathcal{A}_{2\text{PAKE}}$ then generates the messages of Steps 2–4 of protocol 3PAKEsm, using \bar{k}_{AS} and \bar{k}_{BS} as the MAC keys. Finally, $\mathcal{A}_{2\text{PAKE}}$ returns these messages prepended by $T_{2\text{PAKE}}$ and $T'_{2\text{PAKE}}$.

Send Queries. At a high level, the simulation of the Send oracle is similar to that of the Execute oracle. Specifically, $\mathcal{A}_{2\text{PAKE}}$ handles each Send(Π_U^i, m) query as follows.

- (i) If the instance Π_U^i is clean or the message m belongs to Step 2 or later steps, then $\mathcal{A}_{2\text{PAKE}}$ answers the query as in experiment \mathbf{Expr}_0 .
- (ii) Otherwise, $\mathcal{A}_{2\text{PAKE}}$ answers it by making the same query to its own Send oracle. If the query causes Π_U^i to accept, then $\mathcal{A}_{2\text{PAKE}}$ also makes a Test(Π_U^i) query (if it had not previously asked a Test query to the partner of Π_U^i). As in the simulation of the Execute oracle, $\mathcal{A}_{2\text{PAKE}}$ uses the output of this Test query as the MAC key in generating the messages of Steps 2–4 of protocol 3PAKEsm.

Reveal Queries. These queries are answered in the obvious way. Namely, \mathcal{A}_{2PAKE} responds to the query $\text{Reveal}(\Pi_C^i)$ by returning the session key sk_C^i .

Corrupt Queries. When \mathcal{A} queries $\text{Corrupt}(U)$, \mathcal{A}_{2PAKE} makes the same query to its own Corrupt oracle and simply forwards the output to \mathcal{A} .

Test Queries. \mathcal{A}_{2PAKE} answers these queries according to the bit b chosen at the beginning of the simulation. That is, \mathcal{A}_{2PAKE} returns real session keys, which it has computed on its own, if $b = 1$, and otherwise returns random keys chosen uniformly at random from \mathbb{G} .

Now at some point in time, when \mathcal{A} terminates and outputs its guess b' , \mathcal{A}_{2PAKE} outputs 1 if $b = b'$ and outputs 0 otherwise.

From the simulation above, it is easy to see that \mathcal{A}_{2PAKE} has at most time complexity t' and query complexity Q' . The advantage of \mathcal{A}_{2PAKE} in attacking protocol 2PAKE is immediate if we notice the following.

- (i) The probability that \mathcal{A}_{2PAKE} outputs 1 when its Test oracle returns real session keys is equal to $\Pr[\text{Succ}_0]$, the probability that \mathcal{A} correctly guesses the bit b in experiment Expr_0 .
- (ii) The probability that \mathcal{A}_{2PAKE} outputs 1 when its Test oracle returns random keys is equal to $\Pr[\text{Succ}_1]$, the probability that \mathcal{A} correctly guesses the bit b in experiment Expr_1 .

This means that $\text{Adv}_{2PAKE}^{sk}(\mathcal{A}_{2PAKE}) = |\Pr[\text{Succ}_1] - \Pr[\text{Succ}_0]|$. Claim 1 then follows. \square

Experiment Expr₂. Let Repeat be the event that a nonce selected by an instance of a party is selected again by another instance of the same party. The experiment Expr_2 is aborted, and the adversary does not succeed, if the event Repeat occurs. This is the only difference between Expr_1 and Expr_2 . By a straightforward calculation, we get the following.

Claim 2.

$$|\Pr[\text{Succ}_2] - \Pr[\text{Succ}_1]| \leq \frac{(q_{se} + q_{ex})^2}{2|\mathbb{G}|}. \quad (5)$$

Experiment Expr₃. Let Forge be the event that the adversary \mathcal{A} makes a Send query of the form $\text{Send}(\Pi_U^i, V \parallel * \parallel \sigma)$ before querying $\text{Corrupt}(W)$, for some $W \in \text{pid}_U^i$, such that (1) σ is a valid tag on $V \parallel * \parallel \text{sid}_U^i$ and (2) no oracle had not previously generated a tag on $V \parallel * \parallel \text{sid}_U^i$. If Forge occurs, this experiment is aborted and the adversary does not succeed. Then we have the following.

Claim 3.

$$|\Pr[\text{Succ}_3] - \Pr[\text{Succ}_2]| \leq q_{se} \cdot \text{Adv}_{\Sigma}^{\text{uf-cma}}(t', 4, 4). \quad (6)$$

Proof. Assuming that the event Forge occurs, we construct, from \mathcal{A} , an algorithm \mathcal{F} who outputs, with a nonnegligible

probability, a forgery against the MAC scheme Σ . The algorithm \mathcal{F} is given oracle access to $\text{Mac}_k(\cdot)$ and $\text{Ver}_k(\cdot)$. The goal of \mathcal{F} is to produce a message/tag pair (m, σ) such that (1) σ is a valid tag on the message m (i.e., $\text{Ver}_k(m, \sigma) = 1$) and (2) \mathcal{F} had not previously queried its oracle $\text{Mac}_k(\cdot)$ on the message m .

Let n be the number of all active sessions that \mathcal{A} initiates by asking a Send query. First, \mathcal{F} chooses a random $\alpha \in \{1, \dots, n\}$. \mathcal{F} then simulates the oracle calls of \mathcal{A} as in experiment Expr_2 ; except that in the α th session, it answers Send queries by accessing its MAC generation and verification oracles. If Forge occurs in the α th session, \mathcal{F} halts and outputs the message/tag pair generated by \mathcal{A} as its forgery. Otherwise, \mathcal{F} halts and outputs a failure indication. This simulation is perfect unless the adversary \mathcal{A} makes a Corrupt query against a participant of the α th session. But note that the event of \mathcal{A} making such a Corrupt query should not happen if Forge occurs in the α th session.

From the simulation, it is immediate that $\text{Adv}_{\Sigma}^{\text{uf-cma}}(\mathcal{F}) = \Pr[\text{Forge}]/n$. Since $n \leq q_{se}$, we get $\Pr[\text{Forge}] \leq q_{se} \cdot \text{Adv}_{\Sigma}^{\text{uf-cma}}(\mathcal{F})$. Then, Claim 3 follows by noticing that \mathcal{F} has at most time complexity t' and makes at most 4 queries to $\text{Mac}_k(\cdot)$ and $\text{Ver}_k(\cdot)$. \square

Experiment Expr₄. This experiment is different from Expr_3 in that the session key sk of each pair of instances partnered via an Execute query is chosen uniformly at random from \mathbb{G} instead of being computed as $sk = g^{xy} = X^y = Y^x$. As the following claim states, the difference in \mathcal{A} 's advantage between Expr_3 and Expr_4 is negligible if the DDH assumption holds in \mathbb{G} .

Claim 4.

$$|\Pr[\text{Succ}_4] - \Pr[\text{Succ}_3]| \leq \text{Adv}_{\mathbb{G}}^{\text{ddh}}(t'). \quad (7)$$

Proof. Assume that the advantage of \mathcal{A} is nonnegligibly different between Expr_3 and Expr_4 . We prove the claim by constructing, from \mathcal{A} , a distinguisher \mathcal{D} that solves the DDH problem in \mathbb{G} . Let $(g_1, g_2, g_3) \in \mathbb{G}^3$ be an instance of the DDH problem given as input to \mathcal{D} . \mathcal{D} begins by choosing a bit b uniformly at random. \mathcal{D} then invokes \mathcal{A} as a subroutine and proceeds to simulate the oracles. \mathcal{D} answers all the oracle queries of \mathcal{A} as in experiment Expr_3 , except that it handles each $\text{Execute}(\Pi_A^i, \Pi_B^j, \Pi_S^k)$ query by

- (1) selecting two random $a_i, b_i \in \mathbb{Z}_q$,
- (2) computing $X' = g_1^{a_i}$ and $Y' = g_2^{b_i}$,
- (3) returning a transcript generated with X' and Y' in place of X and Y ,
- (4) then setting $sk_A^i = sk_B^j = g_3^{a_i b_i}$.

Let b' be the output of \mathcal{A} . \mathcal{D} outputs 1 if $b = b'$ and outputs 0, otherwise.

Then, the following is clear:

- (i) The probability that \mathcal{D} outputs 1 on a true Diffie-Hellman triple is exactly the probability that \mathcal{A} correctly guesses the bit b in experiment Expr_3 .

- (ii) The probability that \mathcal{D} outputs 1 on a random triple is exactly the probability that \mathcal{A} correctly guesses the bit b in experiment **Expr**₄.

This completes the proof of Claim 4. \square

*Experiment Expr*₅. In this experiment, the session key sk_C^i of each instance Π_C^i activated by a **Send** query is chosen uniformly at random from \mathbb{G} if no one in pid_C^i has been corrupted before Π_C^i determines its session identifier sid_C^i . The difference in \mathcal{A} 's advantage between **Expr**₄ and **Expr**₅ is bounded by the following.

Claim 5.

$$|\Pr[\text{Succ}_5] - \Pr[\text{Succ}_4]| \leq \text{Adv}_{\mathbb{G}}^{\text{ddh}}(t'). \quad (8)$$

Proof. The proof of this claim is essentially similar to that of Claim 4. From the adversary \mathcal{A} whose advantage is non-negligibly different between **Expr**₄ and **Expr**₅, we construct a distinguisher \mathcal{D} that solves the DDH problem in \mathbb{G} . Let $(g_1, g_2, g_3) \in \mathbb{G}^3$ be an instance of the DDH problem given as input to \mathcal{D} . \mathcal{D} begins by selecting a bit b uniformly at random and generating a list **DDHList** which is used to link an instance of the DDH problem to a session identifier, \mathcal{D} then runs \mathcal{A} as a subroutine and simulates the oracles. It handles all the queries of \mathcal{A} as in experiment **Expr**₄ except for **Send** queries.

Consider a query of the form **Send**($\Pi_C^i, U \parallel n_U$) which delivers a random nonce n_U to instance Π_C^i . Whenever such a query is made, \mathcal{D} answers it as follows.

- (i) If n_U is not the last nonce that Π_C^i is expected to receive, \mathcal{D} simply waits for the next nonce.
- (ii) Otherwise, \mathcal{D} defines sid_C^i and checks that anyone in pid_C^i was corrupted.
 - (a) If so, \mathcal{D} responds to the query as in experiment **Expr**₄.
 - (b) If not, \mathcal{D} checks if the list **DDHList** contains an entry of the form (sid_C^i, X', Y', Z') , where $X', Y', Z' \in \mathbb{G}$.
 - (1) If it does, \mathcal{D} computes $\sigma_{CS} = \text{Mac}_{k_{CS}}(C \parallel Y' \parallel sid_C^i)$ and returns $C \parallel Y' \parallel \sigma_{CS}$ in response to the query.
 - (2) Otherwise, \mathcal{D} selects two random $a_i, b_i \in \mathbb{Z}_q$, computes $X' = g_1^{a_i}, Y' = g_2^{b_i}, Z' = g_3^{a_i b_i}$, and $\sigma_{CS} = \text{Mac}_{k_{CS}}(C \parallel X' \parallel sid_C^i)$, returns $C \parallel X' \parallel \sigma_{CS}$ to \mathcal{A} , and finally adds the tuple (sid_C^i, X', Y', Z') to **DDHList**.

When \mathcal{A} makes a **Send** query that causes an instance Π_C^i to accept, \mathcal{D} checks if **DDHList** contains an entry of the form (sid_C^i, X', Y', Z') . If so, \mathcal{D} sets $sk_C^i = Z'$. Otherwise, \mathcal{D} computes sk_C^i as in experiment **Expr**₄. For all other **Send** queries of \mathcal{A} , \mathcal{D} answers them as in experiment **Expr**₄. Now

when \mathcal{A} terminates and outputs its guess b' , \mathcal{D} outputs 1 if $b = b'$ and outputs 0 otherwise.

One can easily see the following.

- (i) The probability that \mathcal{D} outputs 1 on a true Diffie-Hellman triple is exactly the probability that \mathcal{A} correctly guesses the bit b in experiment **Expr**₄.
- (ii) The probability that \mathcal{D} outputs 1 on a random triple is exactly the probability that \mathcal{A} correctly guesses the bit b in experiment **Expr**₅.

This implies Claim 5. \square

In experiment **Expr**₅, the session keys of all fresh instances are chosen uniformly at random from \mathbb{G} and thus the adversary \mathcal{A} obtains no information on the bit b chosen by the **Test** oracle. Therefore, it follows that $\Pr[\text{Succ}_5] = 1/2$. This result combined with the previous claims yields the statement of Theorem 5.

4.2. Proof of Resistance to Undetectable Online Dictionary Attacks. We now claim that **3PAKEsm** is secure against undetectable online dictionary attacks as long as the **2PAKE** protocol is SK-secure.

Theorem 7. Let **Undet** be as defined in Section 2.3 and assume that for any PPT adversary \mathcal{A}' asking at most q_{se} **Send** queries, $\text{Adv}_{2\text{PAKE}}^{\text{sk}}(\mathcal{A}')$ is only negligibly larger than $q_{se}/|\text{PW}|$. Then, for any PPT adversary \mathcal{A} asking at most q_{se} **Send** queries, $\Pr_{3\text{PAKEsm}, \mathcal{A}}[\text{Undet}]$ is only negligibly larger than $2 \cdot q_{se}/|\text{PW}|$.

Proof. Let \mathcal{A} be a PPT adversary who asks q_{se} **Send** queries in mounting an undetectable online dictionary attack against **3PAKEsm**. Consider the experiment **Expr**₁ described in the proof of Theorem 5 (see Section 4.1). By **Undet**₁ (resp., **Undet**₀), we denote the event **Undet** defined in experiment **Expr**₁ (resp., **Expr**₀). We prove Theorem 7 by first proving Claim 6 and then Claim 7. \square

Claim 6. $|\Pr_{3\text{PAKEsm}, \mathcal{A}}[\text{Undet}_1] - \Pr_{3\text{PAKEsm}, \mathcal{A}}[\text{Undet}_0]|$ is only negligibly larger than $q_{se}/|\text{PW}|$.

Claim 7. $\Pr_{3\text{PAKEsm}, \mathcal{A}}[\text{Undet}_1]$ is only negligibly larger than $q_{se}/|\text{PW}|$.

Proof of Claim 6. We prove the claim by constructing an adversary \mathcal{A}' who attacks the SK security of **2PAKE** with advantage equal to $|\Pr_{3\text{PAKEsm}, \mathcal{A}}[\text{Undet}_1] - \Pr_{3\text{PAKEsm}, \mathcal{A}}[\text{Undet}_0]|$. \mathcal{A}' chooses a random bit $b \in \{0, 1\}$ and invokes the adversary \mathcal{A} as a subroutine. \mathcal{A}' then simulates the oracles for \mathcal{A} in the exactly same way as in the simulation for the proof of Claim 1. \mathcal{A}' outputs 1 if **Undet** occurs and 0 otherwise. From the way the oracles are simulated, it is easy to see the following.

- (i) The probability that \mathcal{A}' outputs 1 when its **Test** oracle returns real session keys is equal to the probability that the event **Undet** occurs in experiment **Expr**₀.

- (ii) The probability that \mathcal{A}' outputs 1 when its Test oracle returns random keys is equal to the probability that the event Undet occurs in experiment Expr_1 .

Since \mathcal{A}' makes at most q_{se} Send queries, we obtain the statement of Claim 6. \square

Proof of Claim 7. Assume that $\Pr_{3\text{PAKEsm},\mathcal{A}}[\text{Undet}_1]$ is non-negligibly larger than $q_{\text{se}}/|\text{PW}|$. Given the adversary \mathcal{A} , we construct an adversary \mathcal{A}' against 2PAKE who asks at most q_{se} Send queries but has an advantage nonnegligibly larger than $q_{\text{se}}/|\text{PW}|$.

\mathcal{A}' runs \mathcal{A} as a subroutine while simulating the oracles on its own. \mathcal{A}' handles all the oracle queries of \mathcal{A} as in the experiment Expr_1 except for Send queries. When \mathcal{A} makes a $\text{Send}(\Pi_U^i, m)$ query, \mathcal{A}' checks if m is a message for initiating a new session (of 3PAKEsm) or the Send query belongs to an execution of 2PAKE.

- (1) If both are untrue, \mathcal{A}' responds to the query as in experiment Expr_1 .
- (2) Otherwise, \mathcal{A}' answers it by making the same query to its own Send oracle. If the query prompts Π_U^i to accept, then \mathcal{A}' checks if Π_U^i is clean.
 - (a) If so, \mathcal{A}' sets the MAC key of Π_U^i to be a random key drawn uniformly from $\{0, 1\}^\ell$.
 - (b) Otherwise, \mathcal{A}' makes a $\text{Reveal}(\Pi_U^i)$ query and sets the MAC key of Π_U^i to be the output of this Reveal query.

Let Π_S^t be any server instance against which \mathcal{A} has mounted an online dictionary attack. Let k_S^t be the session key (i.e., the MAC key) that the instance Π_S^t has computed in its execution of 2PAKE. In order for the instance Π_S^t to terminate normally, the adversary \mathcal{A} has to make a query of the form $\text{Send}(\Pi_S^t, C \parallel * \parallel \sigma_{CS})$ such that $\text{Ver}_{k_S^t}(C \parallel * \parallel \text{sid}_S^t, \sigma_{CS}) = 1$. When \mathcal{A} makes such a Send query (i.e., when the event Undet_1 occurs), \mathcal{A}' makes a Test query against the instance Π_S^t . Note that the instance Π_S^t is fresh as (1) it is partnered with no instance and (2) S and C must have not been corrupted. Let \bar{k}_S^t be the key returned in response to the Test query. \mathcal{A}' outputs 1 if $\text{Ver}_{\bar{k}_S^t}(C \parallel * \parallel \text{sid}_S^t, \sigma_{CS}) = 1$ and outputs 0, otherwise. If Undet_1 does not occur, \mathcal{A}' outputs a random bit. Then, it is not hard to see that

$$\begin{aligned} \text{Adv}_{2\text{PAKE}}^{\text{sk}}(\mathcal{A}') &= 2 \cdot \Pr_{2\text{PAKE},\mathcal{A}'}[\text{Succ}] - 1 \\ &= 2 \cdot \left(\Pr_{3\text{PAKEsm},\mathcal{A}}[\text{Undet}_1] \right. \\ &\quad \left. + \frac{1}{2} (1 - \Pr_{3\text{PAKEsm},\mathcal{A}}[\text{Undet}_1]) \right) - 1 \\ &= \Pr_{3\text{PAKEsm},\mathcal{A}}[\text{Undet}_1]. \end{aligned} \tag{9}$$

This completes the proof of Claim 7. \square

Theorem 7 immediately follows from Claims 6 and 7.

5. Conclusion

In this work, we have presented a three-party PAKE protocol whose security does not rely on the existence of random oracles. The model that we used to prove the security of our protocol allows the adversary to ask Corrupt queries and thus captures insider attacks as well as forward secrecy. It is a known fact that proving the security of protocols in such a model is of particular importance in the three-party setting as insider dictionary attacks are most serious threats to three-party PAKE protocols. To the best of our knowledge, our protocol is the first three-party PAKE protocol proven secure against insider, active adversaries in the standard model (i.e., without random oracles and ideal ciphers). Another advantage our protocol has over previously published protocols is that it also achieves provable security against undetectable online dictionary attacks. The latter property is also significant as designing three-party PAKE protocol secure against undetectable online dictionary attacks is an ongoing challenge (as evidenced by the number of three-party PAKE protocols found to be vulnerable to an undetectable online dictionary attack). We leave it as a future work to design a three-party PAKE protocol that achieves not only provable security in the standard model but is more efficient than our protocol.

Conflict of Interests

The authors of the paper do not have a direct financial relation with any institution or organization mentioned in the paper that might lead to a conflict of interest for any of the authors.

Acknowledgment

This work was supported by the Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2010-0020210).

References

- [1] S. M. Bellare and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 72–84, Oakland, Calif, USA, May 1992.
- [2] C. Herley and P. van Oorschot, "A research agenda acknowledging the persistence of passwords," *IEEE Security & Privacy*, vol. 10, no. 1, pp. 28–36, 2012.
- [3] J. Katz and V. Vaikuntanathan, "Round-optimal password-based authenticated key exchange," *Journal of Cryptology*, vol. 26, no. 4, pp. 714–743, 2013.
- [4] W. Wang and L. Hu, "Efficient and provably secure generic construction of three-party password-based authenticated key exchange protocols," in *Progress in Cryptology—INDOCRYPT 2006*, vol. 4329 of *Lecture Notes in Computer Science*, pp. 118–132, Springer, Berlin, Germany, 2006.
- [5] H. Guo, Z. Li, Y. Mu, and X. Zhang, "Cryptanalysis of simple three-party key exchange protocol," *Computers & Security*, vol. 27, no. 1-2, pp. 16–21, 2008.

- [6] J. Nam, J. Paik, H.-K. Kang, U. M. Kim, and D. Won, "An off-line dictionary attack on a simple three-party key exchange protocol," *IEEE Communications Letters*, vol. 13, no. 3, pp. 205–207, 2009.
- [7] C.-Y. Lin and T. Hwang, "On 'a simple three-party password-based key exchange protocol,'" *International Journal of Communication Systems*, vol. 24, no. 11, pp. 1520–1532, 2011.
- [8] S. Wu, Q. Pu, S. Wang, and D. He, "Cryptanalysis of a communication-efficient three-party password authenticated key exchange protocol," *Information Sciences*, vol. 215, pp. 83–96, 2012.
- [9] J. Nam, K.-K. R. Choo, J. Paik, and D. Won, "An offline dictionary attack against a three-party key exchange protocol," Tech. Rep. 2013/666, Cryptology ePrint Archive, 2013.
- [10] J. Nam, K.-K. R. Choo, M. Kim, J. Paik, and D. Won, "Dictionary attacks against passwordbased authenticated three-party key exchange protocols," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 12, pp. 3244–3260, 2013.
- [11] K. K. R. Choo, C. Boyd, and Y. Hitchcock, "Errors in computational complexity proofs for protocols," in *Advances in Cryptology—ASIACRYPT 2005*, vol. 3788 of *Lecture Notes in Computer Science*, pp. 624–643, Springer, Berlin, Germany, 2005.
- [12] K.-K. R. Choo, C. Boyd, and Y. Hitchcock, "Examining indistinguishability-based proof models for key establishment protocols," in *Advances in Cryptology—ASIACRYPT 2005*, vol. 3788 of *Lecture Notes in Computer Science*, pp. 585–604, Springer, Berlin, Germany, 2005.
- [13] K.-K. R. Choo, C. Boyd, and Y. Hitchcock, "The importance of proofs of security for key establishment protocols: formal analysis of Jan-Chen, Yang-Shen-Shieh, Kim-Huh-Hwang-Lee, Lin-Sun-Hwang, and Yeh-Sun protocols," *Computer Communications*, vol. 29, no. 15, pp. 2788–2797, 2006.
- [14] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology—EUROCRYPT 2000*, vol. 1807 of *Lecture Notes in Computer Science*, pp. 139–155, Springer, Berlin, Germany, 2000.
- [15] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptology—EUROCRYPT 2001*, vol. 2045 of *Lecture Notes in Computer Science*, pp. 453–474, 2001.
- [16] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Provable Security*, vol. 4784 of *Lecture Notes in Computer Science*, pp. 1–16, Springer, Berlin, Germany, 2007.
- [17] C.-L. Lin, H.-M. Sun, M. Steiner, and T. Hwang, "Three-party encrypted key exchange without server public-keys," *IEEE Communications Letters*, vol. 5, no. 12, pp. 497–499, 2001.
- [18] T.-F. Lee, T. Hwang, and C.-L. Lin, "Enhanced three-party encrypted key exchange without server public keys," *Computers & Security*, vol. 23, no. 7, pp. 571–577, 2004.
- [19] M. Abdalla and D. Pointcheval, "Interactive Diffie-Hellman assumptions with applications to password-based authentication," in *Financial Cryptography and Data Security*, vol. 3570 of *Lecture Notes in Computer Science*, pp. 341–356, Springer, Berlin, Germany, 2005.
- [20] H.-A. Wen, T.-F. Lee, and T. Hwang, "Provably secure three-party password-based authenticated key exchange protocol using Weil pairing," *IEE Proceedings-Communications*, vol. 152, no. 2, pp. 138–143, 2005.
- [21] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," *IEE Proceedings Information Security*, vol. 153, no. 1, pp. 27–39, 2006.
- [22] R. Lu and Z. Cao, "Simple three-party key exchange protocol," *Computers & Security*, vol. 26, no. 1, pp. 94–97, 2007.
- [23] H.-R. Chung and W.-C. Ku, "Three weaknesses in a simple three-party key exchange protocol," *Information Sciences*, vol. 178, no. 1, pp. 220–229, 2008.
- [24] H.-S. Kim and J.-Y. Choi, "Enhanced password-based simple three-party key exchange protocol," *Computers and Electrical Engineering*, vol. 35, no. 1, pp. 107–114, 2009.
- [25] H.-F. Huang, "A simple three-party password-based key exchange protocol," *International Journal of Communication Systems*, vol. 22, no. 7, pp. 857–862, 2009.
- [26] E. Dongna, Q. Cheng, and C. Ma, "Password authenticated key exchange based on RSA in the three-party settings," in *Provable Security*, vol. 5848 of *Lecture Notes in Computer Science*, pp. 168–182, Springer, Berlin, Germany, 2009.
- [27] T.-F. Lee and T. Hwang, "Simple password-based three-party authenticated key exchange without server public keys," *Information Sciences*, vol. 180, no. 9, pp. 1702–1714, 2010.
- [28] W. Wang, L. Hu, and Y. Li, "How to construct secure and efficient three-party password-based authenticated key exchange protocols," in *Information Security and Cryptology*, vol. 6584 of *Lecture Notes in Computer Science*, pp. 218–235, Springer, Berlin, Germany, 2010.
- [29] T.-Y. Chang, M.-S. Hwang, and W.-P. Yang, "A communication-efficient three-party password authenticated key exchange protocol," *Information Sciences*, vol. 181, no. 1, pp. 217–226, 2011.
- [30] J. Nam, Y. Lee, S. Kim, and D. Won, "Security weakness in a three-party pairing-based protocol for password authenticated key exchange," *Information Sciences*, vol. 177, no. 6, pp. 1364–1375, 2007.
- [31] R. C.-W. Phan, W.-C. Yau, and B.-M. Goi, "Cryptanalysis of simple three-party key exchange protocol (S-3PAKE)," *Information Sciences*, vol. 178, no. 13, pp. 2849–2856, 2008.
- [32] E.-J. Yoon and K.-Y. Yoo, "Cryptanalysis of a simple three-party password-based key exchange protocol," *International Journal of Communication Systems*, vol. 24, no. 4, pp. 532–542, 2011.
- [33] H. Liang, J. Hu, and S. Wu, "Re-attack on a three-party password-based authenticated key exchange protocol," *Mathematical and Computer Modelling*, vol. 57, no. 5-6, pp. 1175–1183, 2013.
- [34] H.-T. Tsai and C.-C. Chang, "Provably secure three party encrypted key exchange scheme with explicit authentication," *Information Sciences*, vol. 238, pp. 242–249, 2013.
- [35] J. Nam, K.-K. R. Choo, J. Paik, and D. Won, "On the security of a password-only authenticated three-party key exchange protocol," Tech. Rep. 2013/540, Cryptology ePrint Archive, 2013.
- [36] J. Nam, K.-K. R. Choo, J. Paik, and D. Won, "Two-round password-only authenticated key exchange in the three-party setting," Cryptology ePrint Archive 2014/017, 2014.
- [37] J.-H. Yang and T.-J. Cao, "Provably secure three-party password authenticated key exchange protocol in the standard model," *Journal of Systems and Software*, vol. 85, no. 2, pp. 340–350, 2012.
- [38] K. Yoneyama, "Efficient and strongly secure password-based server aided key exchange," in *Progress in Cryptology—INDOCRYPT 2008*, vol. 5365 of *Lecture Notes in Computer Science*, pp. 172–184, Springer, Berlin, Germany, 2008.
- [39] H.-Y. Chien and T.-C. Wu, "Provably secure password-based three-party key exchange with optimal message steps," *The Computer Journal*, vol. 52, no. 6, pp. 646–655, 2009.

- [40] N. W. Lo and K.-H. Yeh, "Cryptanalysis of two three-party encrypted key exchange protocols," *Computer Standards & Interfaces*, vol. 31, no. 6, pp. 1167–1174, 2009.
- [41] D.-C. Lou and H.-F. Huang, "Efficient three-party password-based key exchange scheme," *International Journal of Communication Systems*, vol. 24, no. 4, pp. 504–512, 2011.
- [42] C. Lee, S. Chen, and C. Chen, "A computation-efficient three-party encrypted key exchange protocol," *Applied Mathematics & Information Sciences*, vol. 6, no. 3, pp. 573–579, 2012.
- [43] J. Zhao and D. Gu, "Provably secure three-party password-based authenticated key exchange protocol," *Information Sciences*, vol. 184, no. 1, pp. 310–323, 2012.
- [44] S. Wu, K. Chen, Q. Pu, and Y. Zhu, "Cryptanalysis and enhancements of efficient three-party password-based key exchange scheme," *International Journal of Communication Systems*, vol. 26, no. 5, pp. 674–686, 2013.
- [45] K.-K. R. Choo, "A proof of revised Yahalom protocol in the Bellare and Rogaway (1993) model," *The Computer Journal*, vol. 50, no. 5, pp. 591–601, 2007.
- [46] W. Diffie, P. C. van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes and Cryptography*, vol. 2, no. 2, pp. 107–125, 1992.
- [47] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*, Springer, Berlin, Germany, 2003.
- [48] B. S. Kaliski, "An unknown key-share attack on the MQV key agreement protocol," *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 275–288, 2001.
- [49] J. Katz, R. Ostrovsky, and M. Yung, "Efficient and secure authenticated key exchange using weak passwords," *Journal of the ACM*, vol. 57, no. 1, article 3, 2009.

Research Article

A Complete Hierarchical Key Management Scheme for Heterogeneous Wireless Sensor Networks

Chien-Ming Chen,^{1,2} Xinying Zheng,¹ and Tsu-Yang Wu^{1,2}

¹ School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China

² Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen 518055, China

Correspondence should be addressed to Tsu-Yang Wu; wutsuyang@gmail.com

Received 28 February 2014; Accepted 26 March 2014; Published 10 April 2014

Academic Editors: T. Cao, M. Ivanovic, and F. Yu

Copyright © 2014 Chien-Ming Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Heterogeneous cluster-based wireless sensor networks (WSN) attracted increasing attention recently. Obviously, the clustering makes the entire networks hierarchical; thus, several kinds of keys are required for hierarchical network topology. However, most existing key management schemes for it place more emphasis on pairwise key management schemes or key predistribution schemes and neglect the property of hierarchy. In this paper, we propose a complete hierarchical key management scheme which only utilizes symmetric cryptographic algorithms and low cost operations for heterogeneous cluster-based WSN. Our scheme considers four kinds of keys, which are an individual key, a cluster key, a master key, and pairwise keys, for each sensor node. Finally, the analysis and experiments demonstrate that the proposed scheme is secure and efficient; thus, it is suitable for heterogeneous cluster-based WSN.

1. Introduction

Recently, wireless sensor networks (WSN) become more and more popular since they have been deployed in various applications, such as military, environmental monitor, industry automation, and smart space. A WSN is composed of a large number of sensor nodes which work together by collaborating with each other. In fact, sensor nodes are constrained in computing, communication, and energy capability; therefore, energy saving and hardware complexity are necessary to be considered carefully when constructing a WSN. For example, asymmetric cryptographic algorithms such as RSA or high cost operations like modular exponentiation operations are not appropriate for designing security mechanisms for a WSN.

Generally, all sensor nodes in a WSN may be divided into several small groups which are known as clusters [1–3]. Each cluster would have a cluster head responsible for collecting and aggregating sensing data from its cluster members. A cluster-based WSN can be implemented in both homogeneous WSN and heterogeneous WSN. We first consider the case that implementing a cluster-based WSN in

homogeneous WSN. Note that all sensor nodes in homogeneous WSN have the same capabilities. After being deployed, every sensor node within the same cluster elects one as a cluster head. Obviously, the workload of acting a cluster head is heavier than a sensor node; as a result, the sensor node which acts as a cluster head would run out of its battery before other sensor nodes. Although it can be solved by rotating the role of the cluster head periodically over all cluster members, the workload of being a cluster head is still heavy for a sensor node. Actually, several studies [4, 5] have demonstrated that a homogeneous ad hoc network has poor performance and scalability.

Hence, several researches have concentrated on a heterogeneous WSN which incorporate different types of sensor nodes with different capabilities. For example, a WSN may contain a small number of powerful high-end sensor nodes (*H*-Sensors) and a large number of low-end sensor nodes (*L*-Sensors). If implementing a cluster-based WSN in heterogeneous WSN, *H*-Sensors organize *L*-Sensors around them into clusters, and *L*-Sensors forward sensing reports to the corresponding *H*-Sensor. It is commonly referred to as the heterogeneous cluster-based WSN. The advantage of

it is the overall hardware cost of the entire WSN that can be reduced. This is because *L*-Sensors, which only perform the basic functions, can be manufactured very cheap and simple.

In another aspect, security is a vital issue in various WSN applications. Thus, an efficient key management scheme is necessary. Depending on different applications and security requirements, a variety of key management schemes have been proposed for a heterogeneous WSN [6–10]. In fact, most of these schemes place more emphasis on pairwise key establishment and key predistribution. However, due to the property of a heterogeneous cluster-based WSN that makes the topology hierarchical, a hierarchical key management scheme for cluster-based heterogeneous WSN is essential. Except for an pairwise key, which is shared between two sensor nodes, an individual key, which is shared between each sensor node and the base station, a common cluster key for each cluster, and a master key for all sensor nodes are also required.

Moreover, key updating is also required to be considered for the following reasons. First, sensor nodes may be compromised. The affected keys must be updated or revoked. Second, the network topology may be dynamic. For example, sensor nodes are deployed into the sea for aquatic application. If a sensor node moves to another neighboring cluster, some keys may need to be updated. Third, the base station may desire to update the master key periodically for better security level. For the best of our knowledge, no hierarchical key management scheme with key updating functionality for a heterogeneous cluster-based WSN has been proposed.

In this paper, we propose a complete and efficient hierarchical key management system for a heterogeneous cluster-based WSN. Our design only utilizes symmetric cryptographic algorithms and low cost operations such as bitwise XOR operation and modular multiplication. This construction contains two schemes, key generating scheme and key updating scheme. In the key generating scheme, four kinds of keys, which are an individual key, pairwise keys, a cluster key, and a master key, are generated for each *L*-Sensor. In the key updating scheme, we place the emphasis on updating cluster keys and the master key. In order to improve better efficiency, our design reduces the usage of unicasting. In the performance evaluation, we demonstrate that the storage requirement, communication, and computation cost are reasonable. Besides, in the security analysis, we show that the proposed construction is secure and influence of compromising can be confined to the affected cluster. Finally, the experimental results demonstrate that the operations used in our design are practical.

The remainder of this paper is organized as follows. In Section 2, some preliminaries are introduced. We describe the related work, network model, attack model, and our design goals. In Section 3, we propose our key generating scheme. In Section 4, the proposed key updating scheme is introduced. Then, Sections 5 and 6 describe the security analysis and performance evaluation of our design, respectively. We further provide experiments in Section 7. Finally, Section 8 concludes.

2. Preliminaries

In this section, the related work is introduced firstly. Then we describe the network model of heterogeneous cluster-based WSN and the attack model. We also list our design goals.

2.1. Related Work. Since several studies [4, 5] demonstrated that a homogeneous WSN has poor performance and scalability; several recent works investigate a heterogeneous WSN. Duarte-Melo and Liu [11] analyzed the energy consumption and lifetime of a heterogeneous WSN. Girod et al. [12] developed tools to support a heterogeneous WSN and measurement and visualization of operational systems. Du and Lin [13] proposed a differentiated coverage algorithm which can provide different coverage degrees for different areas. Lazos and Poovendran [14] studied the problem of coverage in planar heterogeneous WSNs. Lin et al. [15] proposed an ant colony optimization-based approach to maximize the lifetime of a heterogeneous WSN. Chen et al. [16] proposed a recoverable data aggregation scheme for a heterogeneous cluster-based WSN. As shown above, heterogeneous WSN indeed received lots of attention.

In the view of key management or key distribution schemes, several researches have been proposed [6–9, 17, 18]. Most of these schemes focus on probabilistic key predistribution method. Du et al. [6] presented an asymmetric key management scheme which preloads a large number of keys in each *H*-Sensor while preloading a small number of keys in *L*-Sensors. Later, Hussain et al. [8] also proposed key predistribution scheme, which reduces the storage requirements while maintaining the same security strength. Durresi et al. [7] proposed key predistribution schemes between stationary nodes and nonstationary nodes. Traynor et al. [9] described three keying and trust models for the heterogeneous WSN. Khan et al. [17] presented a key management scheme supporting mobility in a heterogeneous WSN that consists of mobile sensor nodes with few fixed sensor nodes. Shi et al. [18] proposed a resource-efficient authentic key establishment scheme for a heterogeneous WSN.

The above schemes for heterogeneous WSNs put more emphasis on pairwise key distribution and ignore the property of hierarchy. Thus, in this paper, we propose a hierarchical key management construction with the functionality of key updating.

2.2. The Network Model of Heterogeneous Cluster-Based WSN. The network model of a heterogeneous cluster-based WSN contains three components, a base station, a small number of powerful high-end sensor nodes (*H*-Sensors), and a large number of low-end sensor nodes (*L*-Sensors). *H*-Sensors are expected to have more energy than *L*-Sensors. They organize *L*-Sensors into clusters, collect and aggregate sensing data from their cluster members (*L*-Sensors), and send the results to the base station. Besides, an *H*-Sensor is equipped with a tamper-resistant hardware.

On the other hand, *L*-Sensors, which have sensing capability with limited computation, memory, and communication, are small and low-cost devices. Each *L*-Sensor detects a target within its detection range, uses its processing power

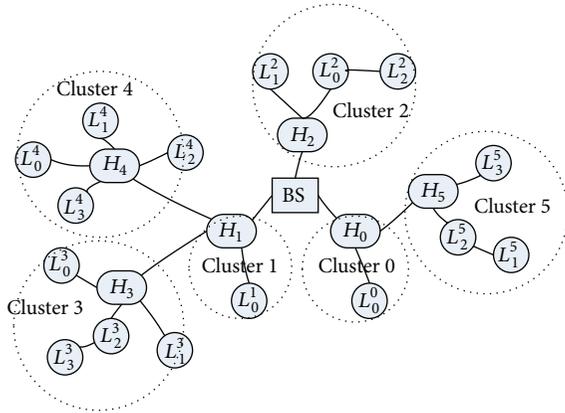


FIGURE 1: An example of a heterogeneous clustered-based WSN.

to locally perform simple computations, and then sends the required data to the corresponding H -Sensor. Besides, an L -Sensor is not equipped with tamper-resistant hardware; therefore, an adversary can obtain the information stored in an L -Sensor after compromising it.

Figure 1 illustrates a simple example of a heterogeneous clustered-based WSN. Note that H_i denotes the H -Sensor i and L_i^j denotes the L -Sensor; i belongs to H_j . Depending on different environments, there might be more than one level of H -Sensors between the base station and L -Sensors. In Figure 1, cluster 3 contains $L_0^3, L_1^3, L_2^3, L_3^3$, and H_3 . H_1 , an upper level H -Sensor of H_3 , forwards the data sent from H_3 to the base station.

Due to the properties of a heterogeneous WSN, the communication capacity of the base station, H -Sensors and L -Sensors are different. The communication can be classified into the following categories.

- (1) Within a cluster: an H -Sensor can broadcast/unicast messages to its cluster member through single hop, whereas messages sent to the H -Sensor may require multihop or still single hop depending on the distance between them.
- (2) Between two neighboring H -Sensors: an H -Sensor can communicate to neighboring H -Sensors through single hop.
- (3) Between the base station and H -Sensors: no doubt, messages sent to the base station require hop by hop. For example, in Figure 1, messages which are sent by H_3 to the base station would pass through H_1 . On the other hand, the base station can broadcast/unicast to all H -Sensors through single hop or multihops.

2.3. Attack Models. Here we discuss the attack model for key management schemes in a WSN. Depending on the abilities of adversary, attacks can be categorized into two situations.

- (1) Without compromising sensor nodes: an adversary can only eavesdrop packets or send false messages to legal sensor nodes without any knowledge.

- (2) Compromising sensor nodes: after compromising a sensor node, an adversary can obtain the information stored in this sensor node. He may calculate other keys or secrets through the compromised information.

Note that detecting compromised sensor nodes which still act as normal sensor nodes is infeasible in all existing detection mechanisms in WSN. However, if an adversary compromises a sensor node and performs abnormal behavior or attacks, it will be detected [19–23]. Besides, in a heterogeneous WSN, it is assumed that every H -Sensor is equipped with a temper-resistant hardware; thus, considering the case that he compromises an H -Sensor is not required.

Based on the above attack models, a key management scheme for a WSN must satisfy the following security requirements.

Requirement 1: all messages, including sensing data and control messages, must be encrypted.

Requirement 2: an adversary cannot compromise the entire WSN with the compromised secrets. More specifically, he cannot derive the secrets that belong to other clusters. Furthermore, if he compromises two sensor nodes which belong to different clusters, he cannot obtain the secrets with other clusters either.

Requirement 3: if compromised sensor nodes are detected, the affected keys must be updated or revoked.

2.4. Design Goals. Four kinds of keys, individual key, master key, cluster key, and pairwise key, are generated in the proposed construction.

- (1) Individual key: every L -Sensor and H -Sensor share an individual key with the base station. The base station can encrypt the secret information with the individual key if required.
- (2) Cluster key: every cluster has one cluster key which is shared with all cluster members including one H -Sensor and several L -Sensors. The cluster key is utilized for encrypting the cluster traffic. An H -Sensor can securely transfer control messages to its cluster members with this key. For example, an H -Sensor may turn some cluster members into sleep mode; it can encrypt this control message with the cluster key. On the other hand, all L -Sensors within the same cluster may encrypt the sensing reports with the same cluster key. The importance of cluster key is also discussed in [24].
- (3) Master key: this master key is shared between all L -Sensors and the base station within the entire network. The base station can securely broadcast the information to all L -Sensors with the master key.
- (4) Pairwise key: the pairwise key is shared with two neighboring L -Sensors. In some applications, two neighboring L -Sensors may require secure channel to protect their communication. For example, if

the cluster key is compromised, L -Sensors in this cluster may encrypt the data with pairwise keys.

In this paper, we also consider how to update some of the above keys efficiently. Sensor nodes may be compromised or move to another neighboring cluster; consequently, the master key and the corresponding cluster keys must be updated. Besides, the master key may be updated periodically for security considerations.

3. The Proposed Key Generating Scheme

In this section, we propose a key generating scheme for a heterogeneous cluster-based WSN. As mentioned above, this scheme generates four kinds of keys, individual key, master key, cluster key, and pairwise key. Notations used in this paper are summarized as shown in Notation section.

3.1. Generating Each Individual Key. Actually, individual keys of each L -Sensor and H -Sensor are preloaded before being deployed. More specifically, an individual key $K_{L_i^j, BS}$ is preloaded to L_i^j and $K_{H_j, BS}$ is preloaded to H_j .

After deployment, H -Sensors partition all L -Sensors into several clusters. Each H -Sensor can realize which L -Sensors are organized in its cluster. Then every H -Sensor reports its cluster information to BS.

Before generating other keys, BS requires to securely assign each L -Sensor a key which is shared with the attached H -Sensor. For example, $K_{L_i^j, H_j}$ which is assigned to L_i^j is shared between L_i^j and H_j . Since BS realizes the cluster information of each cluster, it can securely send the key $K_{L_i^j, H_j}$ encrypted with $K_{H_j, BS}$ or $K_{L_i^j, BS}$ to every H -Sensor and L -Sensor. Hence, all L -Sensors would share the keys with their attached H -Sensors. Note that the key $K_{L_i^j, H_j}$ is transmitted only once before the stages of generating other kinds of keys.

3.2. Generating Cluster Keys. The procedure of generating cluster keys is modified from [25, 26]. To generate the cluster key, each H -Sensor constructs a binary tree and assigns its cluster members to leaf nodes. The root of the binary tree is the cluster key and intermediate nodes are key encryption keys. Note that the key encryption keys are used for updating the cluster key. Each L -Sensor knows all the keys from the parent of its corresponding leaf node up to the root. This set of keys is called the key path. The procedure of generating the cluster key is described as follows.

If there are n_j L -Sensors in cluster j , the H -Sensor H_j will generate random elements $R_0, R_1, \dots, R_{\lceil \log_2(n_j) \rceil - 1}$ and $K_0, K_1, \dots, K_{\lfloor n_j/2 \rfloor - 1}$ to calculate key encryption keys and the cluster key. Both key encryption keys and the cluster key are composed of one random element R_X and several random elements K_Y , where $0 \leq X \leq \lceil \log_2(n_j) \rceil - 1$ and $0 \leq Y \leq \lfloor n_j/2 \rfloor - 1$. R_X means that this key is at level X in the key tree and K_Y means that this key belongs to the key path Y . All keys on the same key path Y have the same random elements K_Y .

The example in Figure 2 shows 8 L -Sensors in cluster 0. H_0 constructs a key tree with 8 leaves and generates random elements $R_0, R_1, R_2, K_0, K_1, K_2$, and K_3 . In Figure 2, $KEK_3^0 (=R_2 \times K_0 \bmod p)$, $KEK_1^0 (=R_2 \times K_0 \times K_1 \bmod p)$, and $CK_0 (=R_0 \times K_1 \times K_2 \times K_3 \bmod p)$ are on the key path 0, KEK_4^0 , KEK_1^0 , and CK_0 are on the key path 1, and so on. KEK_5^0 , KEK_2^0 , and CK_0 belong to key path of L_4^0 and L_5^0 . Note that p is a 128-bit prime number. Since CK_0 is at level 0 in the key tree and belongs to the key path 0, 1, 2, and 3, it is composed of R_0, K_0, K_1, K_2 , and K_3 . After the key tree is constructed, H_0 assigns each cluster member L_i^0 to a leaf node and securely sends the cluster key CK_0 and key encryption keys on the key path to L_i^0 using the key $K_{L_i^0, H_0}$.

3.3. Generating the Master Key. Assume that there are m H -Sensors which are denoted as $\{H_0, H_1, \dots, H_{m-1}\}$ in a WSN. Before generating the master key, BS needs to deliver the secure information $SI_{H_j, BS}$ and $SI_{L_i^j, BS}$ to each H -Sensor and L -Sensor. Note that this secure information is used when generating and updating other keys (see Section 4.2).

BS first chooses a 128-bit prime number p , where p is public, and two secret random numbers $S, W \in Z_p^*$. It also selects m distinct numbers which are denoted as $\{e_0, e_1, \dots, e_{m-1}\}$ from Z_p^* . BS then securely sends the following message to all H -Sensors:

$$\forall 0 \leq j < m, \quad BS \longrightarrow H_j : E_{K_{H_j, BS}} \left(SI_{H_j, BS}, SI_{L_i^j, BS} \right), \quad (1)$$

where $SI_{H_j, BS} = e_j \times W^{-1} \bmod p$ and $SI_{L_i^j, BS} = S \times e_j^{-1} \bmod p$. After receiving it, each H -Sensor, for example, H_j , broadcasts $E_{CK_j} (SI_{L_i^j, BS})$ to all its cluster members. Note that CK_j is the cluster key of cluster j . As a result, all its cluster members can obtain $SI_{L_i^j, BS}$. Note that L -Sensors attached to the same H -Sensor will have the same secure information $SI_{L_i^j, BS}$. Figure 3 shows an example of a WSN with secure information.

After that, BS starts to generate the master key. The procedure of master key generation is described as follows.

Step 1. BS first selects a random number $r \in Z_p^*$ and computes a master key $K_{Master} = S \times r \bmod p$.

Step 2. BS broadcasts $r \times W \bmod p$ to all H -Sensors.

Step 3. When each H -Sensor receives $r \times W \bmod p$ from BS, it calculates the following equations and broadcasts the result to all its cluster members.

For all $0 \leq j \leq m - 1$, H_j calculates

$$\begin{aligned} & \left(\left(SI_{H_j, BS} \right) \times (r \times W \bmod p) \bmod p \right) \oplus CK_j \\ & = \left((e_j \times W^{-1}) \times (r \times W \bmod p) \bmod p \right) \oplus CK_j \quad (2) \\ & = (r \times e_j \bmod p) \oplus CK_j. \end{aligned}$$

We use bitwise XOR operation \oplus to guarantee that messages can be securely sent to legitimate L -Sensors.

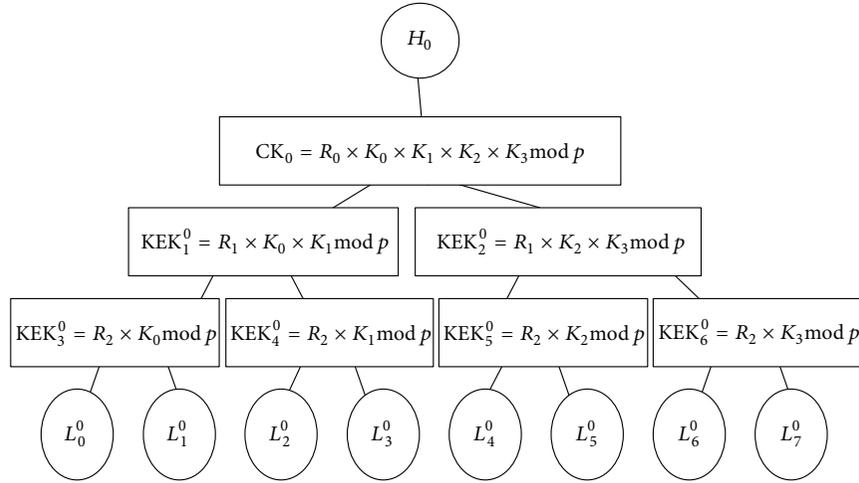


FIGURE 2: An example of a key tree.

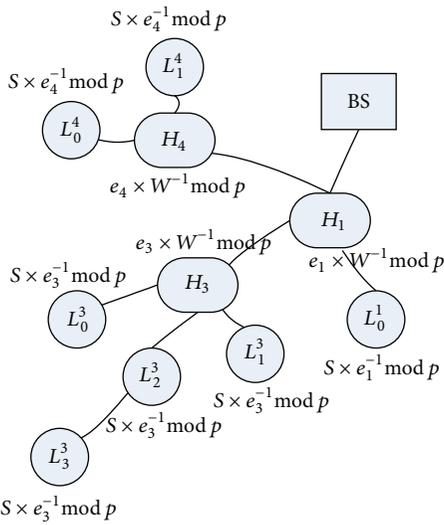


FIGURE 3: An example of the WSN with secure information.

Step 4. When L_i^j receives $(r \times e_j \text{ mod } p) \oplus CK_j$ from H_j , it computes the master key K_{Master} , where

$$\begin{aligned}
 K_{\text{Master}} &= SI_{L_i^j, BS} \times (r \times e_j \text{ mod } p) \\
 &= (S \times e_j^{-1} \text{ mod } p) \times (r \times e_j \text{ mod } p) \quad (3) \\
 &= S \times r \text{ mod } p.
 \end{aligned}$$

By the above steps, all L -Sensors have the same master key K_{Master} .

3.4. *Generating Pairwise Keys.* The pairwise key shared with two neighboring L -Sensors can be generated through their corresponding H -Sensor. For example, if L_0^1 desires to share a pairwise key with L_1^1 , H_1 will generate this pairwise key and send $E_{K_{L_0^1, H_1}}$ (pairwise_key) and $E_{K_{L_1^1, H_1}}$ (pairwise_key) to

L_0^1 and L_1^1 , respectively; consequently, both L_0^1 and L_1^1 can obtain this pairwise key.

4. The Proposed Key Updating Scheme

In this section, we propose the key updating scheme to update some of the generated keys. We discuss these kinds of keys separately as follows.

- (1) Individual key: since it is only shared between BS and each L -Sensor, this key is revoked automatically if an L -Sensor is compromised.
- (2) Cluster key: obviously, if an L -Sensor is compromised, the corresponding cluster requires updating the cluster key. Besides, L -Sensors may move to another neighboring clusters if the deployment of L -Sensors is nonstationary; consequently, both clusters (the original cluster and the target cluster) require updating their cluster key, respectively.
- (3) Master key: similarly, if L -Sensors are compromised, the master key must be updated. Besides, BS may desire to update the master key periodically for better security considerations.
- (4) Pairwise key: for example, assume that two L -Sensors, L_i^n and L_j^n , have shared a pairwise key. If L_i^n is compromised, L_j^n would revoke the shared pairwise key automatically. Besides, these two L -Sensors can also update this key periodically.

4.1. *Updating the Cluster Key.* Here we discuss how to update cluster keys. First, we consider that an L -Sensor leaves a cluster. It is because this L -Sensor is compromised or moves to other neighboring cluster. Second, we consider an L -Sensor joins a cluster.

4.1.1. *An L-Sensor Leaves a Cluster.* In Figure 2, if L_6^0 leaves cluster 0, the keys CK_0 , KEK_2^0 , and KEK_6^0 must be updated. The detailed procedure is described in the following.

Step 1. H_0 selects a random number K_3' from Z_p^* and sends the following key update messages:

$$(i) H_0 \rightarrow \{L_0^0, \dots, L_3^0\} : \text{KEK}_1^0 \oplus ((\text{KEK}_1^0) \times (K_3^{-1} \times K_3' \bmod p) \bmod p),$$

$$(ii) H_0 \rightarrow \{L_4^0, L_5^0\} : \text{KEK}_5^0 \oplus ((\text{KEK}_5^0) \times (K_3^{-1} \times K_3' \bmod p) \bmod p),$$

$$(iii) H_0 \rightarrow \{S_7^0\} : K_{L_7^0, H_0} \oplus ((K_{L_7^0, H_0}) \times (K_3^{-1} \times K_3' \bmod p) \bmod p).$$

Step 2. $S_0^0, S_1^0, S_2^0,$ and S_3^0 can obtain $K_3^{-1} \times K_3' \bmod p$ with KEK_1^0 and then compute the new cluster key CK_0' where

$$\begin{aligned} \text{CK}_0' &= \text{CK}_0 \times (K_3^{-1} \times K_3' \bmod p) \bmod p \\ &= R_0 \times K_0 \times K_1 \times K_2 \times K_3' \bmod p. \end{aligned} \quad (4)$$

Step 3. S_4^0 and S_5^0 can obtain $K_3^{-1} \times K_3' \bmod p$ with KEK_5^0 and then compute the new keys CK_0' and $\text{KEK}_2^{0'}$ where

$$\begin{aligned} \text{KEK}_2^{0'} &= \text{KEK}_2^0 \times (K_3^{-1} \times K_3' \bmod p) \bmod p \\ &= R_1 \times K_2 \times K_3' \bmod p. \end{aligned} \quad (5)$$

Step 4. S_7^0 can obtain $K_3^{-1} \times K_3' \bmod p$ with $K_{L_7^0, H_0}$ and then compute the new keys $\text{CK}_0', \text{KEK}_2^{0'},$ and $\text{KEK}_6^{0'}$ where

$$\begin{aligned} \text{KEK}_6^{0'} &= \text{KEK}_6^0 \times (K_3^{-1} \times K_3' \bmod p) \bmod p \\ &= R_2 \times K_3' \bmod p. \end{aligned} \quad (6)$$

Obviously, it only requires updating the value K_3 in this example. Since L_6^0 has no keys to obtain $K_3^{-1} \times K_3' \bmod p$, it cannot compute the new cluster key and key encryption keys.

4.1.2. An L-Sensor Joins a Cluster. When an L-Sensor joins a new cluster, the target H-Sensor authenticates this joined L-Sensor. This can be accomplished by coordinating with the original H-Sensor. Then BS would deliver a key which will be shared between the target H-Sensor and this L-Sensor. After receiving this key, the H-Sensor would assign this new L-Sensor a leaf node of the key tree. To prevent this new L-Sensor from decrypting the past traffic, all the keys on this key path need to be updated.

Let us take Figure 2 as an illustration. Assume that L_6^0 joins this cluster and has received an individual key $K_{L_6^0, H_0}$. The procedure of updating the cluster key is described as follows.

Step 1. H_0 assigns L_6^0 a leaf node of the key tree. H_0 then selects a random number K_3' from Z_p^* to compute the new keys, $\text{CK}_0', \text{KEK}_2^{0'},$ and $\text{KEK}_6^{0'}$, where

$$\text{CK}_0' = \text{CK}_0 \times (K_3^{-1} \times K_3' \bmod p) \bmod p \quad (7)$$

$$= R_0 \times K_0 \times K_1 \times K_2 \times K_3' \bmod p,$$

$$\text{KEK}_2^{0'} = \text{KEK}_2^0 \times (K_3^{-1} \times K_3' \bmod p) \bmod p \quad (8)$$

$$= R_1 \times K_2 \times K_3' \bmod p,$$

$$\text{KEK}_6^{0'} = \text{KEK}_6^0 \times (K_3^{-1} \times K_3' \bmod p) \bmod p \quad (9)$$

$$= R_2 \times K_3' \bmod p.$$

Step 2. H_0 needs to send the following two messages; one is broadcasted to all L-Sensors within this cluster for updating the key path; one is additionally unicasted to the new L-Sensor L_6^0 :

$$(i) H_0 \rightarrow \{L_0^0, \dots, L_7^0\} : \text{CK}_0 \oplus (K_3^{-1} \times K_3' \bmod p),$$

$$(ii) H_0 \rightarrow L_6^0:$$

$$\begin{aligned} &(K_{L_6^0, H_0} \oplus (K_{L_6^0, H_0} \times \text{CK}_0' \bmod p)) \\ &\parallel (K_{L_6^0, H_0} \oplus (K_{L_6^0, H_0} \times \text{KEK}_2^{0'} \bmod p)) \\ &\parallel (K_{L_6^0, H_0} \oplus (K_{L_6^0, H_0} \times \text{KEK}_6^{0'} \bmod p)). \end{aligned} \quad (10)$$

Since L_6^0 does not have CK_0 , it cannot obtain $K_3^{-1} \times K_3' \bmod p$ to compute the previous keys. On the contrary, all other L-Sensors in this cluster will obtain $K_3^{-1} \times K_3' \bmod p$.

Step 3. L_6^0 computes the new keys $\text{CK}_0', \text{KEK}_2^{0'},$ and $\text{KEK}_6^{0'}$ with $K_{L_6^0, H_0}$.

Step 4. $L_0^0, L_1^0, L_2^0,$ and L_3^0 can compute the new key CK_0' from (7) with the obtained $K_3^{-1} \times K_3' \bmod p$.

Step 5. L_4^0 and L_5^0 can compute the new keys CK_0' and $\text{KEK}_2^{0'}$ from (7) and (8) with the obtained $K_3^{-1} \times K_3' \bmod p$.

Step 6. L_7^0 can compute the new keys $\text{CK}_0', \text{KEK}_2^{0'},$ and $\text{KEK}_6^{0'}$ from (7), (8), and (9) with the obtained $K_3^{-1} \times K_3' \bmod p$.

Similarly, only the value K_3 is required to be updated.

4.1.3. Further Discussion. Notice that in Step 1 of Section 4.1.1, for example, L_0^0 will receive $K_3^{-1} \times K_3' \bmod p$ protected by KEK_1^0 . The reason that we use the equation

$$\text{KEK}_1^0 \oplus ((\text{KEK}_1^0) \times (K_3^{-1} \times K_3' \bmod p) \bmod p) \quad (11)$$

rather than

$$(\text{KEK}_1^0) \times (K_3^{-1} \times K_3' \bmod p) \bmod p \quad (12)$$

is to protect KEK_1^0 . More specifically, if we only use (12), L_4^0 , L_5^9 , and L_7^0 can utilize the obtained $K_3^{-1} \times K_3'$ mod p to further obtain KEK_1^0 . In fact, KEK_1^0 is not revealed to L_4^0 , L_5^0 , and L_7^0 ; therefore, an additional bitwise XOR operation \oplus is required.

Similarly, in Step 2 of Section 4.1.2, the reason we use (10) rather than

$$\begin{aligned} & (K_{L_6^0, H_0} \times CK_0' \text{ mod } p) \parallel (K_{L_6^0, H_0} \times KEK_2^{0'} \text{ mod } p) \\ & \parallel (K_{L_6^0, H_0} \times KEK_6^{0'} \text{ mod } p) \end{aligned} \quad (13)$$

is to protect $K_{L_6^0, H_0}$; otherwise, other cluster members can obtain $K_{L_6^0, H_0}$ with CK_0' , $KEK_2^{0'}$, or $KEK_6^{0'}$.

4.2. Updating the Master Key. Here we discuss how to update the master key in the following situations. First, BS may update the master key periodically to improve the security level. Second, the master key is required to be updated if L -Sensors are compromised.

4.2.1. Updating the Master Key Periodically. The master key may be updated periodically, for example, monthly. The procedure of updating master key is described as follows.

Step 1. BS selects a new random number $r' \in Z_p^*$ and calculates a new master key $K_{\text{Master_new}} = S \times r' \text{ mod } p$. BS broadcasts $r' \times W \text{ mod } p$ to all H -Sensors.

Step 2. While H_j receives $r' \times W \text{ mod } p$, it calculates

$$\begin{aligned} & \left((SI_{H_j, BS}) \times (r' \times W \text{ mod } p) \text{ mod } p \right) \oplus CK_j \\ & = (r' \times e_j \text{ mod } p) \oplus CK_j. \end{aligned} \quad (14)$$

H_j then broadcasts the result to all its cluster members.

Step 3. L_i^j can calculate the new master key $K_{\text{Master_new}}$ where

$$\begin{aligned} K_{\text{Master_new}} &= SI_{L_i^j, BS} \times (r' \times e_j \text{ mod } p) \\ &= (S \times e_j^{-1} \text{ mod } p) \times (r' \times e_j \text{ mod } p) \\ &= S \times r' \text{ mod } p. \end{aligned} \quad (15)$$

In conclusion, all L -Sensors can obtain the new master key $K_{\text{Master_new}}$.

4.2.2. Updating the Master Key If L -Sensors Are Compromised. Assume that L -Sensor L_m^n which is attached to H_n is compromised and L_k^n denotes the L -Sensor k which is also attached to H_n , where $k \neq m$. Besides, there are m H -Sensors which are denoted as $\{H_0, H_1, \dots, H_{m-1}\}$ in the entire WSN. The procedure of updating the master key is described as follows.

Step 1. BS selects a new random number $r' \in Z_p^*$ and calculates a new master key $K_{\text{Master_new}} = S \times r' \text{ mod } p$. BS also selects a random number $e_n' \in Z_p^*$ to compute $e_n^{-1} \times e_n' \text{ mod } p$

and broadcasts $((r' \times W \text{ mod } p) \parallel (E_{K_{H_n, BS}}(e_n^{-1} \times e_n' \text{ mod } p)))$ to all H -Sensors.

Step 2. Except for H_n , each H -Sensor calculates the following equations and broadcasts the result to its cluster members.

For all $0 \leq j \leq m-1, j \neq n$, H_j calculates

$$\begin{aligned} & \left((SI_{H_j, BS}) \times (r' \times W \text{ mod } p) \text{ mod } p \right) \oplus CK_j \\ & = \left((e_j \times W^{-1}) \times (r' \times W \text{ mod } p) \text{ mod } p \right) \oplus CK_j \\ & = (r' \times e_j \text{ mod } p) \oplus CK_j. \end{aligned} \quad (16)$$

Then, all their cluster members can obtain the new master key $K_{\text{Master_new}}$, where

$$\begin{aligned} K_{\text{Master_new}} &= SI_{L_i^j, BS} \times (r' \times e_j \text{ mod } p) \\ &= S \times r' \text{ mod } p. \end{aligned} \quad (17)$$

Step 3. (a) H_n obtains $e_n^{-1} \times e_n' \text{ mod } p$ from $E_{K_{H_n, BS}}(e_n^{-1} \times e_n' \text{ mod } p)$ and further computes the new secure information $SI'_{H_n, BS}$ where

$$\begin{aligned} SI'_{H_n, BS} &= (SI_{H_n, BS}) \times (e_n^{-1} \times e_n' \text{ mod } p) \text{ mod } p \\ &= (e_n \times W^{-1} \text{ mod } p) \times (e_n^{-1} \times e_n' \text{ mod } p) \\ &= e_n' \times W^{-1} \text{ mod } p. \end{aligned} \quad (18)$$

(b) H_n obtains $(r' \times e_n' \text{ mod } p) \oplus CK_n$ from $r' \times W \text{ mod } p$ and new secure information $SI'_{H_n, BS}$. Then H_n broadcasts $((r' \times e_n' \text{ mod } p) \parallel (e_n^{-1} \times e_n' \text{ mod } p)) \oplus CK_n \oplus CK_n'$. Note that CK_n' is the new cluster key of cluster n .

(c) Except for L_m^n , all the other L -Sensors which are also attached to H_n can derive $e_n^{-1} \times e_n' \text{ mod } p$ and $r' \times e_n' \text{ mod } p$ from the obtained message $((r' \times e_n' \text{ mod } p) \parallel (e_n^{-1} \times e_n' \text{ mod } p)) \oplus CK_n \oplus CK_n'$ and then compute $SI'_{L_k, BS}$, where

$$\begin{aligned} SI'_{L_k, BS} &= SI_{L_k, BS} \times (e_n^{-1} \times e_n' \text{ mod } p)^{-1} \\ &= (S \times e_n^{-1} \text{ mod } p) \times (e_n \times e_n'^{-1} \text{ mod } p) \\ &= S \times e_n'^{-1} \text{ mod } p. \end{aligned} \quad (19)$$

(d) After updating $SI'_{L_k, BS}$, all the other L -Sensors except L_m^n within cluster n can compute new master key $K_{\text{Master_new}} = r' \times W \text{ mod } p$ where

$$\begin{aligned} K_{\text{Master_new}} &= SI'_{L_k, BS} \times (r' \times e_n' \text{ mod } p) \\ &= (S \times e_n'^{-1} \text{ mod } p) \times (r' \times e_n' \text{ mod } p) \\ &= S \times r' \text{ mod } p. \end{aligned} \quad (20)$$

Only H_n can obtain $e_n^{-1} \times e_n'$ by decrypting $E_{K_{H_n, BS}}(e_n^{-1} \times e_n' \text{ mod } p)$. Besides, the compromised L -Sensor L_m^n cannot

obtain $r' \times e'_n \bmod p$ and $e_n^{-1} \times e'_n \bmod p$ to compute the new secure information $SI'_{L_k, BS}$ without CK'_n ; therefore, it is unable to obtain the new master key.

Obviously, only the L -Sensors within the affected cluster require updating the secure information before updating the master key; as a result, the influence can be confined locally.

4.2.3. Further Discussion. We have already discussed how to update the master key. It may be questioned why we need to do it in this way. A trivial idea is to let BS generate a random master key (either for master key generation or master key updating). Then BS encrypts it using each cluster key and transmits it to every H -Sensor. This method seems simpler and more straightforward.

The reason is that the proposed scheme attempts to reduce the usage of unicasting. More precisely, in the above method, BS requires to unicast each encrypted master key to corresponding H -Sensor. However, BS only needs to broadcast the same message ($r' \times W \bmod p$) to every H -Sensor in our design. Actually, using broadcast is more efficient than using unicasting. We will demonstrate it through experiments in Section 7.

4.3. Updating the Pairwise Key. Assume that two L -Sensors, L_i^n and L_j^n , have shared a pairwise key. If these two L -Sensors decide to update their shared pairwise key, H_n generates a new pairwise key and sends $E_{K_{L_i^n, H_n}}$ (new_pairwise_key) and $E_{K_{L_j^n, H_n}}$ (new_pairwise_key) to L_i^n and L_j^n , respectively.

5. Security Analysis

In this section, we demonstrate that the proposed construction is secure through the following analyses. We first explain that our design satisfies the following security requirements mentioned in Section 2.3.

Requirement 1: indeed, this requirement is satisfied. All kinds of messages (sensing reports and control messages) will be encrypted with the generated keys. An adversary cannot realize any information without keys.

Requirement 2: if an adversary compromises an L -Sensor which is denoted as L_i^j , he can obtain $K_{\text{master}} = S \times r \bmod p$, $SI'_{L_i^j, BS} = S \times e_j^{-1} \bmod p$, CK_j , and several key encryption keys KEK. However, he is not capable of calculating cluster key and secure information belonging to other clusters. Moreover, the secrets belonging to other clusters cannot be derived even if he compromises two L -Sensors belonging to different clusters.

Requirement 3: actually, the purpose of our key updating scheme can achieve Requirement 3.

Here we analyze the proposed construction in the following aspects.

5.1. The Influence of Compromised L -Sensors. Actually, an adversary can obtain all information which is stored in an

L -Sensor after compromising it. Fortunately, the adversary cannot calculate the information (cluster key and secure information) of other clusters in our design; as a result, the proposed construction ensures that the compromise of L -Sensors does not cause the compromise of the entire network.

To prevent the adversary from decrypting future traffic, the compromised individual key and the pairwise keys are revoked automatically. Besides, BS updates the affected cluster key and the master key. In our design, the adversary cannot obtain this updated information. Moreover, as an example showed in Section 4, only the affected cluster needs to update the secure information before updating the master key. Obviously, the influence of compromising can be confined to be affected cluster.

Besides, if an L -Sensor moves to another cluster, both clusters (original and targeting clusters) would update the cluster keys. It can prevent the L -Sensor from decrypting the traffic of both clusters after it leaves or before it joins. In fact, it can also effectually reduce the influence of compromising. An adversary may eavesdrop and record all packets before compromising L -Sensors. Let us consider the following situations if an L -Sensor L_1^0 moves from cluster 0 to cluster 1. Note that CK_0 and CK_1 are the cluster keys of cluster 0 and cluster 1, respectively.

- (1) CK_0 and CK_1 are updated: CK_0 and CK_1 are updated to CK'_0 and CK'_1 , separately. If L_1^0 which already has moved to cluster 1 is compromised, an adversary would obtain CK'_1 . Hence, he cannot decrypt the messages encrypted with CK_1 . Similarly, if the adversary compromises an L -Sensor within cluster 0 after L_1^0 leaves, he can only decrypt the messages encrypted with CK'_0 .
- (2) CK_0 and CK_1 are not updated: L_1^0 receives CK_1 after arriving to cluster 1. If L_1^0 is compromised, an adversary can decrypt the messages encrypted with CK_1 . More specifically, after compromising L_1^0 , he can retrieve the messages which are sent before L_1^0 joins. Similarly, the same condition happened if he compromises an L -Sensor within cluster 0.

Obviously, the influence of compromising is effectually reduced.

5.2. Confidentiality. An outsider cannot obtain the current cluster keys or master key because the cluster keys and the secure information $S \times e_n^{-1} \bmod p$ which is used for computing the master key are securely distributed to all legitimate L -Sensors. Although a leaving L -Sensor has the old secure information or the old cluster key, this L -Sensor cannot derive the new master key or the new cluster key. To prevent a leaving L -Sensor from obtaining the parameter S or e_n , we choose the length of the prime number p as 128-bit. It is large enough to ensure that deriving correct (x, y) pair from $x \times y \bmod p$ is infeasible. Although a leaving L -Sensor knows the messages $r \times W \bmod p$ and $S \times e_n^{-1} \bmod p$, it cannot compute the current master key $S \times r \bmod p$.

5.3. Integrity. To achieve message authentication and integrity, we can utilize hash algorithms, for example, SHA-1 or MD5, to compute message authentication code (MAC) for a message. For example, in the master key generation phase of the key generating scheme, BS computes the MAC of $r \times W \bmod p$ with K_{Master} , appends it to $r \times W \bmod p$, and then broadcasts it to all H -Sensors. Each H -Sensor H_j broadcasts $r \times e_j \bmod p$ and the received message ($r \times W \bmod p$ and its MAC) to its cluster members. Eventually, each L -Sensor uses the computed master key to recompute the MAC and compares it with the received MAC value. This implies that if an adversary masquerades as BS to send a message, he does not have the master key to compute the corresponding MAC value, and L -Sensors will reject this message.

6. Performance Evaluation

In this section, we show that the proposed construction is efficient in storage, communication, and computation. Besides, it is scalable because the additional overhead of increasing L -Sensors is confined to $\log_2(s)$.

6.1. The Storage Requirement. In the key generating scheme, each L -Sensor requires storing some keys and necessary information. For example, an L -Sensor L_i^j , would store $K_{L_i^j, \text{BS}}$, $K_{L_i^j, H_j}$, $\text{SI}_{L_i^j, \text{BS}}$, a cluster key CK_j , and a common master key K_{Master} . Also, several key encryption keys KEK must be stored to update the cluster key. The number of key encryption keys of L_i^j is about $\lceil \log_2(s) \rceil - 1$ where s is the number of L -Sensors in cluster j . The total storage of these secrets is $128\text{-bit} \times (5 + (\lceil \log_2(s) \rceil - 1))$. Since usually there are at most one hundred L -Sensors within a cluster, total storage is about 176 bytes. Comparing with the current generation of sensor nodes (128 Kbytes in programmable flash memory in MICAz), the storage of the proposed scheme is much less. Besides, as the cluster size grows, the number of keys stored in an L -Sensor increases proportional to the number of L -Sensors within the cluster in an order of $\log_2(s)$. Note that each L -Sensor may also require to store the pairwise keys shared with its neighbors if necessary, but the total storage requirement is still reasonable.

6.2. The Communication Cost. Here we discuss the communication cost of the proposed construction. The communication cost is closely related to two factors. The first one is the message size. Obviously, the number of bits of every message is less than 128-bit. It is reasonable in a WSN. The second one is the transmission types. In fact, using broadcasting is more efficient than using unicasting in a WSN (we will show it in the next section). The majority of transmission in the key generating scheme is using broadcasting. For example, in the master key generation phase, BS broadcasts $r \times W \bmod p$ to all H -Sensors. Similarly, H -Sensors also broadcast the calculating results to its cluster members. The use of unicasting in the proposed scheme is normally involved in the initial stage of some phases. For example, BS unicasts $E_{K_{H_j, \text{BS}}}(SI_{H_j, \text{BS}})$ to all H -Sensors at the initial stage of master key generation phase; thus, this transmission would

be executed only once; even the master key must be updated. Similarly, transmitting $K_{L_i^j, H_j}$ is still executed only once before generating cluster keys and the master key.

In the key updating scheme, every L -Sensor in the affected clusters only receives one message to update the cluster key. In order to avoid using unicasting, we aim to reduce the number of kinds of messages transmitted on the air. As the example shown in Section 4.1, only three kinds of messages are transmitted on the air. Similarly, only two kinds of messages are transmitted in Section 4.1. In the view of updating the master key, it only requires two broadcasts. More specifically, BS broadcasts same information to all H -Sensors, then each H -Sensor broadcasts the calculating result to all cluster members. Only the L -Sensors within the affected cluster require updating the secure information; consequently, the impact can be confined locally. As a result, updating keys incurs less communication overhead and is beneficial for the limited energy of L -Sensors.

6.3. The Computation Cost. The proposed construction utilizes symmetric cryptosystem, such as AES, and modular multiplication. AES is practical and efficient based on previous experimental studies. Another operation, modular multiplication, is always considered as an inefficient operation where the modulus is large, for example, 1024-bit moduli. However, the length of the modulus we adopted is only 128 bits. To demonstrate high effect of the modular multiplication with a 128-bit modulus, we implement it on MICAz sensor nodes. Computation results are given in Section 7.2.

7. Experiments

In our experiments, we choose MICAz sensor nodes. MICAz is capable of ATmega128L microcontroller. The architecture is 8-bit with 8 MHz computation speed. Total programmable memory storage of MICAz sensor is 128 Kbytes. For communication interface, MICAz uses ZigBee (802.15.4) to communicate with other MICAz sensors. Figure 4(a) shows one MICAz sensor. Another device is the base station. Figure 4(b) shows one MICAz sensor plugged on a MIB510 hardware interface, which is the interface of the base station. The MIB510 board is connected to the desktop computer.

7.1. Experiment Assumptions and Design. After deployment, routing paths will be constructed. Normally, a routing path is constructed as a tree structure. In order to simplify this experiment, we assume that all H -Sensors in the routing path have the same degree. Figure 5 illustrates a constructed routing path with degree $d = 2$ and height $h = 4$.

Some researches assume that BS is capable of transmitting data to all H -Sensors through one hop. However, this assumption is not reasonable. In our experiments, we assume that data transmitted to all H -Sensors require multihops. For example, in Figure 5, the message sent to H_6 by BS would pass through H_0 and H_2 . Therefore, if BS desires to update the master key, the following two scenarios may happen using the example shown in Figure 5.

- (1) Using unicasting: BS encrypts the new master key K with each cluster key ($\text{CK}_0, \text{CK}_1, \dots, \text{CK}_{13}$),

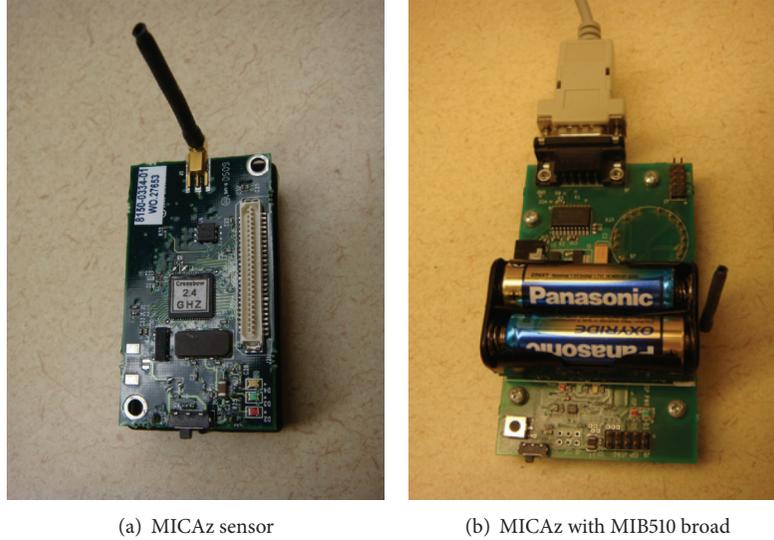
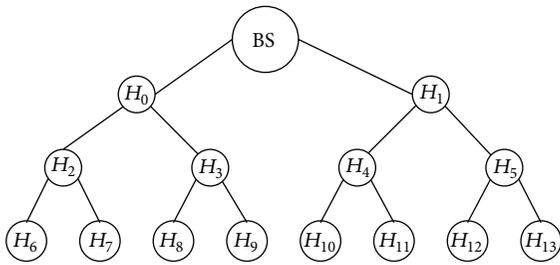


FIGURE 4: (a) Deployed sensor device and (b) base station device.

FIGURE 5: Example of a routing path where $d = 2$ and $h = 4$.

respectively. BS then sends these encrypted master keys to all H -Sensors. Therefore, H_0 receives seven messages which are $E_{CK_0}(K)$, $E_{CK_2}(K)$, $E_{CK_3}(K)$, $E_{CK_6}(K)$, $E_{CK_7}(K)$, $E_{CK_8}(K)$, and $E_{CK_9}(K)$. H_0 obtains $E_{CK_0}(K)$ and then transmits three messages ($E_{CK_2}(K)$, $E_{CK_6}(K)$, and $E_{CK_7}(K)$) to H_2 and transmits the remainder three messages to H_3 . Similarly, H_4 will receive three messages and then transmit two messages. Note that the size of the encrypted master key is 128 bits.

- (2) Using broadcasting: the proposed master key updating method actually uses broadcasting. BS broadcasts a common message to all H -Sensors. Every H -Sensor receives only one message from the upper level H -Sensor and then broadcasts it to lower level H -Sensors. For example, H_2 receives a message from H_0 and then broadcasts it to H_6 and H_7 . Note that the size of the broadcasted message depends on two situations. First, if BS desires to update the master key periodically, the size of the broadcasted message is 128 bits ($r' \times W \bmod p$). Second, BS updates the master key if L -Sensors are compromised. The size of broadcasted message is 256 bits ($((r' \times W \bmod p) \parallel (E_{K_{H_n,BS}}(e_n^{-1} \times e'_n \bmod p))))$.

TABLE 1: Energy consumption of a MICAz sensor node.

	Transmit	Receive
Broadcast 128-bit	14.2 μ J	7.1 μ J
Broadcast 256-bit	16.56 μ J	8.28 μ J
Unicast 128-bit	14.1 μ J	7.05 μ J

Three experiments are performed in this paper.

- (1) Experiment 1: we evaluate the energy consumption of an MICAz sensor node while transmitting or receiving messages with different sizes.
- (2) Experiment 2: we calculate the communication overhead of the entire network. We consider the above two scenarios.
- (3) Experiment 3: we evaluate the cost of AES and modulus multiplication.

In our experiments, we use MICAz sensor nodes to act as H -Sensors and calculate their energy consumption. In fact, energy consumed on powerful devices is the same as one on weak devices, such as MICAz. Besides, energy consumption measurement depends on the number of clock cycles spent [27]. Energy consumption for executing 2090 clock cycles on the ATmega128L microcontroller is equivalent to 7.4 μ J (Joule).

7.2. Experiment Results

Experiment 1. To measure energy consumed on transmitting and receiving data, the number of clock cycles is recorded when a data packet is received and sent. Broadcasting and unicasting executed 10 rounds for different length of data packets. The average results are showed in Table 1. Obviously, the values of broadcast 128-bit and unicast 128-bit are almost equal. Besides, the value of broadcasting 256-bit is larger

TABLE 2: Communication overheads, unit is mJ.

Type	Broadcast 128-bit	Broadcast 256-bit	Unicast 128-bit	Number of nodes	Maximum (uni)	Average (uni)
$d = 2, h = 2$	0.0064	0.0075	0.0852	2	0.0355	0.0426
$d = 2, h = 3$	0.0149	0.0174	0.3551	6	0.0923	0.0592
$d = 2, h = 4$	0.0320	0.0373	1.1221	14	0.2059	0.0802
$d = 2, h = 5$	0.0660	0.0770	3.1105	30	0.4332	0.1037
$d = 2, h = 6$	0.1342	0.1565	7.9964	62	0.8877	0.1290
$d = 2, h = 7$	0.2706	0.3155	19.5862	126	1.7967	0.1554
$d = 2, h = 8$	0.5433	0.6336	46.4019	254	3.6147	0.1827
$d = 2, h = 9$	1.0887	1.2696	107.3052	510	7.2507	0.2104
$d = 2, h = 10$	2.1795	2.5417	243.6559	1022	14.5228	0.2384
$d = 3, h = 2$	0.0852	0.0994	0.2983	3	0.0852	0.0994
$d = 3, h = 3$	0.2769	0.3229	1.7896	12	0.2983	0.1491
$d = 3, h = 4$	0.8520	0.9936	8.1810	39	0.9374	0.2098
$d = 3, h = 5$	2.5773	3.0056	33.1077	120	2.8548	0.2759
$d = 3, h = 6$	7.7532	9.0418	125.1444	363	8.6071	0.3448
$d = 3, h = 7$	23.2809	27.1501	453.0253	1092	25.8640	0.4149

but twice smaller than broadcasting 128-bit. It is because transmitting a 256-bit message still requires one package.

Experiment 2. According to the results in Experiment 1, total energy consumed for a WSN is simulated and evaluated. Table 2 lists the result of communication overhead of the above two scenarios. We consider several routing paths with different degree d and height h . For example, if a routing path is generated as a tree with degree $d = 2$ and height $h = 3$, the total energy consumption of all H -Sensors when broadcasting 128-bit/broadcasting 256-bit/unicasting 128-bit is 0.0149/0.0174/0.3551 mJ, respectively. The number of H -Sensors in this tree is 6. While performing unicasting 128-bit, the maximum energy consumption among all H -Sensors is 0.0923 mJ. Besides, the average energy consumption of all H -Sensors is 0.0592 mJ when performing unicasting 128-bit.

Since sensor nodes may be deployed in a large scale environment, the number of clusters may be up to thousands. Thus, we consider several candidates with different d and h . Obviously, the communication overhead is closely related to the number of H -Sensors. If it grows to hundreds or thousands of nodes, it causes huge energy consumption.

In the view of average energy consumption of an H -Sensor, we can consider the following cases.

- (1) Broadcast 128-bit: in this condition, every node receives one 128-bit message and broadcasts it. The total energy consumption of every node is $21.3 \mu\text{J}$ ($=14.2 + 7.1$).
- (2) Broadcast 256-bit: similarly, the total energy consumption of every node is $24.84 \mu\text{J}$ ($=16.56 + 8.28$).
- (3) Unicast 128-bit: the average energy consumption is listed in Table 2. Obviously, the values are quite larger.

Experiment 3. The goal of this experiment is to evaluate the costs of AES and modulus multiplication. Execution time and

TABLE 3: Time and energy consumption for different operations.

	AES	Modular multiplication
Time (ms)	1.8	3.15
Clock cycle	1658.88	2903.04
Energy (μJ)	5.8735	10.2787

energy consumed by them are recorded. For AES, we choose an AES library based on TinyOS-2.x for comparison. We also implemented modulus multiplication by ourselves. These two operations were executed on MICAz physical sensors for 100 rounds. The average results are given in Table 3. In Table 3, one multiplication over 128-bit modulus is equivalent to 1.75 AES encryptions. As a result, modulus multiplication is feasible on physical sensors.

7.3. Discussion. Through these experiments, we can demonstrate that the proposed key updating scheme is efficient. This is because we utilize broadcasting instead of unicasting. Actually, we must take something into consideration. First, every H -Sensor in the routing path may not have the same degree. Second, BS may have more powerful transmission capability. Let us use Figure 5 as an example. The transmission range of BS may reach the second level H -Sensors. Therefore, the total energy consumption would not equal the result shown in Table 2. However, the overall energy consumption is still high if using unicasting.

8. Conclusion

In this paper, we proposed a complete hierarchical key management construction for heterogeneous cluster-based WSN which only utilizes simple operations. It considered several kinds of keys which are necessary for WSN. Besides, some kinds of keys may require updating; an efficient key

updating scheme is also proposed. In order to provide better efficiency, the majority of transmission in our design is using broadcasting. In fact, using unicasting is inevitable in designing security mechanisms for WSN. Fortunately, the usage of unicasting in our design is normally involved at the initial stage of some phases. In the security analysis, we showed that the influence of compromising is effectually reduced and confined locally. We also showed that the proposed construction is efficient in storage, communication, and computation. Finally, we gave some experiments to further demonstrate two things. First, the operations we used are simple and practical. Second, using unicasting will cause uncontrollable overhead. In conclusion, the proposed construction is appropriate for heterogeneous cluster-based WSN.

Notation

BS :	The base station
H_j :	The H -Sensor j
L_i^j :	The L -Sensor i belongs to H_j
$E_K(M)$:	A message M encrypted with the key K
$K_{L_i^j,BS}$:	The predeployed key shared between L_i^j and BS
$K_{H_j,BS}$:	The predeployed key shared between H_j and BS
$K_{L_i^j,H_j}$:	The key shared between L_i^j and H_j
$SI_{H_j,BS}$:	The secure information shared between H_j and BS
$SI_{L_i^j,BS}$:	The secure information shared between L_i^j and BS
CK_j :	The cluster key for the cluster j
KEK_l^j :	The key encryption key l in the cluster j
K_{Master} :	The master key
\oplus :	Bitwise XOR operation
$\ $:	Concatenation.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work was supported in part by the Project HIT.NSRIF.2014098 by Natural Scientific Research Innovation Foundation in Harbin Institute of Technology, in part by Shenzhen Peacock Project, China, under Contract KQC201109020055A, and in part by Shenzhen Strategic Emerging Industries Program under Grant ZDSY20120613125016389.

References

- [1] H.-C. Shih, J.-H. Ho, B.-Y. Liao, and J.-S. Pan, "Hierarchical gradient diffusion algorithm for wireless sensor networks," in *Recent Trends in Applied Artificial Intelligence*, pp. 480–489, Springer, 2013.
- [2] M. Demirbas, A. Arora, V. Mittal, and V. Kulathumani, "A fault-local self-stabilizing clustering service for wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 9, pp. 912–922, 2006.
- [3] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli, "Localized protocols for ad hoc clustering and backbone formation: a performance comparison," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 4, pp. 292–306, 2006.
- [4] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 1, pp. 3–12, March 2000.
- [5] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.
- [6] X. Du, Y. Xiao, S. Ci, M. Guizani, and H.-H. Chen, "A routing-driven key management scheme for heterogeneous sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 3407–3412, June 2007.
- [7] A. Durresi, V. Bulusu, V. Paruchuri, M. Durresi, and R. Jain, "WSN09-4: key distribution in mobile heterogeneous sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–5, December 2006.
- [8] S. Hussain, F. Kausar, and A. Masood, "An efficient key distribution scheme for heterogeneous sensor networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '07)*, pp. 388–392, August 2007.
- [9] P. Traynor, H. Choi, G. Cao, S. Zhu, and T. La Porta, "Establishing pair-wise keys in heterogeneous sensor networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–12, April 2006.
- [10] P. Traynor, R. Kumar, H. Choi, G. Cao, S. Zhu, and T. La Porta, "Efficient hybrid security mechanisms for heterogeneous sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 663–677, 2007.
- [11] E. Duarte-Melo and M. Liu, "Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 1, pp. 21–25, 2002.
- [12] L. Girod, T. Stathopoulos, N. Ramanathan et al., "A system for simulation, emulation, and deployment of heterogeneous sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 201–213, November 2004.
- [13] X. Du and F. Lin, "Maintaining differentiated coverage in heterogeneous sensor networks," *Eurasip Journal on Wireless Communications and Networking*, vol. 5, no. 4, pp. 565–572, 2005.
- [14] L. Lazos and R. Poovendran, "Stochastic coverage in heterogeneous sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 3, pp. 325–358, 2006.
- [15] Y. Lin, J. Zhang, H. S.-H. Chung, W. H. Ip, Y. Li, and Y.-H. Shi, "An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 3, pp. 408–420, 2012.
- [16] C.-M. Chen, Y.-H. Lin, Y.-C. Lin, and H.-M. Sun, "RCDA: recoverable concealed data aggregation for data integrity in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, pp. 727–734, 2012.
- [17] S. U. Khan, L. Lavagno, and C. Pastrone, "A key management scheme supporting node mobility in heterogeneous sensor networks," in *Proceedings of the 6th International Conference on Emerging Technologies (ICET '10)*, pp. 364–369, October 2010.

- [18] Q. Shi, N. Zhang, M. Merabti, and K. Kifayat, "Resource-efficient authentic key establishment in heterogeneous wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 73, no. 2, pp. 235–249, 2013.
- [19] H.-M. Sun, C.-M. Chen, and Y.-C. Hsiao, "An efficient countermeasure to the selective forwarding attack in wireless sensor networks," in *Proceedings of the IEEE Region 10 Conference (TENCON '07)*, pp. 1–4, November 2007.
- [20] C. M. Chen, Y. H. Lin, Y. H. Chen, and H. M. Sun, "Sashimi: secure aggregation via successively hierarchical inspecting of message integrity on wsn," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 1, pp. 57–72, 2013.
- [21] H.-M. Sun, S.-P. Hsu, and C.-M. Chen, "Mobile Jamming attack and its countermeasure in wireless sensor networks," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia (AINAW '07)*, pp. 457–462, May 2007.
- [22] H.-M. Sun, Y.-H. Lin, Y.-C. Hsiao, and C.-M. Chen, "An efficient and verifiable concealed data aggregation scheme in wireless sensor networks," in *Proceedings of the International Conference on Embedded Software and Systems (ICCESS '08)*, pp. 19–26, July 2008.
- [23] L. Kong, C. M. Chen, H. C. Shih, C. W. Lin, B. Z. He, and J. S. Pan, "An energy-aware routing protocol using cat swarm optimization for wireless sensor networks," in *Advanced Technologies, Embedded and Multimedia for Human-Centric Computing*, pp. 311–318, Springer, 2014.
- [24] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pp. 62–72, October 2003.
- [25] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16–30, 2000.
- [26] R. Di Pietro, L. V. Mancini, Y. W. Law, S. Etalle, and P. Havinga, "LKHW: a directed diffusion-based secure multicast scheme for wireless sensor networks," in *Proceedings of the International Conference on Parallel Processing Workshops (ICPPW '03)*, pp. 397–406, 2003.
- [27] A. S. Wandert, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pp. 324–328, March 2005.

Research Article

Reducing Side Effects of Hiding Sensitive Itemsets in Privacy Preserving Data Mining

Chun-Wei Lin,^{1,2} Tzung-Pei Hong,^{3,4} and Hung-Chuan Hsu³

¹ Innovative Information Industry Research Center (IIIRC), School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China

² Shenzhen Key Laboratory of Internet Information Collaboration, School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China

³ Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan

⁴ Department of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung 804, Taiwan

Correspondence should be addressed to Tzung-Pei Hong; tphong@nuk.edu.tw

Received 16 January 2014; Accepted 26 February 2014; Published 10 April 2014

Academic Editors: T. Cao and M. Ivanovic

Copyright © 2014 Chun-Wei Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Data mining is traditionally adopted to retrieve and analyze knowledge from large amounts of data. Private or confidential data may be sanitized or suppressed before it is shared or published in public. Privacy preserving data mining (PPDM) has thus become an important issue in recent years. The most general way of PPDM is to sanitize the database to hide the sensitive information. In this paper, a novel hiding-missing-artificial utility (HMAU) algorithm is proposed to hide sensitive itemsets through transaction deletion. The transaction with the maximal ratio of sensitive to nonsensitive one is thus selected to be entirely deleted. Three side effects of hiding failures, missing itemsets, and artificial itemsets are considered to evaluate whether the transactions are required to be deleted for hiding sensitive itemsets. Three weights are also assigned as the importance to three factors, which can be set according to the requirement of users. Experiments are then conducted to show the performance of the proposed algorithm in execution time, number of deleted transactions, and number of side effects.

1. Introduction

With the rapid growth of data mining technologies in recent years, useful information can be easily mined to aid managers or decision-makers for making efficient decisions or strategies. The derived knowledge can be simply classified into association rules [1–5], sequential patterns [6–8], classification [9, 10], clustering [11, 12], and utility mining [13–16], among others. Among them, association-rule mining is the most commonly used to determine the relationships of purchased items in large datasets.

Traditional data mining techniques analyze database to find potential relations among items. Some applications require protection against the disclosure of private, confidential, or secure data. Privacy preserving data mining (PPDM) [17] was thus proposed to reduce privacy threats by hiding

sensitive information while allowing required information to be mined from databases. Privacy information includes some personal or confidential information in business, such as social security numbers, home address, credit card numbers, credit ratings, purchasing behavior, and best-selling commodity. In PPDM, data sanitization is generally used to hide sensitive information with the minimal side effects for keeping the original database as authentic as possible. The intuitive way of data sanitization to hide sensitive information is directly to delete sensitive information from amounts of data. Three side effects of hiding failure, missing cost, and artificial cost are then generated in data sanitization process but most approaches are designed to partially evaluate the side effects. Infrequent itemset is, however, not considered in the evaluation process, thus raising the probability of artificial itemsets caused. Besides, the differences between

the minimum support threshold and the frequencies of the itemsets to be hidden are not considered in the above approaches.

In this paper, a hiding-missing-artificial utility (HMAU) algorithm is proposed for evaluating the processed transactions to determine whether they are required to be deleted for hiding sensitive itemsets by considering three dimensions as hiding failure dimension (HFD), missing itemset dimension (MID), and artificial itemset dimension (AID). The weight of each dimension in evaluation process can be adjusted by users. Experimental results showed that the proposed HMAU algorithm has good performance in execution time and the number of deleted transactions. Besides, the proposed algorithm can thus generate minimal side effects of three factors compared to the past algorithm for transaction deletion to hide the sensitive itemsets.

This paper is organized as follows. Some related works are reviewed in Section 2, including the data mining techniques, the privacy preserving data mining, and the evaluated criteria of PPDM. The proposed HMAU algorithm to hide the sensitive itemsets for transaction deletion is stated in Section 3. An illustrated example of the proposed HMAU algorithm is given in Section 4 step by step. Experiments are conducted in Section 5. Conclusion and future works are mentioned Section 6.

2. Review of Related Works

In this section, privacy preserving data mining (PPDM) techniques and evaluated criteria of PPDM are respectively reviewed.

2.1. Privacy Preserving Data Mining Techniques. Data mining is used to extract useful rules from large amounts of data. Agrawal and Srikant proposed Apriori algorithm to mine association rules in two phases to firstly generate the frequent itemsets and secondly derive the association rules [3]. Han et al. then proposed the Frequent-Pattern-tree (FP-tree) structure for efficiently mining association rules without generation of candidate itemsets [18]. The FP-tree was used to compress a database into a tree structure which stored only large items. It was condensed and complete for finding all the frequent patterns. The construction process was executed tuple by tuple, from the first transaction to the last one. After that, a recursive mining procedure called FP-Growth was executed to derive frequent patterns from the FP-tree.

Through various data mining techniques, information can thus be efficiently discovered. The misuse of these techniques may, however, lead to privacy concerns and security problems. Privacy preserving data mining (PPDM) has thus become a critical issue for hiding private, confidential, or secure information. Most commonly, the original database is sanitized for hiding sensitive information [19–21].

In data sanitization, it is intuitive to directly delete sensitive data for hiding sensitive information. Leary found that data mining techniques can pose security and privacy threats [22]. Amiri proposed the aggregate, disaggregate, and hybrid approaches to, respectively, determine whether

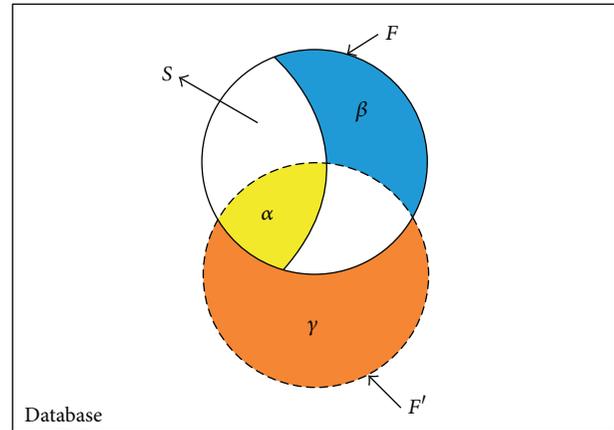


FIGURE 1: Relationship between the side effects and mined rules of the original database and sanitized one.

the transactions or the items are to be deleted for hiding sensitive information [23]. The approaches considered the ratio of sensitive itemsets to nonsensitive frequent itemsets to evaluate the side effects of hiding failures and missing itemsets. Oliveira and Zaiane designed the sliding window algorithm (SWA) [24], in which the victim item with the highest frequency in the sensitive rules related to the current sensitive transaction is selected. Victim items are removed from the sensitive transaction until the disclosure threshold equals 0. Hong et al. proposed a lattice-based algorithm to hide the sensitive information through itemset deletion by a lattice structure to speed up the sanitization process [25]. All the sensitive itemsets are firstly used to build the lattice structure. The sensitive itemsets are then gradually deleted bottom-up from the lowest levels to the highest ones until the frequencies of the sensitive itemsets are lower than the minimum support threshold. Different strategies for hiding sensitive itemsets are still designed in progress to find better results considering of side effects and the dissimilarity of database [21, 26–30].

2.2. Evaluation Criteria. In data sanitization, the primary goal is to hide the sensitive information with minimal influences on databases. Three side effects of hiding failures, missing itemsets, and artificial itemsets are used to evaluate the performance of data sanitization. for data distortion [28, 31, 32] of sensitive itemsets in PPDM. The relationships between the side effects and mined itemsets of the original database and sanitized one are shown in Figure 1.

In Figure 1, F represents the frequent itemsets mined from the original database, F' represents the frequent itemsets mined from the sanitized database, and S represents the sensitive itemsets that should be hidden. The α part is concerned as hiding failures that fail to hide the sensitive itemsets. Thus, α is the intersection of S and F' ($= S \cap F'$). β part is concerned as missing itemsets that mistakenly to delete the nonsensitive frequent rules. Thus, β is the difference between F , S , and F' ($= F - S - F'$). γ part is concerned as artificial itemsets which is unexpectedly generated. Thus, γ is

the difference between F' and $F (= F' - F)$. In PPDM, it is intuitive to delete transactions with sensitive itemsets in the sanitization process. In this paper, α , β , and γ with adjustable weights are considered to evaluate whether the processed transactions are required to be deleted. Besides the above side effects, the number of deleted transactions or items is also a criterion to evaluate the data distortion [32, 33].

3. Proposed Hiding-Missing-Artificial Utility Algorithm

3.1. Definition of Formulas. Data sanitization is the most common way to protect sensitive knowledge from disclosure in PPDM. To avoid the side effects of hiding failures, missing itemsets, and artificial itemsets, minimal distortion of the databases is thus necessary. In this paper, a hiding-missing-artificial utility (HMAU) algorithm is proposed to hide sensitive itemsets through transaction deletion. Three dimensions of hiding failure dimension (HFD), missing itemset dimension (MID), and artificial itemset dimension (AID) are thus concerned to evaluate whether the transactions are required to be deleted for hiding the sensitive itemsets. The transactions with any of the sensitive itemset are first evaluated by the designed algorithm to find the minimal HMAU values among transactions, The transaction with minimal HMAU value will be directly removed from the database. The procedure is thus repeated until all sensitive itemsets are hidden. In order to avoid exposing the already hidden sensitive itemsets again, the minimum count is dynamically updated during the deletion procedure.

The value of each dimension is set from 0 to 1 ($0 < \text{value} \leq 1$). In the proposed formulas, the differences between minimum support threshold and the frequencies of the sensitive itemsets are thus considered to evaluate whether the transactions are required to be deleted instead of only the presence of the itemsets in the transactions.

First, the HFD is used to evaluate the hiding failures of each processed transaction in the sanitization process. When a processed transaction T_k contains a sensitive itemset hs_x , the HFD value of the processed transaction is calculated as

$$HFD^k (hs_x) = \frac{MAX_{HS} - \text{freq}(hs_x) + 1}{MAX_{HS} - [|D| \times \lambda] + 1}, \quad (1)$$

where λ is defined as the percentage of the minimum support threshold, sensitive itemset hs_x is from the set of sensitive itemsets HS, MAX_{HS} is the maximal count of the sensitive itemsets in the set of sensitive itemsets HS, $|D|$ is the number of transactions in the original database D , and $\text{freq}(hs_x)$ is the occurrence frequency of the sensitive itemset hs_x .

Second, the MID is used to evaluate the itemsets of each processed transaction in the sanitization process. When a processed transaction T_k contains a frequent itemset fi_x , the MID value of the processed transaction is calculated as

$$MID^k (fi_x) = \frac{MAX_{FI} - \text{freq}(fi_x) + 1}{MAX_{FI} - [|D| \times \lambda] + 1}, \quad (2)$$

where an itemset fi_x is a frequent itemset from the set of large (frequent) itemsets FI, MAX_{FI} is the maximal count of the

large itemsets in the set of FI, and $\text{freq}(fi_x)$ is the occurrence frequency of the large itemset fi_x .

Third, the AID is used to evaluate the artificial itemsets of each processed transaction in the sanitization process. In AID, only the small 1-itemsets are considered in the sanitization process since it is a nontrivial task to keep all infrequent itemsets. When a processed transaction T_k contains a small 1-itemset si_x , the AID value of the processed transaction is calculated as

$$AID^k (si_x) = \frac{\text{freq}(si_x) - MIN_{SI^1} + 1}{[|D| \times \lambda] - MIN_{SI^1}}, \quad (3)$$

where a small 1-itemset si_x is from the set of small 1-itemsets SI^1 , MIN_{SI^1} is the minimal count of the small 1-itemsets in the set of SI^1 , and $\text{freq}(si_x)$ is the occurrence frequency of the small 1-itemset si_x .

In this paper, a risky bound is designed to speed up the execution time of the proposed HMAU algorithm by avoiding the evaluation of all large itemsets and small 1-itemsets by considering MID and AID. A parameter μ is set as the percentage used to find the upper and lower boundaries of the minimum support threshold. Only the large itemsets and infrequent 1-itemsets within the boundaries are used to determine whether the processed transactions are required to be deleted. For the large itemsets, the minimum support threshold is set as the lower boundary, and the upper boundary is set as

$$\text{freq}(fi_j) \leq [[|D| \times \lambda] \times (1 + \mu)], \quad (4)$$

where $|D|$ is the number of transactions in the original database D , λ is the minimum support threshold, μ is the risky bound, and $\text{freq}(fi_j)$ is the occurrence frequency of the large itemset fi_j .

For small 1-itemsets, the minimum support threshold is set as the upper boundary, and the lower boundary is set as

$$\text{freq}(si_a) \geq [[|D| \times \lambda] \times (1 - \mu)], \quad (5)$$

where $\text{freq}(si_a)$ is the occurrence frequency of the small 1-itemset si_a .

The flowchart of the proposed HMAU algorithm is depicted in Figure 2.

3.2. Notation. See Table 1.

Details of the proposed HMAU algorithm are illustrated as follows.

Proposed HMAU Algorithm.

Input. This includes an original database D , a minimum support threshold ratio λ , a risky bound μ , a set of large (frequent) itemsets $FI = \{fi_1, fi_2, \dots, fi_p\}$, a set of small

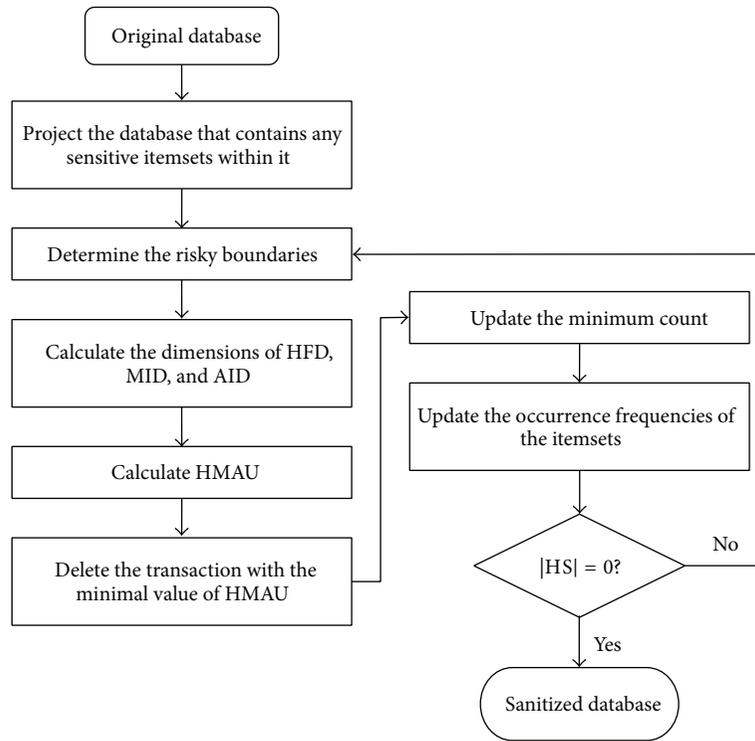


FIGURE 2: Flowchart of the proposed HMAU algorithm.

(nonfrequent) 1-itemsets $SI^1 = \{si_1, si_2, \dots, si_q\}$, and a set of sensitive itemsets to be hidden $HS = \{hs_1, hs_2, \dots, hs_r\}$.

Output. This includes a sanitized database D^* with no sensitive information.

Step 1. Select the transactions to form a projected database D' , where each transaction T_k in D' consists of sensitive itemsets hs_i within it, where $1 \leq i \leq r$.

Step 2. Process each frequent itemset fi_j in the set of FI to determine whether its frequency satisfies the condition $\text{freq}(fi_j) \leq \lceil \lceil |D| \times \lambda \rceil \times (1 + \mu) \rceil$, where $|D|$ is the number of transactions in the original database D and $\text{freq}(fi_j)$ is the occurrence frequency of the large itemset fi_j . Put the fi_j that do not satisfy the condition into the set of FI_{tmp}^1 .

Step 3. Process each small 1-itemset si_a in the set of SI^1 to determine whether its frequency satisfies the condition $\text{freq}(si_a) \geq \lceil \lceil |D| \times \lambda \rceil \times (1 - \mu) \rceil$, where $\text{freq}(si_a)$ is the occurrence frequency of the small 1-itemset si_a . Put the si_a that do not satisfy the condition into the set of SI_{tmp}^1 .

Step 4. Calculate the maximal count (MAX_{HS}) of the sensitive itemsets hs_i in the set of HS as

$$MAX_{HS} = \max \{ \text{freq}(hs_i), \forall hs_i, 1 \leq i \leq r \}, \quad (6)$$

where $\text{freq}(hs_i)$ is the occurrence frequency of the sensitive itemset hs_i in the set of HS .

Step 5. Calculate the HFD of each transaction T_k . Do the following substeps.

Substep 5.1. Calculate the HFD of each sensitive itemset hs_i within T_k as

$$HFD^k(hs_i) = \frac{MAX_{HS} - \text{freq}(hs_i) + 1}{MAX_{HS} - \lceil |D| \times \lambda \rceil + 1}. \quad (7)$$

Substep 5.2. Sum the HFDs of sensitive itemsets hs_i within T_k as

$$HFD^k = \frac{1}{\sum_{i=1}^r HFD^k(hs_i) + 1}. \quad (8)$$

Substep 5.3. Normalize the HFD^k for all transactions T_k in D' .

Step 6. Calculate the maximal count (MAX_{FI}) of the large itemsets fi_j in the set of FI as

$$MAX_{FI} = \max \{ \text{freq}(fi_j), \forall fi_j, 1 \leq j \leq p \}. \quad (9)$$

TABLE 1: The notations used in the proposed HMAU algorithm are described below.

D	An original database, $D = \{T_1, T_2, \dots, T_x, \dots, T_n\}$, in which each T_x represents a transaction
D'	A projected database, $D' = \{T_1, T_2, \dots, T_y, \dots, T_k\}$, in which each T_y contains sensitive itemsets
D^*	A sanitized database, from which no sensitive information can be mined
λ	The minimum support threshold ratio
μ	The risky bound parameter, using which the itemsets within the boundary are used to evaluate the processed transactions
FI	The set of frequent itemsets $FI = \{f_{i_1}, f_{i_2}, \dots, f_{i_j}, \dots, f_{i_p}\}$ in D , in which each itemset f_{i_j} is larger than or equal to the minimum support threshold
f_{i_j}	A frequent itemset
SI^1	The set of infrequent 1-itemsets $SI^1 = \{s_{i_1}, s_{i_2}, \dots, s_{i_a}, \dots, s_{i_q}\}$ in D , in which each itemset s_{i_a} is below the minimum support threshold
s_{i_a}	A small (infrequent) 1-itemset
HS	A set of sensitive itemsets $HS = \{hs_1, hs_2, \dots, hs_i, \dots, hs_r\}$, in which each element represents an itemset that should be hidden in the original database
hs_i	A sensitive itemset
HS_{tmp}	The temporary set of sensitive itemsets outside the boundary
FI_{tmp}	The temporary set of large itemsets outside the boundary
SI^1_{tmp}	The temporary set of small 1-itemsets outside the boundary
HFD	The hiding failure dimension used to consider the side effects of hiding failures
MID	The missing itemset dimension used to consider the side effects of missing itemsets
AID	The artificial itemset dimension used to consider the side effects of artificial itemsets
$HFD^k(hs_i)$	The value of the sensitive itemset hs_i in transaction T_k
$MID^k(f_{i_j})$	The value of the large itemset f_{i_j} in transaction T_k
$AID^k(s_{i_a})$	The value of the small 1-itemset s_{i_a} in transaction T_k
MAX_{HS}	The maximal count of the sensitive itemsets in the set of HS
$freq(hs_i)$	The occurrence frequency of the sensitive itemset hs_i in the set of HS
MAX_{FI}	The maximal count of the large itemsets in the set of FI
$freq(f_{i_j})$	The occurrence frequency of the large itemset f_{i_j}
MIN_{SI^1}	The minimal count of the small 1-itemsets in the set of SI^1
$freq(s_{i_a})$	The occurrence frequency of the small 1-itemset s_{i_a}
w_b	The weights for HFD, MID, and AID, in which $0 < w_b \leq 1$
HMAU	The utility value used to determine whether the processed transactions should be deleted

Step 7. Calculate the MID of each transaction T_k . Do the following substeps.

Substep 7.1. Calculate the MID of each large itemset within T_k as

$$MID^k(f_{i_j}) = \frac{MAX_{FI} - freq(f_{i_j}) + 1}{MAX_{FI} - \lceil |D| \times \lambda \rceil + 1}. \quad (10)$$

Substep 7.2. Sum the MIDs of large itemsets f_{i_j} within T_k as

$$MID^k = \sum_{j=1}^p MID^k(f_{i_j}). \quad (11)$$

Substep 7.3. Normalize the MID^k for all transactions T_k in D' .

Step 8. Calculate the minimal count (MIN_{SI^1}) of the small 1-itemsets s_{i_a} in the set of SI^1 as

$$MIN_{SI^1} = \min \{freq(s_{i_a}), \forall s_{i_a}, 1 \leq a \leq q\}. \quad (12)$$

Step 9. Calculate the AID of each transaction T_k . Do the following substeps.

Substep 9.1. Calculate the AID of each small 1-itemset within T_k as

$$AID^k(s_{i_a}) = \frac{freq(s_{i_a}) - MIN_{SI^1} + 1}{\lceil |D| \times \lambda \rceil - MIN_{SI^1}}. \quad (13)$$

Substep 9.2. Sum the AIDs of small 1-itemsets s_{i_a} within T_k as

$$AID^k = \frac{1}{\sum_{a=1}^q AID^k(s_{i_a}) + 1}. \quad (14)$$

TABLE 2: Original database.

TID	Item
T_1	a, b, c, e
T_2	e
T_3	b, c, e, f
T_4	d, f
T_5	a, b, d
T_6	b, c, e
T_7	a, b, c, d, e
T_8	a, b, e
T_9	c, e
T_{10}	a, b, c, e

Substep 9.3. Normalize the AID^k for all transactions T_k in D' .

Step 10. Calculate the HMAU for HFD, MID, and AID of each transaction T_k as

$$HMAU^k = w_1 \times HFD^k + w_2 \times MID^k + w_3 \times AID^k \quad (15)$$

where $w_1, w_2,$ and w_3 are the predefined weights by users.

Step 11. Remove transaction T_k with $\min\{HMAU^k, \forall T_k, 1 \leq k \leq |D'|\}$ value.

Step 12. Update the minimum count ($= \lceil |D| \times \lambda \rceil$) of sanitized database.

Step 13. Update the occurrence frequencies of all sensitive itemsets in the sets of HS and HS_{tmp} . Put hs_i into the set of HS_{tmp} if $\text{freq}(hs_i) < \text{minimum count}$ ($= \lceil |D| \times \lambda \rceil$), and put hs_i into the set of HS otherwise.

Step 14. Update the occurrence frequencies of all large itemsets in the sets of FI and FI_{tmp} . Put fi_j into the set of FI_{tmp} if $\text{freq}(fi_j) < \text{minimum count}$ ($= \lceil |D| \times \lambda \rceil$), and put fi_j into the set of FI otherwise.

Step 15. Update the occurrence frequencies of all small 1-itemsets in the sets of SI^1 and SI_{tmp}^1 . Put si_a into the set of SI_{tmp}^1 if $\text{freq}(si_a) \geq \text{minimum count}$ ($= \lceil |D| \times \lambda \rceil$), and put si_a into the set of SI^1 otherwise.

Step 16. Repeat Step 2 to Step 15 until the set of HS is empty ($|HS| = 0$).

4. An Illustrated Example

In this section, an example is used to illustrate the proposed algorithm step by step. Consider a database with 10 transactions (tuples) and 6 items (denoted as a to f) shown in Table 2. Each transaction can be considered a set of purchased items in a trade. The minimum support threshold is initially set at 40%, and the risky bound is set at 10%. A set of sensitive

itemsets, $HS = \{be : 6, abe : 4\}$, is considered to be hidden by the sanitization process.

Based on an Apriori-like approach [3], the large (frequent) itemsets and small 1-itemsets are mined. The results are, respectively, shown in Tables 3 and 4.

The proposed algorithm then proceeds as follows to sanitize the database for hiding all sensitive itemsets in HS.

Step 1. The transactions in D are selected with any of the sensitive itemsets in HS. In this example, the transactions 1, 3, 6, 7, 8, and 10 are selected to form the database shown in Table 5.

Step 2. The frequent itemsets in FI are processed to check whether the condition is satisfied, which is calculated as $\text{freq}(fi_j) \leq \lceil \lceil 10 \times 0.4 \rceil \times (1 + 0.1) \rceil$ ($= \text{freq}(fi_j) \leq 5$). The itemsets $\{a, ab, ae, bc, bce\}$ satisfy the condition and are kept in FI; the remaining itemsets, $\{b, c, e, ce\}$, are put into the set of FI_{tmp} .

Step 3. The infrequent 1-itemsets in SI^1 are then processed to check whether the condition is satisfied, which is calculated as $\text{freq}(si_a) \geq \lceil \lceil 10 \times 0.4 \rceil \times (1 - 0.1) \rceil$ ($= \text{freq}(si_a) \geq 3$). The itemset $\{d\}$ satisfies the condition and is kept as SI^1 ; the other itemset, $\{f\}$, is put into the set of SI_{tmp}^1 .

Step 4. The maximal count (MAX_{HS}) among the sensitive itemsets in the set of HS is then calculated. In this example, the maximal count of the sensitive itemsets $\{be\}$ and $\{abe\}$ is calculated as $MAX_{HS} = \max\{6, 4\} = 6$.

Step 5. The HFD of each transaction is calculated to evaluate the side effects of hiding failures of the processed transaction. In this example, transaction 7 is used to illustrate the following steps. According to formula (1), the HFD is calculated as $HFD^7(be) = (6 - 6 + 1)/(6 - 4 + 1) = 0.33$ and $HFD^7(abe) = (6 - 4 + 1)/(6 - 4 + 1) = 1$. The HFD of transaction 7 is calculated as $HFD^7 = 1/(0.33 + 1 + 1) = 0.43$. The other transactions are processed in the same way. The results are shown in Table 6.

The HFDs for all transactions are then normalized as shown in Table 7.

Step 6. The maximal count (MAX_{FI}) among the large itemsets in the set of FI is then calculated. In this example, the large itemsets are $\{a, ab, ae, bc, bce\}$, and the MAX_{FI} is calculated as $MAX_{FI} = \max\{5, 5, 4, 5, 5\} (=5)$.

Step 7. The MID of each transaction is calculated to evaluate the side effects of missing itemsets of the processed transaction. The frequent item $\{a\}$ in transaction 7 is used as an example to illustrate the steps. According to formula (2), the MID of the item $\{a\}$ is calculated as $MID^7(a) = (5 - 5 + 1)/(5 - 4 + 1) = 0.5$. The other frequent itemsets $ab, ae, bc,$ and bce in transaction 7 are calculated in the same way, with $MID^7(ab) = 0.5$, $MID^7(ae) = 1$, $MID^7(bc) = 0.5$, and $MID^7(bce) = 0.5$. The MID of transaction 7 is then calculated as $MID^7 = 0.5 + 0.5 + 1 + 0.5 + 0.5 (= 3)$. The other

TABLE 3: Large itemsets.

Large 1-itemset	Count	Large 2-itemset	Count	Large 3-itemset	Count
<i>a</i>	5	<i>ab</i>	5	<i>abe</i>	4
<i>b</i>	7	<i>ae</i>	4	<i>bce</i>	5
<i>c</i>	6	<i>bc</i>	5		
<i>e</i>	8	<i>be</i>	6		
		<i>ce</i>	6		

TABLE 4: Small 1-itemsets.

Small 1-itemset	Count
<i>d</i>	3
<i>f</i>	2

TABLE 5: Projected database D' .

TID	Item
T_1	<i>a, b, c, e</i>
T_3	<i>b, c, e, f</i>
T_6	<i>b, c, e</i>
T_7	<i>a, b, c, d, e</i>
T_8	<i>a, b, e</i>
T_{10}	<i>a, b, c, e</i>

TABLE 6: Hiding failure dimension for all transactions.

TID	HFD
T_1	0.43
T_3	0.75
T_6	0.75
T_7	0.43
T_8	0.43
T_{10}	0.43

TABLE 7: Normalization of HFDs for all transactions.

TID	HFD
T_1	0.57
T_3	1
T_6	1
T_7	0.57
T_8	0.57
T_{10}	0.57

TABLE 8: Missing itemset dimension for all transactions.

TID	MID
T_1	3
T_3	1
T_6	1
T_7	3
T_8	2
T_{10}	3

TABLE 9: Normalization of MIDs for all transactions.

TID	MID
T_1	1
T_3	0.33
T_6	0.33
T_7	1
T_8	0.67
T_{10}	1

transactions are processed in the same way. The results are shown in Table 8.

The MIDs for all transactions are then normalized as shown in Table 9.

Step 8. The minimal count (MIN_{SI^1}) among the small 1-itemsets in the set of SI^1 is then calculated. In this example, the small 1-itemset has only $\{d\}$, and the minimal count of the small 1-itemset is calculated as $MIN_{SI^1} = \min\{3\} = 3$.

Step 9. The AID of each transaction is calculated to evaluate the side effects of artificial itemsets of the processed transaction. Small 1-itemset $\{d\}$ in transaction 7 is used as an example to illustrate the steps. According to formula (3), the AID of the small 1-itemset $\{d\}$ is calculated as $AID^7(d) = (3 - 3 + 1)/(4 - 3) = 1$; since there is only one itemset in the set of SI^1 , no other calculations are necessary. The AID of transaction 7 is calculated as $AID^7 = 1/(1 + 1) = 0.5$. The other transactions are processed in the same way. The results are shown in Table 10.

The AIDs for all transactions are then normalized as shown in Table 11.

Step 10. The three dimensions for evaluating the selected transactions are then organized as in Table 12. The weights of hiding failures, missing itemsets, and artificial itemsets are, respectively, set to 0.5, 0.4, and 0.1. Note that these values can be defined by users to decide the importance among the dimensions. In this example, the HMAU of transaction 7 is calculated as

$$HMAU^7 = 0.5 \times 0.57 + 0.4 \times 1 + 0.1 \times 0.5 (= 0.735). \quad (16)$$

The other transactions are processed in the same way. The results are shown in the last column of Table 12.

Step 11. The selected transactions in Table 12 are then evaluated to find a transaction with the minimal HMAU value.

TABLE 10: Artificial itemset dimension for all transactions.

TID	AID
T_1	1
T_3	1
T_6	1
T_7	0.5
T_8	1
T_{10}	1

TABLE 11: Normalization of AIDs for all transactions.

TID	AID
T_1	1
T_3	1
T_6	1
T_7	0.5
T_8	1
T_{10}	1

TABLE 12: Three dimensions of each transaction in projected database.

TID	HFD	MID	AID	HMAU
T_1	0.57	1	1	0.785
T_3	1	0.33	1	0.733
T_6	1	0.33	1	0.733
T_7	0.57	1	0.5	0.735
T_8	0.57	0.67	1	0.652
T_{10}	0.57	1	1	0.785

TABLE 13: Sanitized database.

TID	Item
T_2	e
T_4	d, f
T_5	a, b, d
T_7	a, b, c, d, e
T_9	c, e
T_{10}	a, b, c, e

TABLE 14: Large itemsets of the sanitized database.

Large 1-itemset	Count	Large 2-itemset	Count
a	3	ab	3
b	3	ce	3
c	3		
e	4		

In this example, transaction 8 has the minimal value and is directly removed from Table 12.

Step 12. Transaction 8 is deleted in the dataset in this example. The minimum count is updated as $\lceil |10 - 1| \times 0.4 \rceil (= 4)$.

Step 13. The occurrence frequencies of all sensitive itemsets in the sets of HS and HS_{tmp} are, respectively, updated. Since the original database with transaction 8 consisted of the sensitive itemsets $\{be, abe\}$, which was deleted in Step 11, the counts of $\{be, abe\}$ in the set of HS are, respectively, updated as $\{be\} (= 6 - 1) (= 5)$ and $\{abe\} (= 4 - 1) (= 3)$. In this example, the set of HS_{tmp} is empty, so there is nothing to be done in this step. After the updating process, the itemset $\{abe\}$ is put into the set of HS_{tmp} since its count is below the minimum count ($3 < 4$).

Step 14. The occurrence frequencies of all large itemsets in the sets of FI and FI_{tmp} are, respectively, updated. Since the original database with transaction 8 consisted of the large itemsets $\{a, b, e, ab, ae\}$, which was deleted in Step 11, the counts of $\{a, b, e, ab, ae\}$ in the set of FI and FI_{tmp} are, respectively, updated as $\{a\} (= 5 - 1) (= 4)$, $\{b\} (= 7 - 1) (= 6)$, $\{e\} (= 8 - 1) (= 7)$, $\{ab\} (= 5 - 1) (= 4)$, and $\{ae\} (= 4 - 1) (= 3)$. After the updating process, the itemset $\{ae\}$ is put into the set of FI_{tmp} since its count is below the minimum count ($3 < 4$).

Step 15. The occurrence frequencies of all small 1-itemsets in the sets of SI^1 and SI_{tmp}^1 are, respectively, updated. Since the original database with transaction 8 did not consist of any of the small 1-itemsets in SI^1 and SI_{tmp}^1 , nothing is done in this step.

Step 16. In this example, the sensitive itemset $\{abe\}$ is already hidden, but the occurrence frequency of sensitive itemset $\{be\}$ is larger than the minimum count. Steps 2 to 15 are repeated until the set of sensitive itemsets HS is empty ($|HS| = 0$). After all Steps are processed, the sanitized database is obtained as shown in Table 13.

Comparing the original database and the sanitized one, transactions 1, 3, 6, and 8 are removed from the original database, and the minimum count is updated as 3. The updated frequent itemsets of the sanitized database are shown in Table 14.

Comparing the large itemsets in Table 3, the sensitive itemsets $\{be\}$ and $\{abe\}$ are hidden and no artificial itemset is generated. Three itemsets, $\{ae, bc, bce\}$, are, however, missing itemsets of the sanitized database. In this example, the side effects of hiding failures, missing itemsets, and artificial itemsets are 0, 3, and 0, respectively.

5. Experimental Results

Experiments are conducted to show the performance of the proposed HMAU algorithm compared to that of the aggregate algorithm [23] for hiding sensitive itemsets through transaction deletion. The experiments were coded in C++ and performed on a personal computer with an Intel Core i7-2600 processor at 3.40 GHz and 4 GB of RAM running 64-bit Microsoft Windows 7. The real database BMS-WebView-1 [34] and a synthetic database (T7I7N200D20K) [35] from IBM data generator in which T symbolizes the average length of the transactions, I symbolizes the average maximum size

TABLE 15: Details of real and synthetic databases.

Dataset	Number of transactions	Number of items	Maximum transaction size	Average transaction size
BMS-WebView-1	59,602	497	267	2.5
T7I7N200D20K	15,351	200	26	8.7

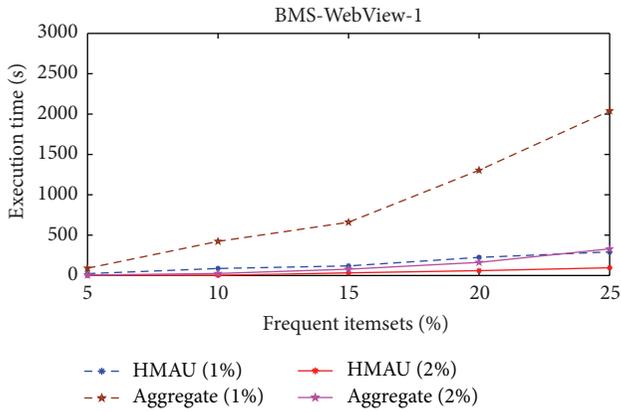


FIGURE 3: Comparison of execution time in BMS-WebView-1 database.

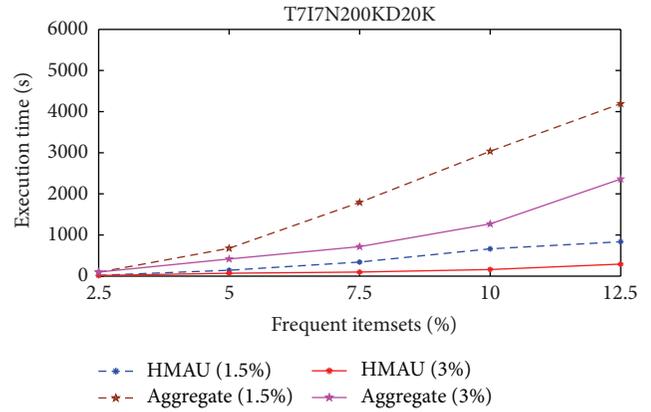


FIGURE 4: Comparison of execution time in T7I7N200D20K database.

of frequent itemsets, N symbolizes the number of differential items, and D symbolizes the size of database were used in the experiments. The details of the two databases are shown in Table 15.

For the BMS-WebView-1 database, the minimum support thresholds were, respectively, set at 1% and 2% to evaluate the performance of the proposed approach, and the percentages of sensitive itemsets were sequentially set from 5% to 25% of the number of frequent itemsets in 5% increments. In the experiments, the weights of HFD, MID, and AID in the proposed algorithm were, respectively, set at 0.5, 0.4, and 0.1.

For the T7I7N200D20K database, the minimum support thresholds were, respectively, set at 1.5% and 3%, and the percentages of sensitive itemsets were sequentially set at 2.5% to 12.5% of the number of frequent itemsets in 2.5% increments. In the experiments, the weights of HFD, MID, and AID in the proposed algorithm were, respectively, set at 0.5, 0.4, and 0.1.

5.1. Comparisons of Execution Time. Figure 3 shows the execution time of two algorithms in BMS-Web-View-1 database. Different minimum support thresholds of two algorithms are then compared in various sensitivity percentages of the frequent itemsets.

The execution time of the proposed HMAU algorithm is faster than those of the aggregate algorithm whether the minimum support threshold is set at 1% or 2%. Experiment is then conducted in T7I7N200D20K database and the results are shown in Figure 4.

From Figures 3 and 4, it is obvious to see that the proposed HMAU algorithm is faster than those of the aggregate method in two different databases.

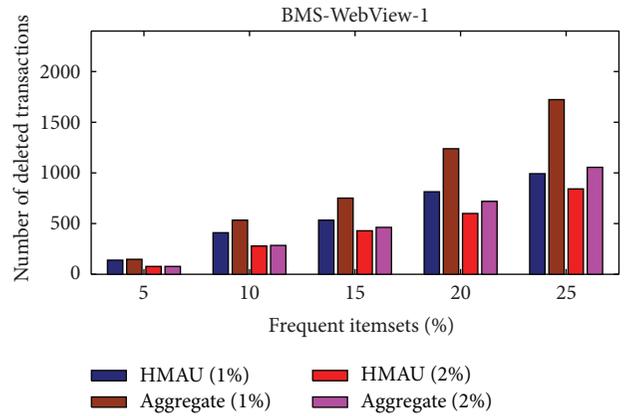


FIGURE 5: Comparison of number of deleted transactions in BMS-WebView-1 database.

5.2. Comparisons of Number of Deleted Transactions. Experiments were also conducted to evaluate the number of deleted transactions of the proposed algorithm in two different databases. For the BMS-WebView-1 database, the results are shown in Figure 5.

From Figure 5, it is obvious to see that the proposed HMAU algorithm deletes fewer transactions than the aggregate algorithm whether the minimum support threshold is set at 1% or 2% in BMS-WebView-1 database, thus achieving lower data distortion. For the T7I7N200D20K database, the results are shown in Figure 6.

From Figure 6, it is obvious to see that when the sensitive itemsets were set at 10% of the frequent itemsets with 1.5% minimum support threshold in T7I7N200D20K database, the proposed HMAU algorithm produced more transactions to be deleted for hiding sensitive itemsets. Since the proposed

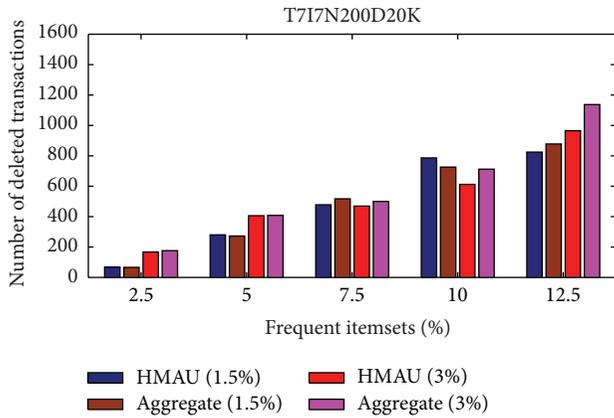


FIGURE 6: Comparison of number of deleted transactions in T7I7N200D20K database.

HMAU algorithm considers the three dimensions together, the selected transactions for deletion may consist of fewer large transactions rather than many sensitive itemsets.

5.3. *Comparisons of Side Effects.* Three side effects are then compared to show the performance of the proposed algorithm in two different databases.

The side effects of hiding failures, missing itemsets, and artificial itemsets are, respectively, symbolized as α , β , and γ . In Table 16, it can be seen that when the minimum support threshold was set at 1%, the proposed HMAU algorithm produces no side effects whereas the aggregate algorithm produces some artificial itemsets since the criteria of artificial itemsets are not considered in aggregate algorithm. Both the two algorithms produce no side effects when the minimum support threshold was set at 2%. The results to evaluate the side effects of the proposed HMAU algorithm in T7I7N200D20K database are shown in Table 17.

From Table 17, it is obvious to see that when the minimum support threshold was set at 1.5%, the proposed HMAU algorithm produces fewer artificial itemsets and missing itemsets than the aggregate algorithm for various sensitivity percentages of the frequent itemsets. The proposed HMAU algorithm produces no side effects at 3% minimum support threshold whereas the aggregate algorithm produces some artificial itemsets.

To summarize the above results for BMS-WebView-1 and T7I7N200D20K databases, the proposed HMAU algorithm outperforms the aggregate algorithm in terms of the execution time, the number of deleted transactions, and the number of side effects.

6. Conclusion and Future Works

In this paper, the HMAU algorithm is proposed for hiding sensitive itemsets in data sanitization process by reducing the side effects through transaction deletion. The formulas of three dimensions as HFD, MID, and AID are defined to

TABLE 16: Comparison of side effects in BMS-WebView-1 database.

Sensitive percentage of FIs (minimum support threshold)	HMAU			Aggregate		
	α	β	γ	α	β	γ
5% (1%)	0	0	0	0	0	0
10% (1%)	0	0	0	0	0	1
15% (1%)	0	0	0	0	0	1
20% (1%)	0	0	0	0	0	3
25% (1%)	0	0	0	0	0	2
5% (2%)	0	0	0	0	0	0
10% (2%)	0	0	0	0	0	0
15% (2%)	0	0	0	0	0	0
20% (2%)	0	0	0	0	0	0
25% (2%)	0	0	0	0	0	0

TABLE 17: Comparison of side effects in T7I7N200D20K database.

Sensitive percentage of FIs (Minimum support threshold)	HMAU			Aggregate		
	α	β	γ	α	β	γ
2.5% (1.5%)	0	0	1	0	1	2
5% (1.5%)	0	0	0	0	3	4
7.5% (1.5%)	0	0	3	0	3	7
10% (1.5%)	0	0	0	0	2	6
12.5% (1.5%)	0	0	1	0	3	6
2.5% (3%)	0	0	0	0	0	0
5% (3%)	0	0	0	0	0	2
7.5% (3%)	0	0	0	0	0	1
10% (3%)	0	0	0	0	0	1
12.5% (3%)	0	0	0	0	0	2

evaluate the correlation between the processed transactions and side effects. The weights of three evaluation dimensions of HFD, MID, and AID can be set by users' interests. In the experiments, both the real dataset and synthetic dataset are used to, respectively, evaluate the performances of the two proposed algorithms. Experimental results showed that the proposed HMAU algorithm outperforms the aggregate algorithm in terms of execution time, number of deleted transactions, and number of side effects.

In the future, the sensitive itemsets to be hidden can be extended to the sensitive association rules to be hidden. More considerations are necessary to be concerned to decrease not only the supports of sensitive itemsets but also the confidence of sensitive association rules. Other distortion approaches such as the noise addition and data modification are also the important issues to hide the sensitive information in PPDM.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the National Science Council of the Republic of China under Contract no. NSC-102-2923-E-390-001-MY3, and by the Natural Scientific Research Innovation Foundation in Harbin Institute of Technology under Grant HIT.NSRIF.2014100.

References

- [1] R. Agrawal, T. Imielinski, and A. Sawmi, "Mining association rules between sets of items in large databases," *ACM SIGMOD Record*, vol. 22, no. 2, pp. 207–216, 1993, Proceedings of the ACM SIGMOD International Conference on Management of Data.
- [2] R. Agrawal, T. Imielinski, and A. Swami, "Database mining: a performance perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 5, no. 6, pp. 914–925, 1993.
- [3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the International Conference on Very Large Data Bases*, pp. 487–499, June 1994.
- [4] T. Hong, C. Lin, and Y. Wu, "Incrementally fast updated frequent pattern trees," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2424–2435, 2008.
- [5] C. W. Lin, T. P. Hong, and W. H. Lu, "The Pre-FUFP algorithm for incremental mining," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9498–9505, 2009.
- [6] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the IEEE 11th International Conference on Data Engineering*, pp. 3–14, March 1995.
- [7] H. Cheng, X. Yan, and J. Han, "IncSpan: incremental mining of sequential patterns in large database," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 527–532, August 2004.
- [8] R. Srikant and R. Agrawal, "Mining sequential patterns: generalizations and performance improvements," in *Proceedings of the International Conference on Extending Database Technology: Advances in Database Technology*, pp. 3–17, 1996.
- [9] S. B. Kotsiantis, "Supervised machine learning: a review of classification techniques," in *Proceedings of the Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pp. 3–24, 2007.
- [10] J. R. Quinlan, *C4. 5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.
- [11] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, pp. 25–71, Springer, 2006.
- [12] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Transactions on Computers*, vol. 22, no. 11, pp. 1025–1034, 1973.
- [13] Y. Liu, W. K. Liao, and A. Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets," in *Advances in Knowledge Discovery and Data Mining*, vol. 3518 of *Lecture Notes in Computer Science*, pp. 689–695, Springer, 2005.
- [14] C. Lin, T. Hong, and W. Lu, "An effective tree structure for mining high utility itemsets," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7419–7424, 2011.
- [15] G. Lan, T. Hong, and V. S. Tseng, "Discovery of high utility itemsets from on-shelf time periods of products," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5851–5857, 2011.
- [16] C. Lin, G. Lan, and T. Hong, "An incremental mining algorithm for high utility itemsets," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7173–7180, 2012.
- [17] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 439–450, 2000.
- [18] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: a frequent-pattern tree approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53–87, 2004.
- [19] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios, "Disclosure limitation of sensitive rules," in *Proceedings of the Workshop on Knowledge and Data Engineering Exchange*, pp. 45–52, July 1999.
- [20] C. W. Lin, T. P. Hong, C. C. Chang, and S. L. Wang, "A greedy-based approach for hiding sensitive itemsets by transaction insertion," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 4, pp. 201–227, 2013.
- [21] Y. Wu, C. Chiang, and A. L. P. Chen, "Hiding sensitive association rules with limited side effects," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 29–42, 2007.
- [22] D. E. O. Leary, "Knowledge discovery as a threat to database security," in *Knowledge Discovery in Databases*, pp. 507–516, AAAI/MIT Press, 1991.
- [23] A. Amiri, "Dare to share: protecting sensitive knowledge with data sanitization," *Decision Support Systems*, vol. 43, no. 1, pp. 181–191, 2007.
- [24] S. R. M. Oliveira and O. R. Zaïane, "Protecting sensitive knowledge by data sanitization," in *Proceedings of the 3rd IEEE International Conference on Data Mining*, pp. 613–616, November 2003.
- [25] T. Hong, C. Lin, K. Yang, and S. Wang, "A lattice-based data sanitization approach," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '11)*, pp. 2325–2329, October 2011.
- [26] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, "Hiding association rules by using confidence and support," in *Proceedings of the 2001 International Workshop on Information Hiding*, pp. 369–383, 2001.
- [27] K. Duraiswamy, D. Manjula, and N. Maheswari, "Advanced approach in sensitive rule hiding," *CCSE Modern Applied Science*, vol. 3, no. 2, pp. 98–107, 2009.
- [28] A. Gkoulalas-Divanis and V. S. Verykios, "Exact knowledge hiding through database extension," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 5, pp. 699–713, 2009.
- [29] S. P. Patil and T. M. Patwar, "A novel approach for efficient mining and hiding of sensitive association rule," in *Proceedings of the 2012 Nirma University International Conference on Engineering*, pp. 1–6, 2012.
- [30] C. Wu, Y. Huang, and J. Chen, "Privacy preserving association rules by using greedy approach," in *Proceedings of the WRI World Congress on Computer Science and Information Engineering*, vol. 4, pp. 61–65, April 2009.
- [31] T. Hong, C. Lin, C. Chang, and S. Wang, "Hiding sensitive itemsets by inserting dummy transactions," in *Proceedings of the IEEE International Conference on Granular Computing (GrC '11)*, pp. 246–249, November 2011.
- [32] T. P. Hong, C. W. Lin, K. T. Yang, and S. L. Wang, "Using TF-IDF to hide sensitive itemsets," *Applied Intelligence*, vol. 38, no. 4, pp. 502–510, 2013.

- [33] B. Dai and L. Chiang, "Hiding frequent patterns in the updated database," in *Proceedings of the International Conference in Information Science and Applications (ICISA '10)*, pp. 1–8, April 2010.
- [34] Z. Zheng, R. Kohavi, and L. Mason, "Real world performance of association rule algorithms," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 401–406, August 2001.
- [35] IBM Quest Data Mining Project, "Quest synthetic data generation code," <http://www.almaden.ibm.com/cs/quest/syndata.html>.

Research Article

Separable and Error-Free Reversible Data Hiding in Encrypted Image with High Payload

Zhaoxia Yin,¹ Bin Luo,¹ and Wien Hong²

¹ Key Laboratory of Intelligent Computing & Signal Processing, Ministry of Education, Anhui University, No. 111 Jiulong Road, Hefei 230601, China

² Department of Information Management, Yu Da University, No. 168 Hsueh-fu Road, Tanwen Village, Zaoqiao Township, Miaoli County 36143, Taiwan

Correspondence should be addressed to Bin Luo; adyzx@qq.com

Received 24 February 2014; Accepted 20 March 2014; Published 6 April 2014

Academic Editors: T. Cao and F. Yu

Copyright © 2014 Zhaoxia Yin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a separable reversible data-hiding scheme in encrypted image which offers high payload and error-free data extraction. The cover image is partitioned into nonoverlapping blocks and multigranularity encryption is applied to obtain the encrypted image. The data hider preprocesses the encrypted image and randomly selects two basic pixels in each block to estimate the block smoothness and indicate peak points. Additional data are embedded into blocks in the sorted order of block smoothness by using local histogram shifting under the guidance of the peak points. At the receiver side, image decryption and data extraction are separable and can be free to choose. Compared to previous approaches, the proposed method is simpler in calculation while offering better performance: larger payload, better embedding quality, and error-free data extraction, as well as image recovery.

1. Introduction

As a kind of technique of hiding data into cover media, for example, a digital image, data hiding might often produce a distorted version of the cover, known as stego-image. Based on whether the original cover can be reconstructed or not, data-hiding techniques can be classified into two kinds: irreversible [1, 2] and reversible [3–11]. Aiming at recovering the original image with no error, reversible data hiding is generally based on two technologies: difference expansion (DE) [3, 6, 7] and histogram shifting (HS) [5, 8–10]. In general, DE-based methods provide higher payload than HS-based methods at the sacrifice of image quality. As a result, HS-based methods are more popular when the image quality is an issue. Since the maximal embedding capacity of HS-based data-hiding methods equals the number of pixels in the peak point, more peak and zero pairs are required to enhance the payload. However, for most natural images, a few nonoverlapping peak and zero pairs could be found. Therefore, embedding capacity is limited.

Moreover, most of the existing reversible data-hiding methods are only suitable for unencrypted covers. However,

in some application scenarios, content owners wish to encrypt the original images for maintaining secrecy or protecting privacy. Meanwhile, an inferior assistant or a channel administrator may desire to append some additional data within the cipher-text images without knowing the decryption key and the plaintext content. As a legal receiver, it is required that the original plaintext content can be recovered error-free after image decryption and data extraction. Reversible data hiding in encrypted image satisfies these needs [12–14].

In [13], data is appended by flipping three LSB of cipher-text image encrypted by simple exclusive-OR operation and extracted with the aid of spatial correlation in natural image. The original image can be recovered with no error when embedding payload is not too large. The performance is further improved by Hong et al. [14] using side match with a block-recovery order. In both [13, 14], the appended data can only be extracted after image decryption. In other words, a receiver having data-hiding key but no content-owner key cannot extract any information. To overcome this problem, a separable reversible data-hiding scheme [15] is

proposed, in which the original image is encrypted using symmetric key and then data can be appended using data-hiding key. With an encrypted image containing additional data, a receiver having the data-hiding key can extract the appended data exactly, while a receiver having the symmetric key can decrypt the received data to obtain an image similar to the original one. If the receiver has both keys, he can extract the additional data and recover the original image at the same time.

Method [15] is undoubtedly a great idea. However, there is a limitation in the aspect of embedding payload. Data cannot be extracted exactly and the original image cannot be recovered completely when the payload is more than 0.04 bpp (this maximal value of effective payload may fluctuate along with different cover images). In other words, error appears and error rate rises as the payload increases. Beyond that, many parameters adopted in method [15] make it a little complicated for implementation. To solve these issues, we propose an improved work in this paper. After a cover image is being encrypted with a content-owner key, additional data can be embedded into the encrypted image based on block sorting and block histogram shifting with a data-hiding key. Compared with the existing methods, the proposed scheme has the following advantages: (1) simpler calculation and higher efficiency; (2) larger payload and better embedding quality; and (3) error-free recovery with high payload.

2. Proposed Scheme

The data extraction methods used in [13, 14] require estimating block smoothness. An incorrect estimation may result in the failure of data extraction and image recovery. Although a large block size reduces the extraction error rate, it decreases the payload as well.

The proposed method embeds data by shifting pixels locally in encrypted image. The content owner partitions the cover image into nonoverlapping blocks and encrypts the cover image using multigranularity encryption: coarse-grained encryption permutes blocks in global images while fine-grained encryption permutes the pixels in each block to construct a meaningless encrypted image. Although all pixels are permuted, pixels in each block still preserve the same image histogram. Therefore, the HS method is applicable for embedding data into permuted blocks if pairs of peaks and zeros in each block are properly determined. In our approach, two basic pixels are randomly selected from the permuted block and used to indicate two peak points. Since the pixels having the same values as peaks contribute to the payload, it is likely to embed more than one bit per block to achieve high payload. More importantly, the embedded data bits can be extracted exactly. In addition, since the values of basic pixels are preserved during data embedding, they can be exploited to estimate the smoothness of blocks roughly and indicate the priority of embedding sequence.

The proposed method is described briefly as follows. In image encryption and data embedding phase, the content owner encrypts the original image I using a symmetric content-owner key κ_c to produce an encrypted image E .

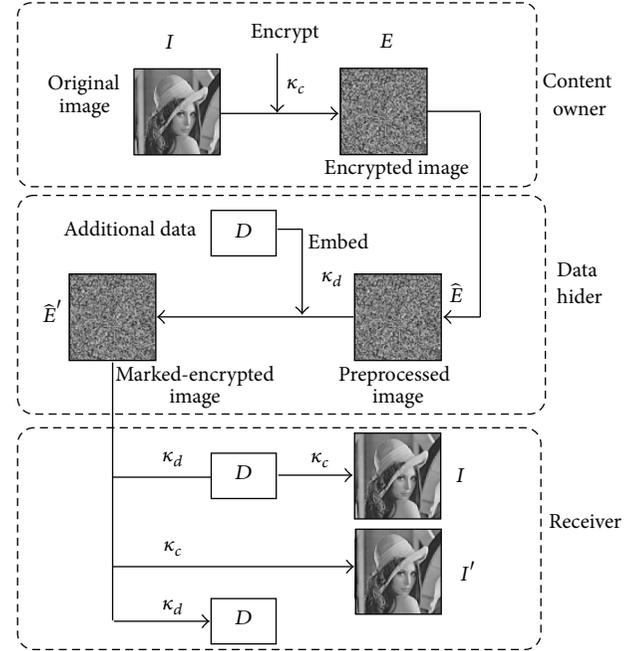


FIGURE 1: Framework of the proposed method.

Then, the data-hider processes image E to generate image \hat{E} . Additional data D is embedded into \hat{E} with data-hiding key κ_d and marked-encrypted image \hat{E}' is obtained. In data extraction and image recovery phase, there are three options for legal receivers. Image decryption and data extraction are separable and can be free to choose. The embedded data D can be easily extracted from \hat{E}' with κ_d . Since only part of pixels is modified by one grayscale unit to conceal D , direct decryption on \hat{E}' with κ_c generates a decrypted image I' , which is very similar to the original version I . If κ_c and κ_d are both adopted, the cover image I can be restored error-free and the embedded data D can be extracted accurately. The framework of the proposed method is shown in Figure 1.

The goal of the proposed method is to improve embedding payload, quality, and efficiency via simple calculation. The last and most important, keep error-free recovery as the payload increases while [13–15] cannot.

2.1. Image Encryption. Firstly, the cover image I is divided into N nonoverlapping blocks $\{B_i\}_{i=0}^{N-1}$. Each block B_i is composed of $m \times n$ pixels. Then, multigranularity encryption is adopted by using random permutation to obtain the permuted blocks $\{\hat{B}_i\}_{i=0}^{N-1}$: pixels permutation in each block and blocks permutation in the whole cover image. Thus, the encrypted image E is generated. Parameters m and n and the integer s_c adopted as the seed of random permutation compose the content-owner key κ_c .

2.2. Data Embedding. After receiving E together with block size $m \times n$, the data hider partitions E into N blocks $\{\hat{B}_i\}_{i=0}^{N-1}$. For each block \hat{B}_i , two basic pixels $\hat{b}_{i,L}$ and $\hat{b}_{i,R}$ are randomly selected and other $m \times n - 2$ pixels are denoted by $\{\hat{q}_{i,j}\}_{j=0}^{m \times n - 3}$;

that is, $\widehat{B}_i = \{\widehat{b}_{i,L}, \widehat{b}_{i,R}, \widehat{q}_{i,j}\}_{j=0}^{m \times n - 3}$. Next, to estimate the smoothness of each block, the difference $\widehat{d}_i = |\widehat{b}_{i,R} - \widehat{b}_{i,L}|$ is calculated. Blocks with smaller \widehat{d}_i are likely smoother than blocks with larger \widehat{d}_i , and it is known that smoother blocks are in favor of HS. As a result, blocks with smaller \widehat{d}_i will be chosen to have higher priority for carrying data. Let $\{\widehat{d}_{\varphi(i)}\}_{i=0}^{N-1}$ be the sorted result of $\{\widehat{d}_i\}_{i=0}^{N-1}$ after being sorted in the ascending order. The sorted sequence $\{\varphi(i)\}_{i=0}^{N-1}$ is then employed as the embedding sequence of blocks. At last, two peaks $\widehat{P}_{i,L}$ and $\widehat{P}_{i,R}$ in each block are determined as follows: $\widehat{P}_{i,L} = \min(\widehat{b}_{i,L}, \widehat{b}_{i,R})$, $\widehat{P}_{i,R} = \max(\widehat{b}_{i,L}, \widehat{b}_{i,R})$. To ensure that each block has two distinct peaks, we simply set $\widehat{P}_{i,R} = \widehat{P}_{i,L} + 1$ when $\widehat{P}_{i,L} = \widehat{P}_{i,R}$.

To avoid saturated pixels (pixels valued 0 or 255) from overflow or underflow during embedding, saturated pixels have to be preprocessed by modifying one grayscale unit and noted in a location map L . To do this, visit blocks $\{\widehat{B}_{\varphi(i)}\}_{i=0}^{N-1} = \{\{\widehat{b}_{\varphi(i),L}, \widehat{b}_{\varphi(i),R}, \widehat{q}_{\varphi(i),j}\}_{j=0}^{m \times n - 3}\}_{i=0}^{N-1}$ sequentially and append a bit "1" to L when $\widehat{q}_{\varphi(i),j} \in \{1, 254\}$. If $\widehat{q}_{\varphi(i),j} \in \{0, 255\}$, append a bit "0" to L and modify $\widehat{q}_{\varphi(i),j}$ to $\widehat{q}'_{\varphi(i),j}$ using the following equation:

$$q'_{\varphi(i),j} = \begin{cases} 254, & q_{\varphi(i),j} = 255 \\ 1, & q_{\varphi(i),j} = 0 \\ q_{\varphi(i),j}, & \text{otherwise.} \end{cases} \quad (1)$$

Let the processed block be $B'_{\varphi(i)}$. The embedding capacity of $B'_{\varphi(i)}$, denoted by C_i (bits), equals the number of nonbasic pixels valued $\widehat{P}_{\varphi(i),L}$ and $\widehat{P}_{\varphi(i),R}$. Continue the preprocessing procedures until the condition $\sum_{i=0}^{M-1} C_i \geq |L| + |D|$ is satisfied, where D is the additional data and M is the minimal number of blocks that are used for embedding L and D . We denote the preprocessed encrypted image by \widehat{E} . Once \widehat{E} is obtained, the data hider concatenates L and D to form a string of message bits S and then scans the pixels $\{\{q'_{\varphi(i),j}\}_{j=0}^{m \times n - 3}\}_{i=0}^{M-1}$ in $\{B'_{\varphi(i)}\}_{i=0}^{M-1}$ to conceal S as follows. If the scanned pixel $q'_{\varphi(i),j}$ is valued $\widehat{P}_{\varphi(i),L}$ or $\widehat{P}_{\varphi(i),R}$, a bit $s \in \{0, 1\}$ extracted from S is embedded by modifying $q'_{\varphi(i),j}$ to $q''_{\varphi(i),j}$ according to the following equation:

$$q''_{\varphi(i),j} = \begin{cases} q'_{\varphi(i),j} - s, & q'_{\varphi(i),j} = P_{\varphi(i),L} \\ q'_{\varphi(i),j} + s, & q'_{\varphi(i),j} = P_{\varphi(i),R} \end{cases} \quad (2)$$

Otherwise, pixels are either maintained or shifted by one unit using the following equation:

$$q''_{\varphi(i),j} = \begin{cases} q'_{\varphi(i),j}, & P_{\varphi(i),L} < q'_{\varphi(i),j} < P_{\varphi(i),R} \\ q'_{\varphi(i),j} - 1, & q'_{\varphi(i),j} < P_{\varphi(i),L} \\ q'_{\varphi(i),j} + 1, & q'_{\varphi(i),j} > P_{\varphi(i),R} \end{cases} \quad (3)$$

After embedding, blocks $\{B'_{\varphi(i)}\}_{i=0}^{M-1}$ are modified to $\{B''_{\varphi(i)}\}_{i=0}^{M-1}$, and the marked-encrypted image \widehat{E}' is generated.

The parameters $m, n, |L|, |S|$ and the seed S_d used to randomly select basic pixels compose the data-hiding key κ_d .

2.3. Data Extraction and Image Recovery. If the receiver has data-hiding key κ_d , the embedded additional data D can be extracted directly from the marked-encrypted image \widehat{E}' . To extract D , \widehat{E}' is firstly partitioned into blocks $\{B''_{\varphi(i)}\}_{i=0}^{N-1} = \{\{\widehat{b}_{i,L}, \widehat{b}_{i,R}, \widehat{q}''_{i,j}\}_{j=0}^{m \times n - 3}\}_{i=0}^{N-1}$ sized $m \times n$. Peaks $\widehat{P}_{\varphi(i),L}$ and $\widehat{P}_{\varphi(i),R}$ and the differences $\{\widehat{d}_i\}_{i=0}^{N-1} = \{|\widehat{b}_{i,R} - \widehat{b}_{i,L}|\}_{i=0}^{N-1}$ are then determined. Block smoothness $\{\widehat{d}_i\}_{i=0}^{N-1}$ is sorted in ascending order and the result is denoted by $\{\widehat{d}_{\varphi(i)}\}_{i=0}^{N-1}$. The sorted sequence $\{\varphi(i)\}_{i=0}^{N-1}$ is then employed as the extracting sequence of blocks. At last, the embedded data can be extracted from pixels $\{\widehat{q}''_{\varphi(i),j}\}_{j=0}^{m \times n - 3}$ in block $B''_{\varphi(i)}$ using the following equation:

$$s = \begin{cases} 0, & \widehat{q}''_{\varphi(i),j} = \widehat{P}_{\varphi(i),L} \text{ or } \widehat{q}''_{\varphi(i),j} = \widehat{P}_{\varphi(i),R} \\ 1, & \widehat{q}''_{\varphi(i),j} = \widehat{P}_{\varphi(i),L} - 1 \text{ or } \widehat{q}''_{\varphi(i),j} = \widehat{P}_{\varphi(i),R} + 1. \end{cases} \quad (4)$$

The first $|L|$ extracted bits compose the location map L , and the other $|D|$ bits compose the additional data D . If the receiver also has κ_c , original cover image I can be perfectly recovered by firstly restoring the pixels $\widehat{q}'_{\varphi(i),j}$ from $\widehat{q}''_{\varphi(i),j}$ using the following equation:

$$\widehat{q}'_{\varphi(i),j} = \begin{cases} \widehat{q}''_{\varphi(i),j}, & \widehat{P}_{\varphi(i),L} < \widehat{q}''_{\varphi(i),j} < \widehat{P}_{\varphi(i),R} \\ \widehat{q}''_{\varphi(i),j} + 1, & \widehat{q}''_{\varphi(i),j} < \widehat{P}_{\varphi(i),L} \\ \widehat{q}''_{\varphi(i),j} - 1, & \widehat{q}''_{\varphi(i),j} > \widehat{P}_{\varphi(i),R} \end{cases} \quad (5)$$

To recover $\{\widehat{q}_{\varphi(i),j}\}_{i=0}^{M-1}$ from $\{\widehat{q}'_{\varphi(i),j}\}_{i=0}^{M-1}$, if $\widehat{q}'_{\varphi(i),j} \in \{1, 254\}$, extract a bit b from L . If the extracted bit $b = 1$, set $\widehat{q}_{\varphi(i),j} = \widehat{q}'_{\varphi(i),j}$. Otherwise; that is, $b = 0$, set $\widehat{q}_{\varphi(i),j} = 0$ when $\widehat{q}'_{\varphi(i),j} = 1$; set $\widehat{q}_{\varphi(i),j} = 255$ when $\widehat{q}'_{\varphi(i),j} = 254$. Repeat until the encrypted image $E = \{\widehat{B}_{\varphi(i)}\}_{i=0}^{N-1} = \{\{\widehat{P}_{i,L}, \widehat{P}_{i,R}, \widehat{q}_{i,j}\}_{j=0}^{m \times n - 3}\}_{i=0}^{N-1}$ is reconstructed accordingly. With the content owner key κ_c , E can be exactly decrypted to the original cover image I . Note that if the receiver only has κ_c but no κ_d , an image I' that is very similar to the original one can be obtained.

3. Experimental Results

A number of gray images sized 512×512 were used as original cover images in our experiment. Figures 2(a) and 2(b) show the original image Lena and its encrypted version ($m = 4, n = 4$). After embedding 33910 bits of additional data into Figure 2(b), the stego-encrypted image was obtained, as shown in Figure 2(c) in which the embedding rate is 0.13 bpp. With the image shown as Figure 2(c), the receiver having the data-hiding key could extract the embedded data from it. The directly decrypted image only using the symmetric cryptographic key is given as Figure 2(d), and the value of PSNR between (a) and (d) is 50.51 dB. Using both the data-hiding key and cryptographic key, we successfully extracted



FIGURE 2: (a) Original Lena and error-free recovered version, (b) encrypted version, (c) stego-encrypted image containing additional data with embedding rate 0.13 bpp, and (d) directly decrypted version with PSNR 50.51 dB.

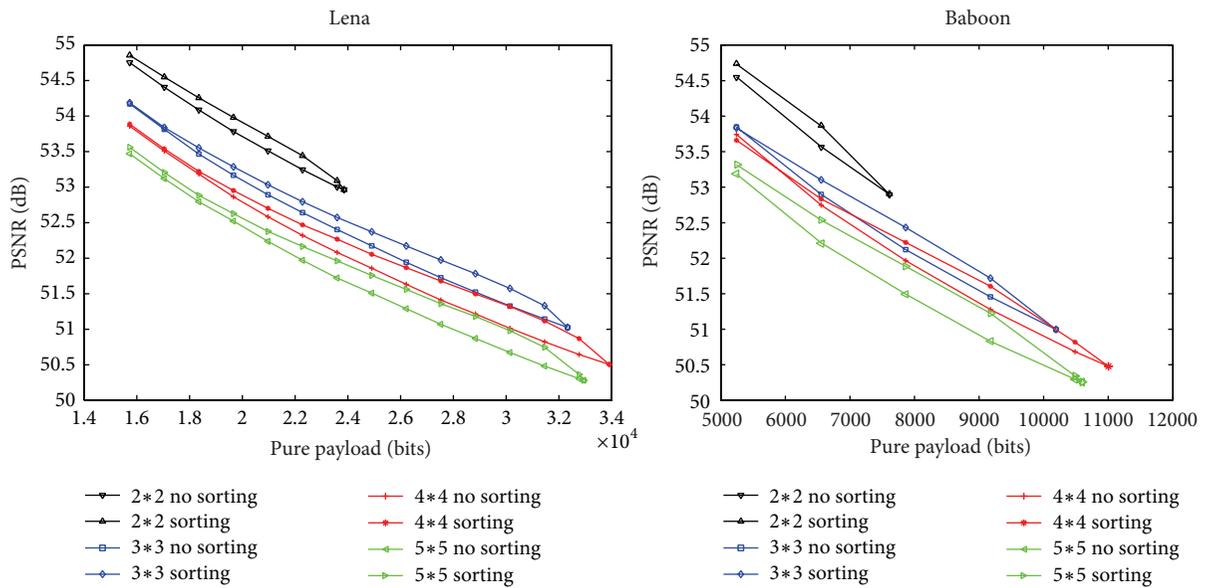


FIGURE 3: Performance of different strategies with different parameters.

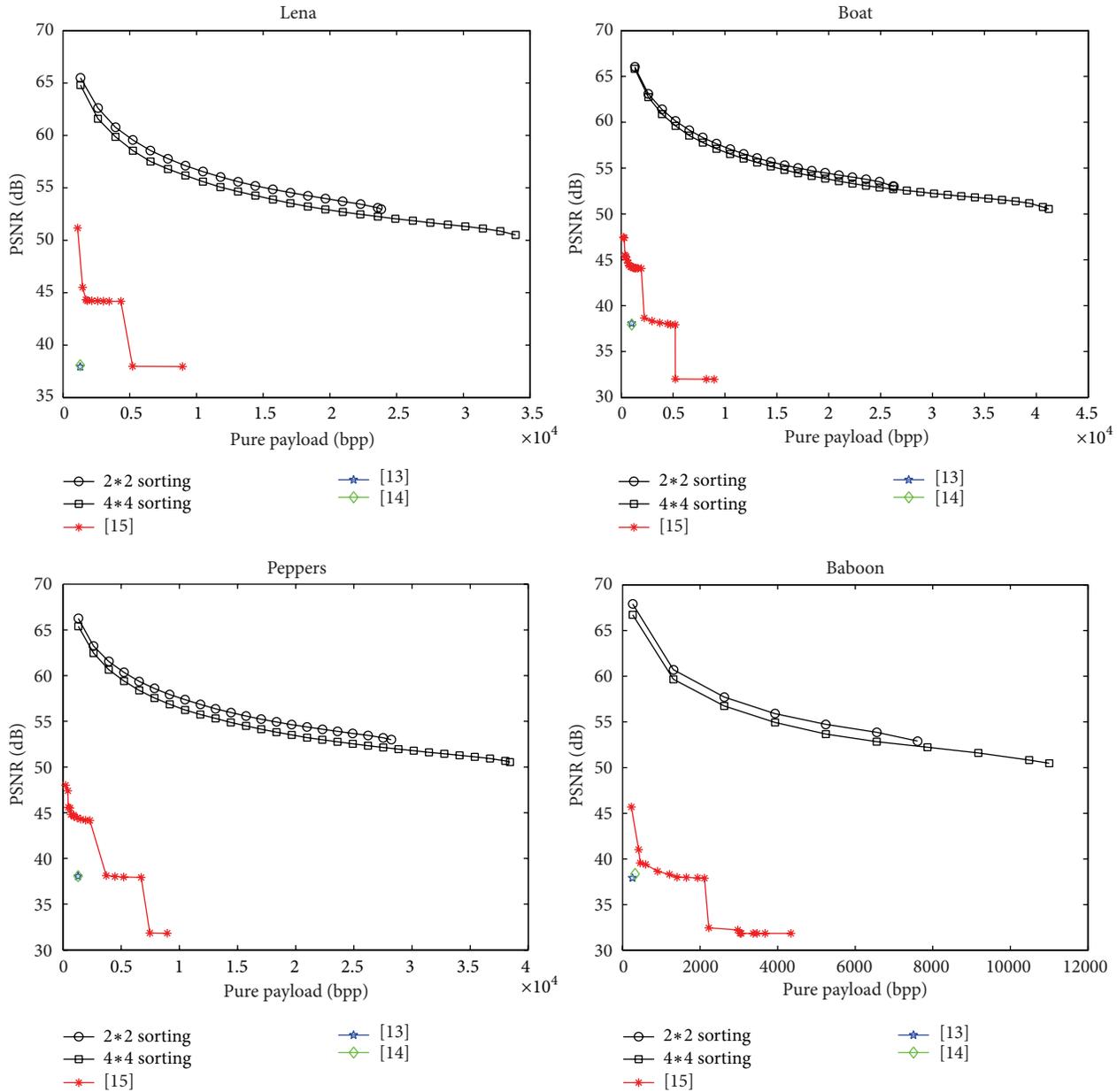


FIGURE 4: Rate-distortion performance between different approaches.

the additional data and recovered the original image error-free.

Table 1 summarizes the embedding payloads, PSNR in directly decrypted images (PSNRdec), and PSNR in recovered images (PSNRrec) when different block sizes were used for image Lena. Each “+∞” in Table 1 indicates that the mean square errors between the recovered version and the original image are 0 and the cover was reconstructed error-free.

To explore the influence of different parameters (block sizes $m \times n$) and different strategies (block sorting or no sorting) on embedding performance of proposed method, we compared different results obtained from a smooth image Lena and a complex image Baboon. In order to obtain enough experimental data and assure validity of conclusions, for each

test image, 4 block sizes ($2 \times 2, 3 \times 3, 4 \times 4, 5 \times 5$) are adopted. For each block size, 10 integers are chosen as s_c to generate different E . Then, embed 10 distinct D into each E and cross-test 100 times. After removal of the highest and lowest points, take the average to investigate PSNR-payload curves. Take Lena and Baboon as examples shown in Figure 3. The abscissa represents the pure embedding payload and the ordinate is the value of PSNR between I and I' .

From Figure 3 we can draw some conclusions. (1) The smaller the block size, the better the PSNR. If embedding quality is preferable, block size 2×2 is good choice. (2) When block size is larger than 4×4 , the performance would be worse. If large payload is desirable, either 4×4 or 3×3 could be chosen. (3) Under the same block size, the performance of

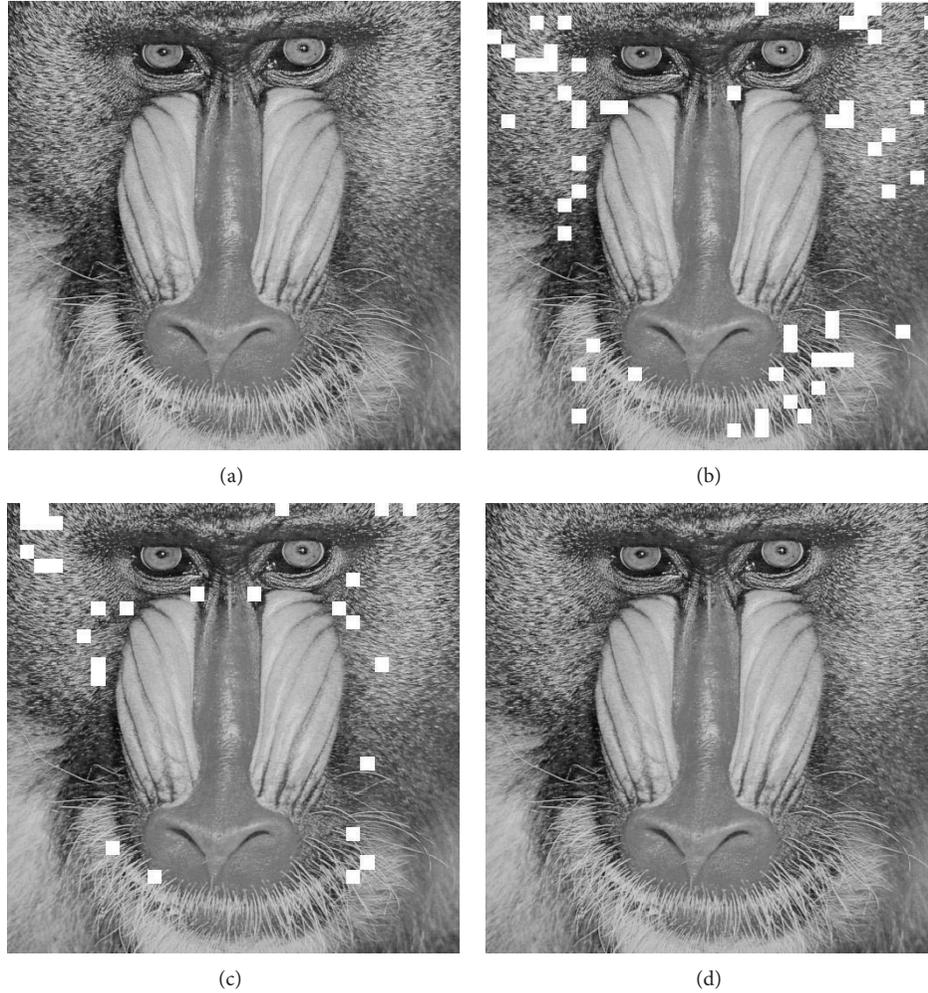


FIGURE 5: (a) Original Baboon, (b) recovered version by [13], (c) recovered version by [14], and (d) recovered version by proposed method.

TABLE 1: Maximum payload, PSNR in directly decrypted images (PSNRdec) and PSNR in recovered images (PSNRrec) when different block size was used for image Lena.

Block Size	Maximum Payload		Image Quality	
	bpp	Bits	PSNRdec	PSNRrec
2×2	0.091	23855	52.96	$+\infty$
3×3	0.123	32328	51.03	$+\infty$
4×4	0.129	33910	50.51	$+\infty$
5×5	0.126	32943	50.28	$+\infty$
8×8	0.114	29870	50.07	$+\infty$

block sorting strategy is better than that of no sorting strategy. Taking image Lena as an example, with 26214 bits of pure payload and 3×3 block size, the PSNR is 52.17 dB (sorting) and 51.94 dB (no sorting), respectively. Looking into the reason, smoother blocks are in favor of HS and have higher priority for carrying data in block sorting strategy. So, for appending the same amount of bits into the same encrypted image based on HS, fewer pixels are modified by using

block sorting strategy. That is the reason leading to higher PSNR. (4) Smooth image Lena provides better performance than complex one, Baboon. It is known that, for HS-based methods, smoother blocks often provide larger capacity than complex ones. Therefore, the full payload of Lena is larger than that of Baboon. For the same nonfull payload, fewer blocks are used in Lena and embedding distortion is smaller. So PSNR is higher.

We compared the proposed scheme with methods [13–15] in Figure 4, which indicates that the proposed scheme has the best performance. All results are derived from the best parameters under a condition that the original image can be recovered without any error.

Note that the data extraction in [13, 14] is not separable from the content decryption. However, using the proposed scheme or method [15], data extraction and image encryption are separable and can be free to choose. With the proposed scheme, since both the similarity of neighboring pixels in local level and block smoothness are fully exploited, more redundant space can be created to carry data. So the performance curve of the proposed scheme is better than those of other methods.

TABLE 2: Comparison of performance in the three aspects of Payload, PSNR and Error rate (ER) between different approaches on Lena.

	Block Size							
	4 × 4				8 × 8			
	Payload (bits)	PSNRdec (dB)	PSNRrec (dB)	Error rate (%)	Payload (bits)	PSNRdec (dB)	PSNRrec (dB)	Error rate (%)
[13]	16384	37.93	43.49	15.26	4096	37.93	54.80	1.10
[14]	16384	37.94	51.91	4.74	4096	37.93	59.02	0.42
Proposed	33910	50.51	+∞	0	29870	50.07	+∞	0

TABLE 3: Maximum payload and corresponding PSNR values.

Method	Lena		Peppers		Boat		Baboon	
	payload	PSNR	payload	PSNR	payload	PSNR	payload	PSNR
[13]	1024	37.94	1296	38.03	1024	38.06	256	37.92
[14]	1296	38.08	1296	38.05	1024	37.93	324	38.37
[15]	8956	37.96	8956	31.82	8956	31.96	4340	31.86
Proposed	33910	50.51	38420	50.54	41194	50.54	11004	50.48

We also compared the proposed scheme under the same block size with the nonseparable method in [13, 14]. The results are shown in Table 2, from which we see that the proposed scheme has 2 times gain of payload under the block size 4×4 and 7 times gain under block size 8×8 together with improvement of PSNR value in directly decrypted image when meeting the perfect recovery condition.

Furthermore, we take Baboon as an example to verify details. Under the same block size 8×8 , we compare recovered images generated by different methods in Figure 5, where the incorrect recovered blocks are marked by white.

Comparing Figures 5(b), 5(c), and 5(d), we see that the proposed method recovers the image blocks error-free and more accurate than that of [13, 14]. Although the experiments were based on Baboon, experiments on other test images also showed the similar result, which indicates that the proposed method offers a better performance for data extraction and image recovery.

Finally, we summarize maximum payload and corresponding PSNR of Lena, Peppers, Boat, and Baboon in Table 3. For the same cover image, the maximum payload of the proposed scheme is much more than that of [13–15] and the embedding quality is the best.

4. Conclusion

This paper proposed a separable and error-free reversible data-hiding scheme in encrypted image, which significantly outperforms the previous methods in the three aspects of payload, PSNR, and error rate. Compared with [13, 14], not only can cover images be reconstructed with no error, but also image decryption and data extraction are separable. Compared with [15], the proposed method improves both PSNR and the effective payload via simpler calculation using few parameters and achieves higher efficiency. The last and most important advantage of our method is that it can keep error-free recovery as the payload increases while the others cannot.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper. The authors of the paper do not have a direct financial relation that might lead to a conflict of interests for each other.

Acknowledgments

This research work is supported by the National Natural Science Foundation of China under Grant no. 61073116, Excellent Young Talents Fund Program of Higher Education Institutions of Anhui Province under Grant no. 2012SQRL014, and Youth Research Foundation of Anhui University under Grant no. 02303305. The paper has not been previously published, is not currently submitted for review to any other journal, and will not be submitted elsewhere before a decision is made by this journal.

References

- [1] J. M. Bahi, J. F. Couchot, and C. Guyeux, "Steganography: a class of secure and robust algorithms," *The Computer Journal*, vol. 55, pp. 653–666, 2012.
- [2] W. Hong and T. Chen, "A novel data embedding method using adaptive pixel pair matching," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 176–184, 2012.
- [3] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.
- [4] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253–266, 2005.
- [5] Z. Ni, Y. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.
- [6] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. Choo, "A novel difference expansion transform for reversible data embedding,"

- IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 456–465, 2008.
- [7] Y. Hu, H. Lee, and J. Li, “DE-based reversible data hiding with improved overflow location map,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 250–260, 2009.
- [8] W. Tai, C. Yeh, and C. Chang, “Reversible data hiding based on histogram modification of pixel differences,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906–910, 2009.
- [9] P. Tsai, Y. Hu, and H. Yeh, “Reversible image hiding scheme using predictive coding and histogram shifting,” *Signal Processing*, vol. 89, no. 6, pp. 1129–1143, 2009.
- [10] X. Gao, L. An, Y. Yuan, D. Tao, and X. Li, “Lossless data embedding using generalized statistical quantity histogram,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 8, pp. 1061–1070, 2011.
- [11] W. Hong, “Adaptive reversible data hiding method based on error energy control and histogram shifting,” *Optics Communications*, vol. 285, no. 2, pp. 101–108, 2012.
- [12] W. Puech, M. Chaumont, and O. Strauss, “A reversible data hiding method for encrypted images,” in *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, vol. 6819 of *Proceedings of SPIE*, p. 9, 2008.
- [13] X. Zhang, “Reversible data hiding in encrypted image,” *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.
- [14] W. Hong, T. S. Chen, and H. Y. Wu, “An improved reversible data hiding in encrypted images using side match,” *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199–202, 2012.
- [15] X. Zhang, “Separable reversible data hiding in encrypted image,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, 2012.

Research Article

A Coverage and Slicing Dependencies Analysis for Seeking Software Security Defects

Hui He,¹ Dongyan Zhang,² Min Liu,¹ Weizhe Zhang,¹ and Dongmin Gao¹

¹ School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

² Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing, China

Correspondence should be addressed to Weizhe Zhang; wzzhang@hit.edu.cn

Received 20 January 2014; Accepted 26 February 2014; Published 2 April 2014

Academic Editors: T. Cao and F. Yu

Copyright © 2014 Hui He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Software security defects have a serious impact on the software quality and reliability. It is a major hidden danger for the operation of a system that a software system has some security flaws. When the scale of the software increases, its vulnerability has becoming much more difficult to find out. Once these vulnerabilities are exploited, it may lead to great loss. In this situation, the concept of Software Assurance is carried out by some experts. And the automated fault localization technique is a part of the research of Software Assurance. Currently, automated fault localization method includes coverage based fault localization (CBFL) and program slicing. Both of the methods have their own location advantages and defects. In this paper, we have put forward a new method, named Reverse Data Dependence Analysis Model, which integrates the two methods by analyzing the program structure. On this basis, we finally proposed a new automated fault localization method. This method not only is automation lossless but also changes the basic location unit into single sentence, which makes the location effect more accurate. Through several experiments, we proved that our method is more effective. Furthermore, we analyzed the effectiveness among these existing methods and different faults.

1. Introduction

Software is the soul of the information systems [1] and plays a crucial role in the information society. High-tech product is often a software-intensive system, from cell phone daily used to Internet applications connected the global. Software has become a part of people's lives. However, software defects seriously affect software quality and software reliability, and it is the major hidden danger of information systems operating stably. Constant software failure and accident make people have to pay much attention to software quality and reliability. So on the one hand, the existence of software defects has brought a great challenge to the safe and reliable operation of information systems. On the other hand, to a greater extent it has affected national security, social stability, and economic development [2].

Software security testing is an extremely challenging task. Statistics show that, in a typical software development project, software testing often accounts for about 50% of the total workload. For some high security and reliability software, testing time even accounts for 60% of the development cycle

[3, 4]. As an important part of the automation in the software development process, automated software testing has caused widespread concern. On the other hand, with the continuous development of software technology, new software error or defect types continue to be found. It brings new problems and challenges to software testing technology. Therefore, the research of accurate, high degree of automation software debugging method, on the one hand, can help correctly understand the trigger conditions of the defect and repair defect reasonably. On the other hand, you can greatly reduce the scope of the code to be checking the number of program execution, workload, and cost. And software fault diagnosis is a critical step in the software security testing.

2. Software Fault Localization Methods

Software fault localization and diagnosis technology is an important technology in the software defect detection methods. Although the study contains multiple ways, the software only during the test and run-time environment integrated consideration, so software testing in defect detection cannot

be replaced. At the same time, the growth of software size and software structure and morphological changes make fault localization and diagnosis in the software testing more difficult. It directly indicates the need for software fault localization and diagnostic techniques.

In the existing research, typical fault localization methods include coverage fault localization and the incremental software debugging (Delta Debugging), based on program slicing fault localization. According to different starting points and analysis of objects, the methods above can be divided into two categories, defined as single execution record fault localization and based on the statistical fault localization [5]. Single execution record fault localization method is execution results oriented. It means that, in the fault localization process, not only do we need to know if the execution result is correct, but also, more importantly, we need to know clearly the error result in the execution results corresponds to which part of the error variables. However, existing fault localization methods based on statistics typically only need to know whether the software execution process is successful or not. Then collect two or more recordings during different execution processes, and calculate the fault possibility of code segment by statistical methods.

Fault localization based on program slicing is a representative of single execution record methods. Incremental debugging method based on coverage can fall into the fault localization method based on statistics.

CBFL (coverage based fault localization) method first inserts additional code, characterizing the behavior of software into the source code, and the program is called Cartridge. Then, execute the test cases and collect execution record. Finally, calculate the value of the defect tendency by counting the successful and failure rate in the process of a number of executions (commonly known as code coverage). Then isolate and sort code by the value based on debugging tendency, which narrows the scope of the code to be checked. Representative methods of CBFL are Tarantula [6], SBI [6], JACCARD [7], and OCHIAI [8].

Incremental fault localization method is called Delta Debugging; first by simplifying and separating the input method get a practicable test case, which has the closest running track with a failed test case. Then locate the fault in the program by comparing the two test cases' states at some point.

Program slicing based fault localization method, first proposed by Mark Weise, introduced the concept of the program slicing [9, 10] and the first static slicing algorithm and applied to the auxiliary debugger. A static slice represents a statement collection possibly affecting the output in the program [11]. Because debugging often needs to analyze the statements actually run in the program, dynamic slice is defined as a subset of statements set run in the program. It refers to the statements collection really impacting the output during the run. For dynamic slicing, the effectiveness of fault localization is judged by hit rate and searching scope.

In conclusion, the root of CBFL methods' defects is to ignore control dependencies and data dependencies between different statements in the program. So it will be a good choice to process the code by dependencies between programs and

then locate faults by CBFL. While the root of program slicing fault localization methods' defects is a too large amount of defect codes, it is necessary to reduce the size of sliced codes appropriately. Research shows that, in sliced codes, some statements are not necessary. That is to say, there is no dependency between them and the expected output. Then we can consider reducing the sliced code amount by finding out these independent codes. The improvements of these two methods are required for analyzing the structure of the program and clearing the control dependencies and data dependencies between programs. So it is necessary to introduce the contents of analyzing the program structure in the following subsections.

3. Data Dependencies Analysis

3.1. Definition of Data Dependencies. For different types of code statements, we set different data dependencies.

From Figure 1, we know that the left value and the right value are defined as two linked lists. For different types of statements, we define different data dependencies. For different code statements, we define different data dependencies, too. Specific code definitions are shown in Table 1.

3.2. Data-Dependent Extraction. We focused on reverse data dependence analysis on the same program execution paths and extract the data dependencies on this path. Based on a particular variable, we traversed the stored data dependence reversed.

3.2.1. Data-Dependent Extraction Algorithm. Before reversing data dependencies analysis, it is essential to get different execution paths of the program based on analysis of control flow and then merge them.

Objects of reverse data dependencies extraction are execution paths extracted. Extraction process is along execution paths from back to front. Store the data dependencies before extracting reversely. The process of data dependency analysis and storage is actually doing text analysis of the program. Get types of code statements by lexical analysis and syntax analysis. Then gather left and right values of code segments to form linked lists, depending on the data dependencies on different code statements, in accordance with different ideas of extraction. At last, store the linked lists into data dependencies table. Specific extraction algorithm is as Algorithm 1.

The algorithm describes the execution algorithm of data dependencies extraction. The concrete implementation of the algorithm is given in the form of pseudocode. Different code types have different ideas of extraction and should be classified in detail.

3.2.2. Data-Dependent Storage and Traversal Algorithms. In view of the definition of a data dependency and for subsequent traversal looking for convenience, we designed a data-dependent storage structure as Figure 2.

We define the basic node as a structure array, which is used to store each code line. This structure contains two fields,

Source program	Dependencies	Left linked list	Right linked list
(1) int a, b, c, x, y ;	①	$a \rightarrow b \rightarrow c \rightarrow x \rightarrow y$	null
(2) $a = 10$;	②	a	null
(3) $b = a + c$;	③	b	$a \rightarrow c$
(4) if ($a > b$) {	④	$a \rightarrow b$	$a \rightarrow b$
(5) $x = a * b$;	⑤	$a \rightarrow b \rightarrow x$	$a \rightarrow b$
(6) While ($t! = 0$) {	⑥	$a \rightarrow b \rightarrow t$	$a \rightarrow b \rightarrow t$
(7) $y = 3$;	⑦	$a \rightarrow b \rightarrow t \rightarrow y$	$a \rightarrow b \rightarrow t$
(8) $x = y \% 3$;	⑧	$a \rightarrow b \rightarrow t \rightarrow x$	$a \rightarrow b \rightarrow t \rightarrow y$
(9) }	⑨	null	null
(10) }	⑩	null	null
(11) return x ;	⑪	null	null

FIGURE 1: An example of data dependencies definition.

TABLE 1: Definitions of different statements data dependencies.

Statement type	Definition of left and right values	Examples of statements	Left and right linked lists	
Assignment statement				
Definition declaration	Left value is the linked list defining variables, and right value is null.	Int a, b ;	$a \rightarrow b$	$a \rightarrow b$
Direct assignment	Left value is the linked list defining variables, and right value is null.	$a = 10$;	a	a
Indirect assignment	Left value is the assigned variable, and right value is the assigned value.	$a = b + c$;	a	$b \rightarrow c$
Control statement				
While	Left value and right value are the same in these statements, which are variables involved in(). Left and right values of these statements are included in the left and right linked lists of all statements in the control domains.	While ($a > b$)	$a \rightarrow b$	$a \rightarrow b$
If else		If ($a > 10$)	a	a
For		For ($i = 1; i < b; i++$)	$i \rightarrow b$	$i \rightarrow b$
Console statement	Left value and right value are both null.	return x ;	Null	Null
Function call				
User-defined	Related to parameter types.	Abs($*a, b$)	a	Abs()
Library function	Left and right values are both null.	Print f()	Null	Null

which are the left and right pointers, and link to each of the lines of code around value chains. It also contains a code line number, in order to search in the subsequent traversal store table easily. Each node consists of three fields, including the variable name of the node name, node type, and flag (value of 1 indicates a function and a value of 0 indicates normal variable), as well as a link to the next pointer field of the node next.

In the following section we will show how to traverse to find the specified variable. We use Figure 3 to illustrate basic search process.

As indicated in the figure, we specify the variables x ; beginning to find from the last row of the stores table, find the row to x as an l -value (8 lines); line 8 is added into S collection, and collection S started with empty value. Then start to search from each variable of line 8 right chain in turn. Take the right chain of a variable a as an example, we specify a as a new variable, then we search the line ahead. The search process is exactly as x . We put the searched line number into S collection until all line numbers are found for the specified variable. Then, we find b to specify the line number variables. Until the reverse traversal to the row of right meets *null*

```

Input: Source file source.c
Output: Stored linked lists
Symbolic representations:
Remove (): function of removing comments and blank lines
Source1.c: file after intermediate treatment (got after executing 1-5)
Buffer: global string variable
Get: gather variables, form linked lists
Left: left linked list
Right: right linked list
Store: store linked lists into storage table
Get_control_end: find the end of control structure

(1)  {Repeat buffer = getline (source.c)  //remove blanks and comments
(2)    If existing note or blank line
(3)    Remove (note or blank line)
(4)    Else go to 1
(5)  Until EOF}
(6)  {Repeat: buffer = getline (source1.c)
(7)    If buffer is Assignment{//process assignment statement
(8)      Flag = 0
(9)      Get (left and right)//extract left and right
(10)     Store (left and right)// store left and right
(11)   If buffer is function{//process function call statements
(12)     Flag = 1
(13)     Do nothing}
(14)   If buffer is control_statement{//process control statements
(15)     Flag = 2
(16)     Get_control_end//find out the scope of the control domain
(17)     Get (left and right)
(18)     Store (left and right)
(19)     Recursive{//recursive processing control statements
(20)   Until EOF}
    
```

ALGORITHM 1: Data_dependent_extraction algorithm (source.c).

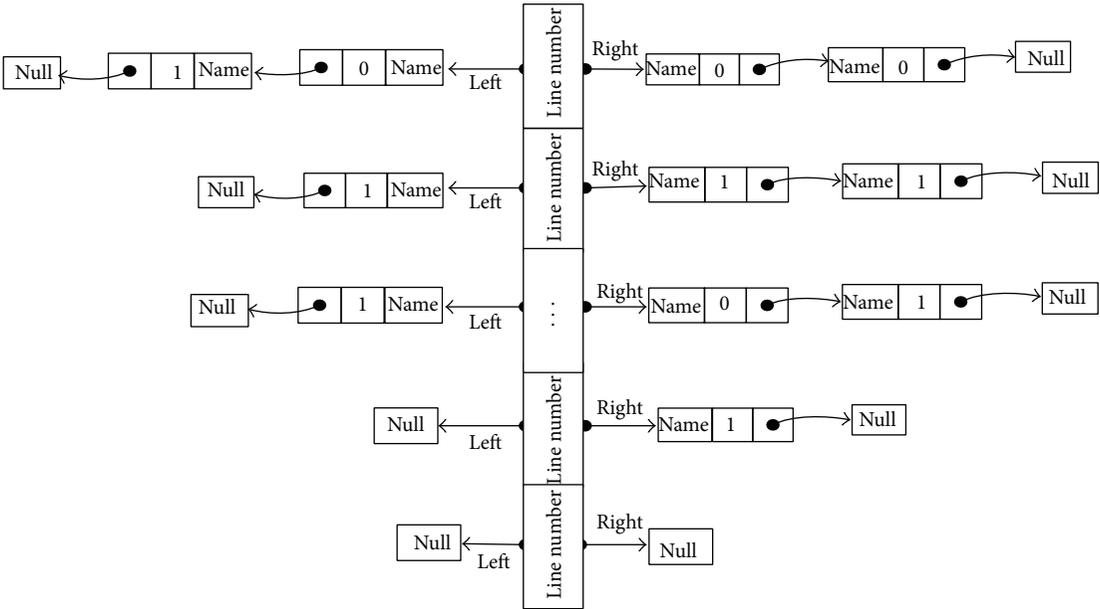


FIGURE 2: Data-dependent storage diagram.

Source code	Left chain	Right chain
(1) int a, b, c, x, y;	$a \rightarrow b \rightarrow c \rightarrow x \rightarrow y$	null
(2) a = 10;	a	null
(3) b = a + c;	b	$a \rightarrow c$
(4) if (a > b){	$a \rightarrow b$	$a \rightarrow b$
(5) x = a * b;	$a \rightarrow b \rightarrow x$	$a \rightarrow b$
(6) While (t != 0){	$a \rightarrow b \rightarrow t$	$a \rightarrow b \rightarrow t$
(7) y = 3;	$a \rightarrow b \rightarrow t \rightarrow y$	$a \rightarrow b \rightarrow t$
(8) x = y % 3;	$a \rightarrow b \rightarrow t \rightarrow x$	$a \rightarrow b \rightarrow t \rightarrow y$
(9) }	null	null
(10) }	null	null
(11) return x;	null	null

FIGURE 3: Data-dependent traverse diagram.

values, it marks traverse to the end on the specified variable. This is a recursive lookup process. The figure only shows the first step to find variable *a*, without recursive lookup process. At the end of the first step of variable *a*, set $S = \{8, 7, 6, 5, 4, 2\}$.

Until we find all variables in the right chain of line *x*, we traverse the collection *s* and eliminate duplicate rows to ensure uniqueness of the collection elements. The traversal algorithm is shown in Algorithm 2.

4. Coverage and Program Structure Slicing Fault Localization

4.1. Basic Idea Description. Considering the software security flaws and advantages of both CBFL and program date slicing fault localization and combining with the theoretical basis of the analysis of program structure, we propose a fault localization solution CPSS (coverage and program structure slicing) based on program execution path structure data dependency analysis.

The basic idea description is as shown in Figure 4.

Extraction execution paths: it is the major work of the control flow analysis, according to the execution of test cases, combining with the relevant command GCC, extract program execution path.

Execution path merge: every execution integration path is streamlined into a complete program dependence graph, for convenient follow-up coverage of statistical analysis.

Reverse data dependence analysis: through the application of text analysis, it is an important task of the reverse data

dependence model and extracts the data on the execution path dependency.

Calculating defect rate: combined with the statistical error correction of quantitative calculation model, programming calculation of different defects tends to code statements.

Sorting code and isolation: the high and low order is from the high to low value according to the defects.

In order to more clearly show our error locating methods, this section gives a method of our overall execution system diagram as Figure 5.

Our research work mainly divides into five parts.

(1) *Constructing Experimental Environment Module.* Experiments involve SIR itemsets using the GCC command and the use of a shell script, so setup of appropriate experimental environment is the assurance that follow-up experiments can go smoothly. Set up the experimental environment; the main work is to set the environment variable to write a shell script, marking procedure, and hand all false and hand all errors and a series of work.

(2) *Extracting Program Control Flow and Extracting the Execution Path Module.* Based on the analysis of the control flow, we use GCC command to extract the program control flow diagram, based on the extraction of control flow file execution path of extraction by programming.

(3) *Extracting the Data Flow Module.* Through reverse data dependence model to extract the execution path of data dependency. We reduced the amount of code execution path by program slicing.

```

Input: the execution path (given in the form of program code fragment)
Output: program slicing (the set  $s$ ,  $k$  number of element)
Symbol indicates:
Left: Each code line of the left chain
Right: Each code line of the right chain
Store[ $n$ ]: Storage table
Var: Specified variable
L_pointer: Left temporary pointer variables
R_pointer: Right temporary pointer variables
Initialization:  $S \leftarrow \Phi$ 

(1) {Repeat  $n = n - 1$ 
(2)  Left = store[ $n$ ] -> left
(3)  Right = store[ $n$ ] -> right
(4)  line_num =  $n$ 
(5)  L_pointer = left
(6)  {Repeat: L_pointer = L_pointer -> next
(7)    If (L_pointer -> name) == var
(8)      go to 13
(9)  Until L_pointer == null}
(10) go to 1
(11) Until (L_pointer -> name) == var}
(12) Add line_num to  $S$  //code line numbers added to  $S$  set
(13) R_pointer = right
(14) {Repeat R_pointer = R_pointer -> next
(15)  var = R_pointer -> name
(16)  traverse (var) // Recursive traversal algorithm
(17) Until R_pointer == null} // Find next to the variable field is null
(18) {Repeat  $t_i \in S$  //Simplified repeat statements in collection
(19)   $j = i + 1$ 
(20)  {Repeat  $t_j \in S$ 
(21)    if  $t_j = t_i$ 
(22)      delete  $t_j$ 
(23)  Until  $j = k$ }
(24) Until  $i = k$ }

```

ALGORITHM 2: Traversal algorithm: traverse (var).

(4) *Quantitative Calculation Module*. Reference to quantitative calculation method, combining with the former several modules are program slicing, the locating method based on coverage, through the calculation of programming implementation code defects.

(5) *Ordering Module*. According to quantitative calculation module, result of a code defect level inclination value is as a basis for ordering code statement. Isolation of code statements tended to have a high priority in debugging.

4.2. *Control Flow-Oriented Execution Paths Extraction and Mergence*. It is vital to extract different execution paths of the program in reverse data dependencies process, and among them, extracting control flow, execution paths, and paths mergence are utmost basis of execution paths tracing.

Through comparing experiments on the program control flow graph above the execution results of the related test cases, we found that each test execution case corresponds to an execution path and failure test cases correspond to failure

execution paths. And we also found out that the distribution of the failure of implementation errors in different control flow path is important for the improvement of positioning accuracy.

For example, suppose there are two basic block execution paths e_1 and e_2 covering $b, b_1 \rightarrow b_2 \rightarrow b_4$ and $b_1 \rightarrow b_3 \rightarrow b_4$, respectively. Set override basic block b process in the implementation of the following two conditions.

The first case, all cover $b_1 \rightarrow b_3 \rightarrow b_4$, is successful implementation, all cover $b_1 \rightarrow b_2 \rightarrow b_4$ is failed implementation, and the successful and failure implementation perform the same times as follows:

$$\begin{aligned}
 &\text{failed } (b_1 \rightarrow b_2 \rightarrow b_4) = 2 \\
 &\text{Passed } (b_1 \rightarrow b_2 \rightarrow b_4) = 0 \\
 &\text{failed } (b_1 \rightarrow b_3 \rightarrow b_4) = 0 \\
 &\text{passed } (b_1 \rightarrow b_3 \rightarrow b_4) = 2.
 \end{aligned} \tag{1}$$

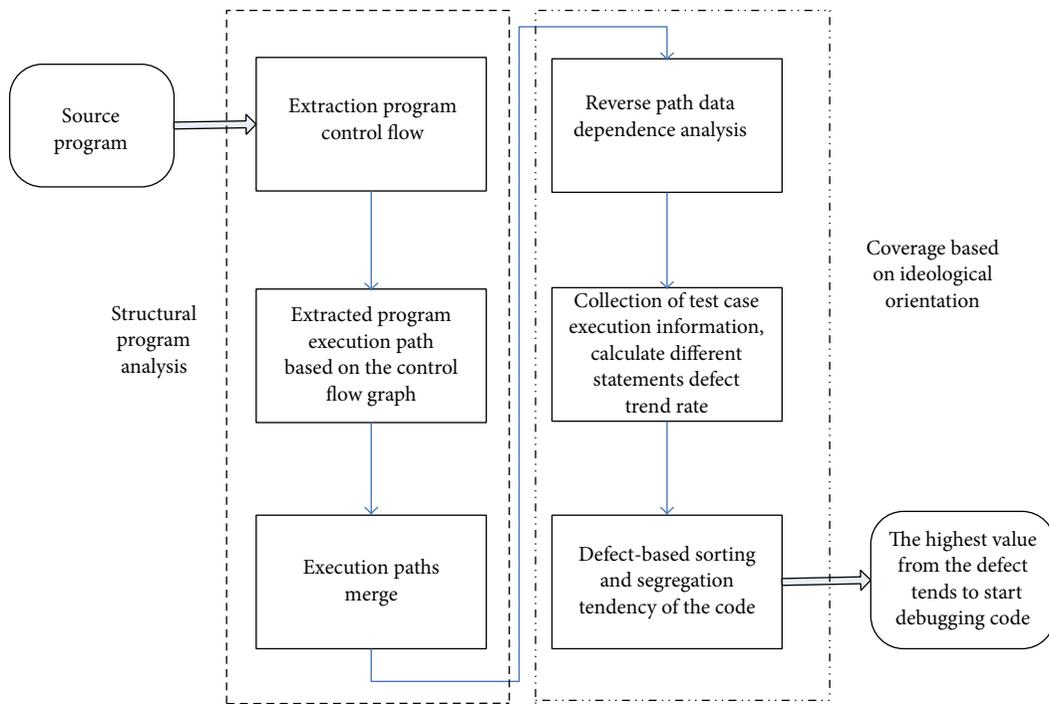


FIGURE 4: Basic idea description block diagram.

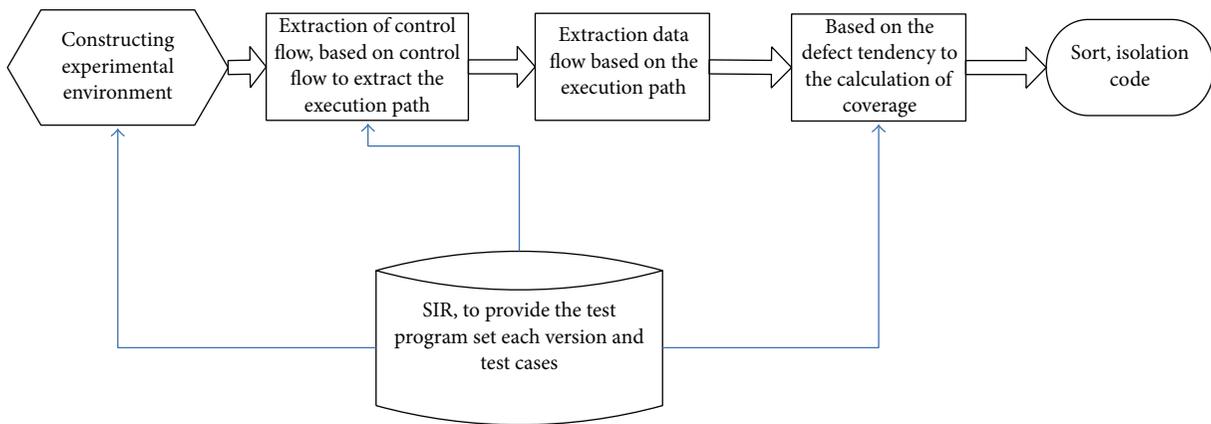


FIGURE 5: System structure diagram.

For the second case, cover $b_1 \rightarrow b_3 \rightarrow b_4$ path of execution, implementation times of successes and failures are the same as follows:

$$\begin{aligned}
 &\text{failed } (b_1 \rightarrow b_2 \rightarrow b_4) = 1 \\
 &\text{passed } (b_1 \rightarrow b_2 \rightarrow b_4) = 1 \\
 &\text{failed } (b_1 \rightarrow b_3 \rightarrow b_4) = 1 \\
 &\text{passed } (b_1 \rightarrow b_3 \rightarrow b_4) = 1.
 \end{aligned}
 \tag{2}$$

The basic code and the successful execution of the block b have the number of the same failures. For the first case, since

$b_1 \rightarrow b_3 \rightarrow b_4$ execution failure does not exist, the error status of basic block b does not pass the $b_1 \rightarrow b_3 \rightarrow b_4$ delivery. If basic block b is flawed, $b_1 \rightarrow b_3 \rightarrow b_4$ execution failure should exist. The same reasoning applies to the first case, the basic block b flawed great possibility.

The same control flow for different execution statistics is just double-counted, but different control flow paths of execution paths will fail to solve the program, which has a positive effect.

Therefore, By considering multiple execution paths in the actual testing process result, we merge different paths according to their corresponding control flow after all execution paths data are preprocessed.

5. Experimental Results and Analysis

5.1. Comparative Analysis of the Fault Localization Effect. In this section, we will take advantage of method CPSS proposed in our paper to locate the error in project flex v1 [12] to verify the effectiveness of the proposed method. We will compare the results obtained by using Tarantula, CT, and SBI.

To judge the CPSS's seeking effect, we need to contrast the result with the existing location methods. We chose three location methods, Tarantula, CT, and SBI. Specific experimental data are shown in Table 2.

Table 2 shows the respective proportional band of the detected code. Four methods are able to detect the percentage of faults. The abscissa is the ratio of code required to be detected to the total amount of codes. Ten percentage points represent that how much code is detected of each method. The vertical axis is the proportion of detected fault, which indicates what percentage of faults can be located in the abscissa specified levels. Under 20% of the code detection situation, CPSS can detect 85.71% of the faults, and other methods should be less than this figure. And only detecting 40% of the code CPSS can locate all faults; other methods cannot make it clearly. In the comparison of the remaining three methods, obviously, the SBI method is better than the other two methods.

Next, we will compare CPSS with SBI, the best method in the remaining, in different fault types, and analyze the pros and cons of these two methods in the location effects.

5.2. Comparative Analysis of SBI and CPSS Location Effect. According to the classification in Section 5.1, condition judgment fault, assignment fault, and function fault, this paper tries to compare CPSS with SBI and reveal the location effect of CPSS for further different fault styles.

5.2.1. Comparison of the Condition Judgment Fault. Condition judgment fault refers to faults from if condition and condition judgment statement of while loop statement, and condition judgment is stricter or looser to make results not matching the expectation. Project flex v1 has 4 condition faults.

As can be seen from Figure 6, under the detection rate of 1% which can best embody the location effect, CPSS may locate 25% of condition faults; that is, the CPSS can accurately locate 25% of the condition faults, while SBI method cannot directly and accurately locate any condition fault. If the detection rate is 20%, CPSS may locate 75% of faults, but SBI method may locate only 50%. Within the range of 1%–30% of code detection, CPSS curve has been higher than the SBI curve, which means at this stage the effect of CPSS is better than SBI. Until the detection rate is greater than 30%, do they overlap with each other? In both after 40% of the code is detected, the detection rate reaches 100%; that is, detecting 40% of the code can fully locate all the condition faults, and the result can be accepted. Taken together, CPSS method is better than the SBI method when locating condition faults.

In many cases, the formation of condition fails because the values do not meet desired objectives. But CPSS method

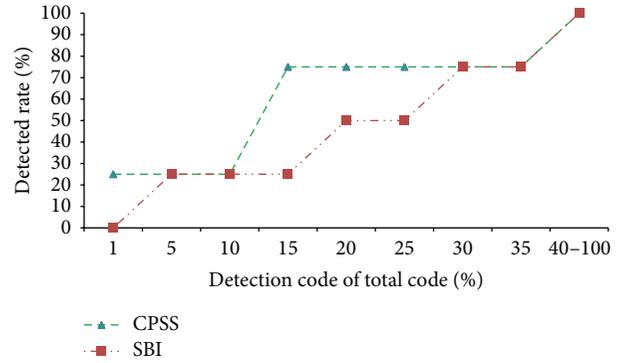


FIGURE 6: Comparison of location effect of condition judgment fault.

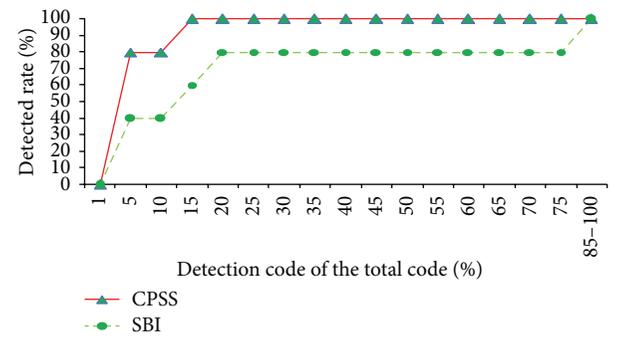


FIGURE 7: Location effect of the assignment fault.

combines reverse data dependency model. During the reverse data dependence analysis process, we start from the end of the data stream that variables carry, reverse lookup step-by-step, analyze the change of variable values, and trace the source of faulted variables. This leads to the advantage of our CPSS method in condition fault localization.

5.2.2. Comparison of the Assignment Fault. What causes the assignment fault is the result inconsistent with the expected values because of miscalculation or wrong type of assignment. The fault localization effect of CPSS and SBI methods is shown in Figure 7.

As can be seen from Figure 7, with the allowed detection rate 1%, the ordinate values of CPSS and SBI methods curve are both 0%; that is to say, they are unable to directly and accurately locate the assignment faults. However, when the detection rate reaches 5%, the localization ratio of the CPSS method is quickly increased to 80%, but the SBI method is 40%. It can be roughly considered that the difference of location effect is half. With code detection rate of 15%, CPSS may locate 100% of faults, while SBI method only locates 60%. So it can better reflect the advantages of CPSS method in locating assignment faults. Figure 7 also shows that, in the range of 0%–80% of the code detecting, the CPSS curve has been higher than the SBI method. And only detecting 15% of the code CPSS can fully detect assignment faults, while SBI needs to detect 80% of the code, so CPSS method is obviously better than SBI method in assignment faults location.

TABLE 2: Within the scope of each code to detect the error ratio.

Testing code rate%	CPSS	Tarantula	CT	SBI
1	7.14	6.93	5.26	7.14
10	50.00	47.36	26.36	50.00
20	85.71	63.16	36.84	71.43
30	92.86	71.31	51.17	78.57
40	100.00	79.51	52.72	92.86
50	100.00	86.89	59.70	92.86
60	100.00	87.71	62.80	92.86
70	100.00	88.53	70.55	92.86
80	100.00	92.63	75.20	100.00
90	100.00	100.00	82.18	100.00
100	100.00	100.00	100.00	100.00

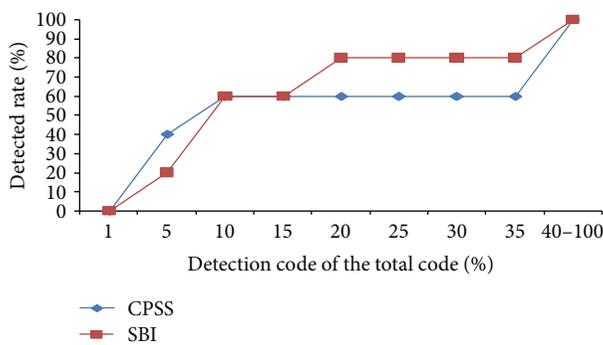


FIGURE 8: Location effect of the function fault.

Abstractly, the program code is only the carrier of the internal data stream, but the control structure is the framework of the program. Essentially, the reason of the assignment fault is that the correct flow of the program data stream goes wrong. Thus, the flow of data changes from the expected values. The location effect of CPSS method better than the SBI method is rooted in: the CPSS method does the control flow analysis and data flow analysis and extracts the data dependencies of the program, before collecting coverage rate and calculating. Discover the cause of faults deeply from the fault occurring, and pick up the code causing faults to search, so there is no doubt that it greatly narrows the scope of the code detection. It can locate the fault conveniently and quickly.

5.2.3. *Comparison of the Function Fault.* Function faults are caused by the difference between the result created by wrong parameter passing and type mismatching in the function call process and the expected values.

From Figure 8, we can see the two methods in the process of locating function faults, having respective advantages in location effect. When the detected code is less than 10%, the CPSS method curve is higher than SBI; that is, the CPSS method is better than SBI. In the stage of 15%–40%, the SBI method is better than CPSS. Figure 8 also shows that, when the detection rate is 1%, faults detected account for 0% of the total number, indicating that neither CPSS nor SBI method

is able to locate faults accurately and directly. With code detection rate of 40%, both curves reach the highest point of 100%, and then curves trend stably parallel to the x -axis; that is, almost all the function faults can be detected with 40% of the code detection rate. All in all, each of the two methods has its advantages in the function fault localization, but when the detection rate is very low, the location effect of CPSS is better than SBI's. A good fault localization method needs to have a low code detection rate but is able to detect a high percentage of faults. According to this evaluation standard, we can loosely think CPSS is little better than SBI.

6. Conclusion

Software fault localization and diagnosis are a critical step in the software debugging. Automated fault localization method can help developers to quickly find the location of the program fault and improve the efficiency of development. In this paper, based on Reverse Data Dependence Analysis Model to extract data dependencies between program statements by analyzing the advantages and disadvantages of the more generally applicable CBFL and program structure slicing method, for shortcomings, we proposed a new fault localization method CPSS while retaining the advantages.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was partially supported by the National Basic Research Program of China (973 Program) under Grant no. 2011CB302605, the National High Technology Research and Development Program of China (863 Program) under Grants nos. 2011AA010705 and 2012AA012506, and the National Science Foundation of China (NSF) under Grants nos. 61173145 and 61202457.

References

- [1] K. Liu, Z. Shan, J. Wang, J. He, Z. Zhang, and Y. Qin, "Overview on major research plan of trustworthy software," *National Natural Science Foundation of China*, vol. 3, pp. 145–151, 2008 (Chinese).
- [2] C. Shen, H. Zhang, D. Feng, Z. Cao, and J. Huang, "Summary of information security," *Science in China*, vol. 37, no. 2, pp. 129–150, 2007 (Chinese).
- [3] B.-X. Fang, T.-B. Lu, and C. Li, "Survey of software assurance," *Journal on Communications*, vol. 30, no. 2, pp. 106–117, 2009 (Chinese).
- [4] W. Zhang, X. Wang, B. Lu, and T.-H. Kim, "Secure encapsulation and publication of biological services in the cloud computing environment," *BioMed Research International*, vol. 2013, Article ID 170580, 8 pages, 2013.
- [5] Z. Zhang, W. K. Chan, T. H. Tse, B. Jiang, and X. Wang, "Capturing propagation of infected program states," in *Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC-FSE '09)*, pp. 43–52, ACM, Amsterdam, The Netherlands, August 2009.
- [6] Y. Yu, J. A. Jones, and M. J. Harrold, "An empirical study of the effects of test-suite reduction on fault localization," in *Proceedings of the 30th International Conference on Software Engineering (ICSE '08)*, pp. 201–210, ACM, New York, NY, USA, May 2008.
- [7] D. Hao, L. Zhang, M.-H. Liu, H. Li, and J.-S. Sun, "Test-data generation guided by static defect detection," *Journal of Computer Science and Technology*, vol. 24, no. 2, pp. 284–293, 2009.
- [8] M. B. Swarup and P. S. Ramaiah, "A software safety model for safety critical applications," *International Journal of Software Engineering and Its Applications*, vol. 3, no. 4, pp. 21–32, 2009.
- [9] P. R. Srivastava and T.-H. Kim, "Application of genetic algorithm in software testing," *International Journal of Software Engineering and Its Applications*, vol. 3, no. 4, pp. 87–96, 2009.
- [10] W. Zhang, H. He, Q. Zhang, and T.-H. Kim, "PhoneProtector: protecting user privacy on the android-based mobile platform," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 282417, 10 pages, 2014.
- [11] M. Beldjehem, "A unified granular fuzzy-neuro framework for predicting and understanding software quality," *International Journal of Software Engineering and Its Applications*, vol. 4, no. 4, pp. 17–36, 2010.
- [12] T. M. Chilimbi, B. Liblit, K. Mehra, A. V. Nori, and K. Vaswani, "Holmes: effective statistical debugging via efficient path profiling," in *Proceedings of the 31st International Conference on Software Engineering (ICSE '09)*, pp. 34–44, IEEE Computer Society, Los Alamitos, Calif, USA, May 2009.

Research Article

Identifying Network Public Opinion Leaders Based on Markov Logic Networks

Weizhe Zhang, Xiaoqiang Li, Hui He, and Xing Wang

School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Weizhe Zhang; wzzhang@hit.edu.cn

Received 20 January 2014; Accepted 26 February 2014; Published 1 April 2014

Academic Editors: T. Cao and F. Yu

Copyright © 2014 Weizhe Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Public opinion emergencies have important effect on social activities. Recognition of special communities like opinion leaders can contribute to a comprehensive understanding of the development trend of public opinion. In this paper, a network opinion leader recognition method based on relational data was put forward, and an opinion leader recognition system integrating public opinion data acquisition module, data characteristic selection, and fusion module as well as opinion leader discovery module based on Markov Logic Networks was designed. The designed opinion leader recognition system not only can overcome the incomplete data acquisition and isolated task of traditional methods, but also can recognize opinion leaders comprehensively with considerations to multiple problems by using the relational model. Experimental results demonstrated that, compared with the traditional methods, the proposed method can provide a more accurate opinion leader recognition and has good noise immunity.

1. Introduction

As the Internet enters into the We-media era, every individual can be a message sender. However, public opinion emergencies may affect the social activities significantly due to the mixed netizen qualities. To avoid adverse effect, it is necessary to have a comprehensive understanding of the development trend of public opinion and recognize special communities like opinion leaders.

The earliest domestic and foreign researches on the discovery of network opinion leaders focused on using the opinion leader theory in the traditional social sciences and research method transplantation directly to recognize the internet leaders but failed to achieve ideal results. They often determined community opinion leaders based on the quantitative data analysis. These methods [1], regardless of the characteristics of objective fact, only depend on logic reasoning and could not represent the public opinion transmission characteristics on the new media of network. Recently, scholars began to explore the difference between network environment and offline practical society through quantitative measurement. The accuracy of the leader discovery model [2–5] witnessed a continuous improvement from the clustering analysis of statistical properties [6] to social

network analysis based on complex network [7–9] and to the semantic analysis of text content.

1.1. Attributes of Public Opinion Participants. Behaviors of every netizen involved in the public opinion transmission are described by inherent attribute, content attribute, and social network attribute.

Inherent attribute refers to the independent attributes of participants from the concerning public opinion events such as career, position, internet age, logins, community credits, fans population, and concerns.

Content attribute refers to the behavioral description of the participants in a certain public opinion event including posts, replies, comments received, reposts, number of mentions, number of words, and emotional tendency.

Social network attribute refers to the mutual relationship of participants in the network mainly including fans and concerns of the participants.

1.2. Shortcomings of Existing Research Methods. Existing network opinion leader discovery is based on the recognition model involving only one or two attributes of participants. No network opinion leader recognition involving all

three attributes of participants has been reported yet. This will affect the accuracy of the opinion leader recognition method.

Existing network opinion leader discovery views the attributes of participants as independent and identically distributed (IID) data. In the theory of probability statistics, mutually independent sequence of variables or other random variables that have the same probability distribution is called IID. However, attribute data of participants are relational data. Different attributes of participants are mutually correlated instead of being independent from each other. For example, fans' population of inherent attribute often is proportional to comments received of content attribute; participants enjoying high attentions from opinion leader are more likely to be the opinion leader. The incomprehensive understanding of such relations will make some opinion leaders remain unidentified. Furthermore, existing network opinion leader discovery has no modeling solution to the relational data.

As a result, simultaneous application of all three attributes and the exploration of relationships in public opinion data can improve the performance of network opinion leader recognition method.

2. Markov Logic Networks

Markov Logic Networks refer to a learning method of statistical relation that is combining the Markov network and first-order logic together. It was proposed by Singla and Domingos [10] in 2004 and then improved by Domingos and his students.

Markov network [11] is a joint distribution model of random variable set ($X = (x_1, x_2, \dots, x_n)$). It is composed of an undirected graph (G) and a potential function (Φ_k) set. Every random variable occupies a node of the graph and each group has a potential function in the model. Potential function is a nonnegative real function, which represents the state of the corresponding group. The joint distribution of Markov network is as follows:

$$P(X = x) = \frac{1}{Z} \prod_k \Phi_k(x_{\{k\}}), \quad (1)$$

where $x_{\{k\}}$ is the state of random variables in the group and Z is partition function (state sum) that is defined as $\sum_{x \in X} \prod_k \Phi_k(x_{\{k\}})$. Weight all characteristic values of potential-use states of each group in the Markov network. Then sum them and calculate the exponentiation. Finally, a log-linear model can be gained as follows:

$$P(X = x) = \frac{1}{Z} \exp \left(\sum_j \omega_j f_j(x) \right). \quad (2)$$

Characteristic function can be any real function of state. In this paper, characteristic function refers to the dual characteristics value ($\{0, 1\}$). Equation (1) is the most direct expression of potentials, in which every possible state of each group has a corresponding characteristic value and a weight. Equation (1) is related to the power of groups.

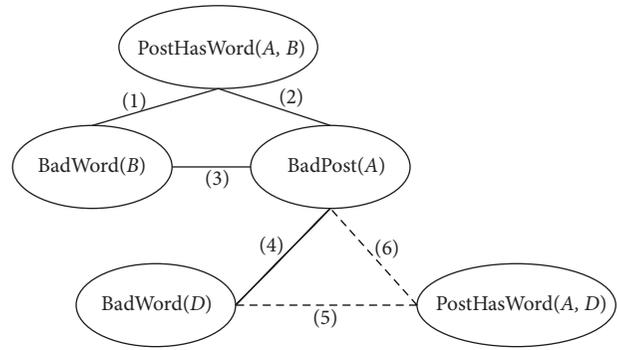


FIGURE 1: Closed Markov Logic Network.

However, the amount of characteristic values can be reduced freely by some methods (e.g., logic function of state), thus enabling the characteristic function to provide a simpler expression under large number of groups compared with the potential functions. This is the principle of the Markov Logic Networks.

Markov Logic Networks are a first-order logic knowledge base where every code has a weight. This first-order logic knowledge base can be viewed as the template of Markov Logic Networks. Viewed from probability, the Markov Logic Networks provide a simple language to define large Markov network as well as a flexible and modularized integration with abundant knowledge. Viewed from the first-order logic, the Markov Logic Networks provide a sound processing to knowledge base with uncertainties, defects, and even contradictories, thus decreasing the vulnerability.

Take the data set of Skyline for example. The simplest situation is as follows: suppose the knowledge base only contains the formula F_1 (weight = 1.5), and the corresponding MLN of the knowledge base is $\{(F_1, 1.5)\}$:

$$F_1 : \text{PostHasWord}(y, x) \rightarrow (\text{BadWord}(x) \rightarrow \text{BadPost}(y)), \quad (3)$$

where x and y are individual variables; $\text{BadWord}(x)$, $\text{BadPost}(y)$, and $\text{PostHasWord}(y, x)$ are predicates, representing whether x has bad word, y is a bad post, and y contains x . F_1 means that if y has the bad word x , y can be deduced as a bad post.

Given an individual constant set $C = \{A, B, D\}$, a closed Markov Logic Network can be generated (Figure 1).

2.1. Markov Logical Reasoning Method. Markov logical reasoning is equal to the probabilistic reasoning of the complex relationship. The basic task of reasoning is to inference the most possible state of world y according to the given evidence x (word set).

There are two basic types of reasoning: First, we search the most possible state satisfying some evidence and probability of computer random condition. Next, lazy reasoning and relieving reasoning are compared in improving the performance of above two types of reasoning in processing more complicated relationships. The lazy reasoning only requires

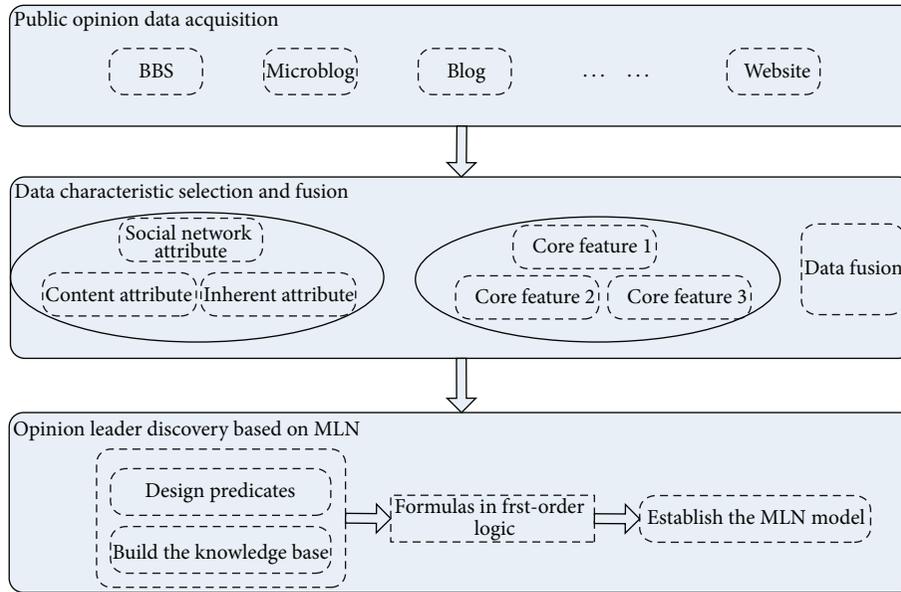


FIGURE 2: Overall structure of the network opinion leader recognition system based on the Markov Logic Networks.

adopting instantiation “default” value to the deviated basic values, while relieving reasoning divides indistinctive atoms into one group and views them as an independent unit.

2.2. Markov Logic Networks Learning. The Markov Logic Networks learning includes structure learning and parameter learning. Structure learning is to learn the model structure (network structure of the Markov Logic Networks) from data. In structure learning, it is more difficult to learn the rules. There are two structure learning methods based on the inductive logic programming (ILP): (1) learn model structure by using CLAUDIEN (an LP system) directly; (2) learn structure of Markov Logic Networks from the relational database by combining LP and feature introduction in Markov network together. Parameter learning is committed to find the satisfying rule weight with maximum likelihood or maximum conditional likelihood of associated database. Parameter learning includes production parameter learning and discriminant parameter learning. Parameter learning is needed under known structure of Markov network to acquire weight of each rule. However, when the structure of Markov network is unknown, structure learning is needed before the parameter learning to get the network structure.

Markov Logic Networks learning includes parameter learning and structure learning. Parameter learning has production parameter learning and discriminant parameter learning as well as their corresponding formulas and approximate algorithms. Structure learning also includes top-down structure learning and bottom-up structure learning.

3. System Framework

The overall structure of the network opinion leader recognition system based on the Markov Logic Networks is shown in Figure 2.

The designed network opinion leader recognition system includes three modules: public opinion data acquisition module, data characteristic selection and fusion module, and opinion leader discovery module based on Markov Logic Networks. The public opinion data acquisition module is for data collection concerning specific public opinion event. Data characteristic selection and fusion module is for processing and analyzing collected data to disclose the relationship between core characteristics and attributes. The opinion leader discovery module based on Markov Logic Networks is to design predicates, build the knowledge base, and establish the MLN model according to the relationship between core characteristics and attributes.

3.1. Opinion Leader Discovery Module Based on Markov Logic Networks. The technical route of opinion leader recognition based on Markov Logic Networks is presented in Figure 3. The top data input, initialization module, and structure learning module are the training module of the model, whereas the bottom data (Leader) verification, verification module, and data output are the verification module of the model.

The primary task of the training module is to design predicate. Predicate design has two stages: (1) original design of initial predicate set according to the characteristic matrix and relational knowledge gained by the data characteristic extraction and fusion module as well as personal priori knowledge; (2) repetitive reasonable adjustment of predicate design according to the learning results of the model and follow-up experimental analysis of learning results until getting satisfying experimental result.

The initialization module will convert contents in the corpus into DB according to existing predicate design. The structure learning module can conduct the structure learning through the available learnstruct program of

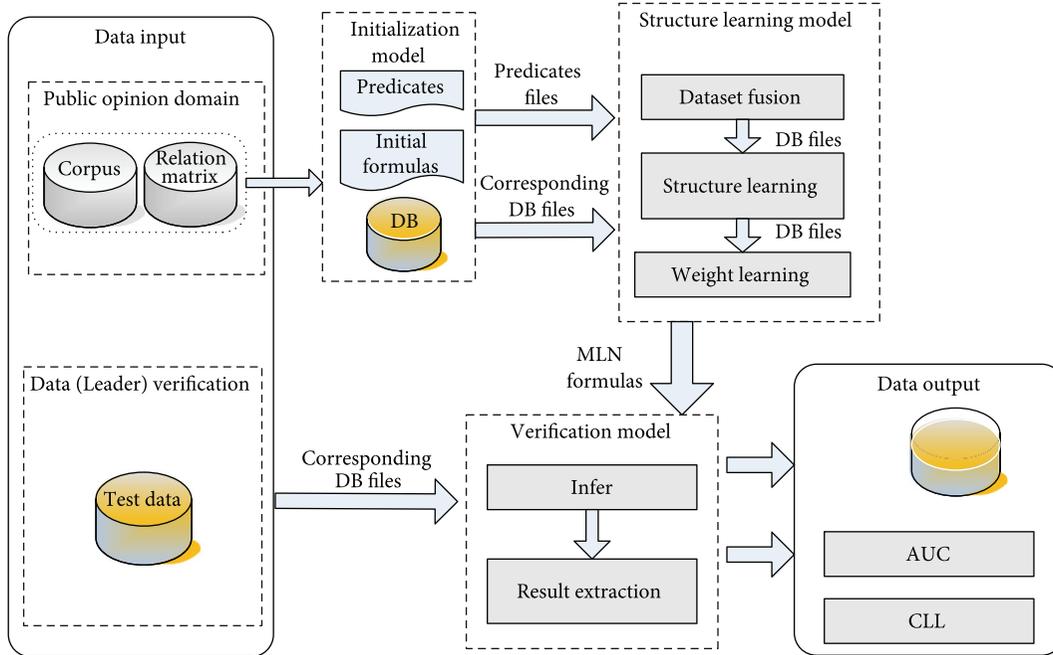


FIGURE 3: Technical route of opinion leader recognition based on Markov Logic Networks.

Alchemy. Beam search is the default structure learning algorithm. Weight learning can be implemented by the available learning program. Weighted MLN clause can be gained through the structure learning and weight learning, which will be used for reasoning. In the designed system, these weighted MLN clauses are used to deduce user's identity.

In the verification module of the model, both verification data and training data are converted into DB according to the predicate design. The verification module is mainly used to deduce user's identity.

Reasoning results contain all possible predicates and their possibilities. Take the predicate of *teacherby(course, teacher)* for instance. Course represents course and teacher represents teacher, meaning that the course is taught by the teacher. If the course is Chinese, possible teacher set is {Tom, Jack, Lily}. Then, the reasoning results may be as follows:

$$\begin{aligned}
 &\text{teacherby}(\text{Chinese}, \text{Tom}) \quad 0.9275 \\
 &\text{teacherby}(\text{Chinese}, \text{Jack}) \quad 0.7465 \\
 &\text{teacherby}(\text{Chinese}, \text{Lily}) \quad 0.3556.
 \end{aligned}
 \tag{4}$$

Result extraction means to select the reasoning result with highest probability as the final result. In the above case, Tom will be selected as the teacher of Chinese. The data output module will calculate AUC and CLL.

4. Experimental Verification

4.1. Recognition of Nonrelational Data Model. Netizens were classified according to the nonrelational data model provided by Weka. First of all, the original data have to be converted

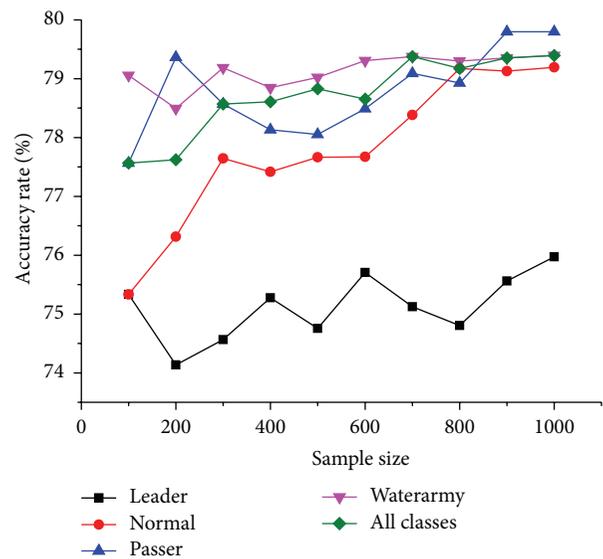


FIGURE 4: Group recognition of nonrelational data model.

into ARFF. ARFF mainly includes attribute assertion and data [12, 13].

We chose SVM to classify netizens involved in the “Xu-Ting Event” on the legal forum of Skyline Gossip as Leader, Normal, Passer, and Waterarmy. In our experiment, every netizen was recognized independently. Experimental results are presented in Figure 4.

It can be seen from Figure 4 that the nonrelational data model achieved a recognition accuracy of about 77%, which presented a continuous growth as the sample size increases.

TABLE 1: Designed predicates of content attribute.

Predicate	Meaning
Post(<i>person_id</i> , <i>post_id</i>)	User who is represented by <i>person_id</i> published a post which is represented by <i>post_id</i> .
ReplyNumOfPost(<i>post_id</i> , <i>level_replynumofpost</i>)	The reply number of the post which is represented by <i>post_id</i> is <i>level_replynumofpost</i> .
ClickNumOfPost(<i>post_id</i> , <i>level_clicknumofpost</i>)	The click number of the post which is represented by <i>post_id</i> is <i>level_clicknumofpost</i> .
TotalPostNum(<i>person_id</i> , <i>level_totalpostnum</i>)	The post number of the user who is represented by <i>person_id</i> is <i>level_totalpostnum</i> .
TotalReplyNum(<i>person_id</i> , <i>level_totalreplynum</i>)	The reply number of the user who is represented by <i>person_id</i> is <i>level_totalreplynum</i> .
TotalBeReplyNum(<i>person_id</i> , <i>level_totalbereplynum</i>)	The number of replies to the user who is represented by <i>person_id</i> is <i>level_totalbereplynum</i> .
Correlation(<i>person_id</i> , <i>post_id</i> , <i>level_correlation</i>)	The correlation level between the user who is represented by <i>person_id</i> published content in the post which is represented by <i>post_id</i> and post topics is <i>level_correlation</i> .
Sentiment(<i>person_id</i> , <i>level_sentiment</i>)	The degree of the emotional tendencies bases on the content published by the user who is represented by <i>person_id</i> is <i>level_sentiment</i> .

TABLE 2: Designed predicates of social network attribute.

Predicates	Meaning
FansNum(<i>person_id</i> , <i>level_fansnum</i>)	The fans number of the user who is represented by <i>person_id</i> is <i>level_fansnum</i> .
FollowNum(<i>person_id</i> , <i>level_follownum</i>)	The follow number of the user who is represented by <i>person_id</i> is <i>level_follownum</i> .
Follow(<i>person_id</i> , <i>person_id</i>)	A user who is represented by the first <i>person_id</i> follows another user who is represented by the second <i>person_id</i> .
Reply(<i>person_id</i> , <i>person_id</i> , <i>post_id</i> , <i>num</i>)	In the post which is represented by the <i>post_id</i> , the number of a user who is represented by the first <i>person_id</i> reply to another user who is represented by the second <i>person_id</i> is <i>num</i> .

TABLE 3: Designed predicates of inherent attribute.

Predicate	Meaning
Gender(<i>person_id</i> , <i>gender</i>)	The gender of the user who is represented by <i>person_id</i> is <i>gender</i> .
Age(<i>person_id</i> , <i>level_age</i>)	The age of the user who is represented by <i>person_id</i> is <i>level_age</i> .
NetworkAge(<i>people</i> , <i>level_networkage</i>)	The network age of the user who is represented by <i>person_id</i> is <i>level_networkage</i> .
LogNum(<i>person_id</i> , <i>level_lognum</i>)	The login number of the user who is represented by <i>person_id</i> is <i>level_lognum</i> .
CommunityCredits(<i>person_id</i> , <i>level_communitycredits</i>)	The community credits of the user who is represented by <i>person_id</i> is <i>level_communitycredits</i> .
HasPosition(<i>person_id</i>)	The user who is represented by <i>person_id</i> has a communities position.
Role(<i>person_id</i> , <i>role</i>)	The role of the user who is represented by <i>person_id</i> is <i>role</i> .

4.2. Recognition of Relational Data Model. Firstly, we have to design predicates. Our designed predicates according to the characteristic selection and personal priori knowledge are listed in Tables 1, 2, and 3, including three classes (social network attribute, content attribute, and inherent

attribute). These three classes of attribute are used to describe different network behaviors and individual characteristics of netizens.

After the predicate design, we have to convert original data into DB.

```

follow(person_id, person_id)
reply(person_id, person_id)
post(person_id, post_id)
replynumofpost(post_id, level_replynumofpost)
clicknumofpost(post_id, level_clicknumofpost)
totalpostnum(person_id, level_totalpostnum)
totalreplynum(person_id, level_totalreplynum)
totalbereplynum(person_id, level_totalbereplynum)
act(person_id, role)
repeat(person_id, level_repeatnum)

```

Box 1

```

10.4475 totalbereplynum(a1,a2)
10.7128 clicknumofpost(a1,a2)
10.4475 totalreplynum(a1,a2)
-6.70724 replynumofpost(a1,a2) v !replynumofpost(a1,a3)
5.77628 !repeat(a1,a2) v !repeat(a1,a3)
5.77628 !totalbereplynum(a1,a2) v !totalbereplynum(a1,a3)
5.83318 !totalpostnum(a1,a2) v !totalpostnum(a1,a3)
5.77628 !totalreplynum(a1,a2) v !totalreplynum(a1,a3)
5.83318 !clicknumofpost(a1,a2) v !clicknumofpost(a1,a3)
4.7222 !post(a1,a2) v !post(a3,a2)
5.83318 !act(a1,a2) v !act(a1,a3)

```

Box 2

TABLE 4: Valuable clauses learned from the “Xu-Ting Event”.

Weight	Formula
2.76133	!fansnum(a1,Level_fansnum_10To49) v !follownum(a1,Level_follownum_Lessthan10) v gender(a1,Male) v !networkage(a1,Level_networkage) v !lognum(a1,Level_log_num_1000To4999) v !communitycredits(a1,Level_communitycredits_1000To9999)
2.63516	gender(a1,Female) v !lognum(a1,Level_log_num_1000To4999)
3.21897	!fansnum(a1,Level_fansnum_10To49) v !follownum(a1,Level_follownum_Lessthan10) v age(a1,Level_realage_Morethan35) v !networkage(a1,Level_networkage) v !lognum(a1,Level_log_num_1000To4999) v !communitycredits(a1,Level_communitycredits_1000To9999)
3.77246	gender(a1,Female) v !lognum(a1,Level_log_num_Lessthan1000)
4.27395	!reply(a1,a2) v act(a2,Leader) v !act(a1,Leader)
5.65332	gender(a1,a2) v !age(a1,a3) v !age(a1,a4) v lognum(a1,a5) v lognum(a1,a6)
6.06442	!communitycredits(a1,a2) v !communitycredits(a1,a3) v !totalreplynum(a1,a4) v !totalreplynum(a1,a5)
6.06101	!networkage(a1,a2) v !networkage(a1,a3)

```

//Waterarmy
repeat(a1,+a2)=>act(a1,+a3)
//Leader
totalreplynum(a1,+a2)^totalpostnum(a1,+a3)^totalbereplynum(a1,+a4)=>act(a1,+a5)
post(a1,a2)^replynumofpost(a2,+a3)^clicknumofpost(a2,+a4)=>act(a1,+a5)
//Try to take advantage of relational data
act(a1,+a2)^follow(a1,a3)=>act(a3,+a4)
act(a1,+a2)^reply(a1,a3)=>act(a3,+a4)
//Passer
//totalreplynum(a1,+a2)^totalpostnum(a1,+a3)^totalbereplynum(a1,+a4)=>act(a1,+a5)

```

Box 3

```

5.8819 act(a1,Waterarmy) v !repeat(a1,Level_repeatnum_20To49)
5.28195 act(a1,Waterarmy) v !repeat(a1,Level_repeatnum_Morethan50)
4.26896 !totalpostnum(a1,Level_totalpostnum_2To2)
      v !totalreplynum(a1,Level_totalreplynum_10To49)
      v !totalbereplynum(a1,Level_totalbereplynum_2To9)
      v act(a1,Normal)
4.93641 !totalpostnum(a1,Level_totalpostnum_Lessthan2)
      v !totalreplynum(a1,Level_totalreplynum_Lessthan2)
      v !totalbereplynum(a1,Level_totalbereplynum_Lessthan2)
      v act(a1,Passer)
3.76921 !totalpostnum(a1,Level_totalpostnum_Lessthan2)
      v !totalreplynum(a1,Level_totalreplynum_Lessthan2)
      v !totalbereplynum(a1,Level_totalbereplynum_Morethan200)
      v act(a1,Leader)
3.74401 !reply(a1,a2) v act(a2,Normal) v !act(a1,Passer)
4.34903 !reply(a1,a2) v act(a2,Leader) v !act(a1,Leader)
    
```

Box 4

act(Person15,Waterarmy) 0.505	act(Person15,Normal) 0.813969
act(Person15,Leader) 0.495	act(Person15,Passer) 0.99995
act(Person30,Waterarmy) 0.99995	act(Person30,Normal) 0.99995
act(Person30,Leader) 0.99995	act(Person30,Passer) 0.894961
act(Person21,Waterarmy) 0.495	act(Person21,Normal) 0.823968
act(Person21,Leader) 0.504	act(Person21,Passer) 0.99995
act(Person55,Waterarmy) 0.99995	act(Person55,Normal) 0.99995
act(Person55,Leader) 0.99995	act(Person55,Passer) 0.893961
act(Person51,Waterarmy) 0.490001	act(Person51,Normal) 0.794971
act(Person51,Leader) 0.513999	act(Person51,Passer) 0.99995

Box 5

Next, we have to implement structure learning and weight learning. The input MLN file (predicate statements) for structure learning is shown as Box 1.

And the structure learning results are shown as Box 2.

The input MLN file (predicate statements and design statements) for weight learning is shown as Box 3.

The design statements are recognition results of four groups.

The weight learning results are represented as Box 4.

Table 4 lists some learned valuable clauses such as the following:

```

4.27395 !reply(a1,a2) v act(a2,Leader) v
!act(a1,Leader).
    
```

This clause means that if a1 is the leader and a1 replies a2, then a2 is a leader too. This is true in real life.

The learned valuable clauses were selected for reasoning of the test set. According to the experimental results (shown as Box 5), Person15, Person30, Person21, Person55, and Person51 were identified as Passer, Normal, Passer, Leader, and Passer, respectively. Four were recognized accurately and 1 was recognized wrongly.

The recognition accuracy comparison results of relational data model and nonrelational data model to different events

are listed in Table 5, finding that the relational data model has higher recognition accuracy compared with the nonrelational data model.

5. Conclusions

This paper firstly summarizes and evaluates the shortcomings of existing opinion leader recognition method, describes the advantages of Markov Logic Networks in opinion leader recognition, and summarizes the associated theories of Markov Logic Networks including basic concepts as well as theoretical models (reasoning and learning). The Markov Logic Networks combine the probability theory and first-order logic perfectly, integrating logic/relation expressions, uncertainty processing, and learning. Secondly, this paper designs and implements a network opinion leader recognition system based on previous theories. This designed system firstly collects some public opinion data as the training set for structure learning of Markov Logic Networks and then uses the learning results to reasoning the control results of corresponding public opinion domain of the test data. The experimental results are compared and analyzed to evaluate their validity. Thirdly, this paper carries out an experimental verification, which verifies

TABLE 5: Recognition accuracy comparison of relational data model and non-relational data model.

Event name	Forum	The accuracy of IID model (%)	The accuracy of relation model (%)
Xu-Ting Event	Legal Forum	79.5	82.5
Xu-Ting Event	Tianya By-talk	77.4	80.6
Three years of great Chinese famine	Discussion about the history	77.8	81.8
Three years of great Chinese famine	Tianya By-talk	76.9	80.8

the superiority of the designed network opinion leader recognition system.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant no. 61173145, the National Basic Research Program of China under Grant no. G2011CB302605, and the National High Technology Research and Development Program of China under Grant no. 2011AA010705.

References

- [1] Y. Li, S. Ma, Y. Zhang, R. Huang, and Kinshuk, "An improved mix framework for opinion leader identification in online learning communities," *Knowledge-Based Systems*, vol. 43, pp. 43–51, 2013.
- [2] R. Goldsborough, "The influence of active online users," *Black Issues in Higher Education*, vol. 19, no. 5, pp. 30–31, 2002.
- [3] N. Matsumura, Y. Ohsawa, and M. Ishizuka, "Mining and characterizing opinion leaders from threaded online discussions," in *Proceedings of the 6th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, 2002.
- [4] N. Matsumura, "Topic diffusion in a community," in *Chance Discovery*, pp. 84–97, Springer, Berlin, Germany, 2003.
- [5] N. Matsumura, "Collaborative communication strategies in online community," in *The 4th International Workshop on Social Intelligence Design (SID '05)*, 2005.
- [6] Z. Zhai, H. Xu, and P. Jia, "Identifying opinion leaders in BBS," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT '08)*, vol. 3, pp. 398–401, IEEE, December 2008.
- [7] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.
- [8] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, 1999.
- [9] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.
- [10] P. Singla and P. Domingos, "Markov logic in infinite domains," In press, <http://arxiv.org/abs/1206.5292>.
- [11] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.
- [12] W. Zhang, H. He, and J. Ye, "A two-level cache for distributed information retrieval in search engines," *The Scientific World Journal*, vol. 2013, Article ID 596724, 6 pages, 2013.
- [13] W. Zhang, B. Wang, H. He, and Z. Tan, "Public opinion leader community mining based on the heterogeneous network," *Acta Electronica Sinica*, vol. 40, no. 10, pp. 1927–1932, 2012.

Research Article

On the Improvement of Wiener Attack on RSA with Small Private Exponent

Mu-En Wu,¹ Chien-Ming Chen,^{2,3} Yue-Hsun Lin,⁴ and Hung-Min Sun⁵

¹ Department of Mathematics, Soochow University, Taipei, Taiwan

² School of Computer Science and Technology, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China

³ Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen, China

⁴ CyLab, Carnegie Mellon University, Pittsburgh, PA 15213, USA

⁵ Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

Correspondence should be addressed to Hung-Min Sun; hmsun@cs.nthu.edu.tw

Received 7 February 2014; Accepted 27 February 2014; Published 27 March 2014

Academic Editors: T. Cao and F. Yu

Copyright © 2014 Mu-En Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

RSA system is based on the hardness of the integer factorization problem (IFP). Given an RSA modulus $N = pq$, it is difficult to determine the prime factors p and q efficiently. One of the most famous short exponent attacks on RSA is the Wiener attack. In 1997, Verheul and van Tilborg use an exhaustive search to extend the boundary of the Wiener attack. Their result shows that the cost of exhaustive search is $2r + 8$ bits when extending the Wiener's boundary r bits. In this paper, we first reduce the cost of exhaustive search from $2r + 8$ bits to $2r + 2$ bits. Then, we propose a method named EPF. With EPF, the cost of exhaustive search is further reduced to $2r - 6$ bits when we extend Wiener's boundary r bits. It means that our result is 2^{14} times faster than Verheul and van Tilborg's result. Besides, the security boundary is extended 7 bits.

1. Introduction

During the past 30 years, RSA [1] has been one of the most popular public-key cryptosystems worldwide. It has been widely used in several applications [2–4]. The security of RSA is often based on the hardness of the integer factorization problem (IFP), which remains a well-studied problem [5, 6]. Current RSA standards suggest that an RSA modulus N should be at least 1024 bits long. Using the best-known factoring algorithms, the expected workload of factoring a 1024 bit modulus is 2^{80} , which is currently believed to be infeasible. However, although the use of a large RSA modulus achieves a high security level, the encryption and decryption procedures involve heavy exponential modular multiplications, which make RSA inefficient. Therefore, many approaches have been investigated for speeding-up the RSA encryption (or signature-verification) and RSA decryption (or signature-signing) [7–12]. Furthermore, since the signing

task is often executed by lightweight devices, such as smart cards, mobile phones, or PDAs, the research on speeding-up signature-signing is more practical and important.

The most popular method for reducing the signing time is to apply a small private exponent d since the complexity of signing depends on the bit-length of d . In order to achieve this goal, the order of choosing e and d is exchanged. d is first chosen in the RSA-key generation algorithm, and the corresponding public exponent e satisfying $ed \equiv 1 \pmod{\varphi(N)}$ is then calculated. These RSA variants are called RSA-Small- d . Nevertheless, the variants of RSA-Small- d have the security flaws [13–18]. In fact, instances of RSA with $d < N^{1/4}$ can be efficiently broken by Wiener attack [16]. Besides, Boneh and Durfee's lattice-based attack [19] indicated that an instance of RSA-Small- d with $d < N^{0.292}$ should be considered to be an unsafe system.

In 1997, Verheul and van Tilborg [20] used an exhaustive search to further extend the boundary of the Wiener attack.

Suppose $r = \log_2 d - \log_2 N^{1/4}$; their result shows that an exhaustive search for $2r + 8$ bits is required to extend the Wiener's boundary r bits. Assume that an exhaustive search for 64 bits is feasible in terms of current computational abilities; solving r for the equation " $2r + 8 = 64$ " yields $r = 28$, which implies that the boundary of the Wiener attack should be raised up to $N^{1/4} 2^{28}$.

In this paper, we attempt to reduce the cost of exhaustive search of Verheul and van Tilborg's result. We propose an approach to reduce the cost of exhaustive search when we desire to extend Wiener's boundary. This approach includes two steps.

Step 1. We investigate a method for searching as many MSBs (most significant bits) of $p + q$ as possible, which is equivalent to estimating $p + q$ as accurately as possible. In this step, to extend Wiener's boundary r bits, an exhaustive search requires $2r + 2$ bits. It means that our result is better than Verheul and van Tilborg's cost, which requires an exhaustive search for $2r + 8$ bits.

Step 2. We develop an approach, called "Estimated Prime Factor (EPF)," to estimate $p + q$, and then we derive two integers p_E and q_E , which are the estimations of p and q , respectively. Using EPF, the first 8 MSBs of $p + q$ can be determined. This result is more accurate than the traditional estimation, which estimates $p + q$ by $2\sqrt{N}$. Applying EPF can further reduce the cost of exhaustive search. More specifically, to extend Wiener's boundary r bits, an exhaustive search requires $2r - 6$ bits. As compared to Verheul and van Tilborg's result, which requires an exhaustive search for $2r + 8$ bits, the security boundary is extended 7 bits.

1.1. Our Contribution. The contributions of this paper are summarized as follows.

- (1) We first reduce the cost of exhaustive search from $2r + 8$ (Verheul and van Tilborg's result) bits to $2r + 2$ bits when we extend Wiener's boundary r bits. It means that exhaustive search is 2^6 times faster in Step 1. Besides, the security boundary is extended 3 bits.
- (2) We propose a novel approach, named EPF, for estimating the prime factors of N . With EPF, the cost of the exhaustive search for $2r + 2$ bits (mentioned in contribution (1)) is further reduced to $2r - 6$ bits. Compared with Verheul and van Tilborg's result, exhaustive search is 2^{14} times faster. Besides, the security boundary is extended 7 bits.

1.2. Organization. The remainder of this paper is organized as follows. Section 2 presents the preliminaries of this paper. Section 3 describes Step 1 of our approach. In Section 4, we propose the EPF to estimate the prime factors of an RSA modulus. Next, Step 2 of our approach which is applying EPF is proposed in Section 5. Finally, we present our conclusions and future works in Section 6.

2. Preliminary

In this section, we introduce the preliminaries of this paper which include RSA and its variants and the Wiener attack.

2.1. RSA and Its Variants. The RSA cryptosystem [1] consists of three parts, RSA-key generation, encryption, and decryption which are described as follows.

2.1.1. RSA-Key Generation, Encryption, and Decryption. The RSA-key generation outputs the RSA key: (N, e, d) . First, randomly choose two large prime numbers p and q and compute $N = pq$, where N is called RSA modulus. Secondly, let e , called public exponent, be a randomly chosen integer such that $\gcd(e, \varphi(N)) = 1$, where $\varphi(\cdot)$ is Euler's phi function. Then, let d , called private exponent, be the multiplicative inverse modulo $\varphi(N)$ (i.e., $ed \equiv 1 \pmod{\varphi(N)}$). The pair (e, N) is the public key and the pair (d, N) is the private key.

From the key relation $ed \equiv 1 \pmod{\varphi(N)}$, there exists a unique positive integer k satisfying

$$ed = 1 + k \cdot \varphi(N). \quad (1)$$

We call (1) as the RSA-key equation. To encrypt a plaintext message $M \in \mathbb{Z}_N$, compute $C \equiv M^e \pmod{N}$. The result C is called the ciphertext of M . To execute RSA decryption, a ciphertext $C \in \mathbb{Z}_N$ is decrypted by raising it to the d th power modulo N . From Lagrange's theorem, it follows that

$$C^d \pmod{N} = M^{ed} \pmod{N} \equiv M \pmod{N} = M. \quad (2)$$

Usually, one often selects e as small as possible due to the reason of efficient encryption (or signature-verification). The smallest e is suggested to be $2^{32} + 1$ rather than $2^{16} + 1$ while a known affine relation between two messages exists [21]. We call the RSA system with small public exponent e as "RSA-Small- e ." On the other hand, since the cost of decryption (or signature-signing) can be significantly reduced when the private exponent d is much smaller than $\varphi(N)$, in order to simply reduce the decryption (or signature-signing) time, one can select a small private exponent d first in RSA-key generation. Such variant is called RSA-Small- d , which is shown in the following.

2.1.2. RSA-Small- d . Generating instances of RSA with a small private exponent is easy with the observation that the RSA-key equation (1) is symmetric with respect to the public and private exponents. We simply follow the same key generation of original RSA but exchange the choosing order of public and private exponents.

One of the drawbacks of RSA-Small- d is its inefficient encryption. Since the public exponent e in RSA-Small- d is always computed as the inverse of d modulo $\varphi(N)$, it is expected with high probability that e will be almost the same size as $\varphi(N)$. In conclusion, RSA-Small- d saves the decryption (or signature) cost while the encryption cost remains large.

2.2. *The Wiener Attack.* One of the most famous short exponent attacks on RSA is the Wiener attack. Boneh and Durfee [22] showed in 1990 that RSA-Small- d should be considered insecure when $d < N^{1/4}$. He achieved the attack through the technique of continued fractions. In the following paragraph, we briefly introduce the continued fractions and the Wiener attack. The details can be referenced in [16].

Definition 1 (continued fractions). For any positive real number α , define $\alpha = \xi_0, a_i = \lfloor \xi_i \rfloor, \xi_{i+1} = 1/(\xi_i - a_i)$ for $i = 0, 1, 2, \dots$. Then α can be expanded into the following form:

$$\alpha_i = a_0 + 1/(a_1 + 1/(a_2 + 1/(a_3 + 1/\dots))). \quad (3)$$

The form of (3) is called the continued fraction expression of α . For simplicity, we write (3) to be $\alpha = (a_0, a_1, a_2, \dots)$. In addition, denote $\alpha_i = (a_0, a_1, \dots, a_i)$ as the i th convergent of the continued fraction expansion of α , which means

$$\alpha_i = a_0 + 1/(a_1 + 1/(a_2 + 1/(\dots + 1/a_i))). \quad (4)$$

If α is a rational number, then the process of computing its continued fraction expression, see (3), will cease in some index k . That is, $\alpha = \alpha_k$. If α is irrational, then the process will go on unceasingly.

Theorem 2. Denote h_i/k_i as the fraction form of (4); that is, $h_i/k_i = \alpha_i$, where h_i and k_i are positive integers. Then, h_i and k_i can be calculated by defining $h_{-2} = 0, k_{-2} = 1, h_{-1} = 1,$ and $k_{-1} = 0$. And $h_i = a_i h_{i-1} + h_{i-2}$ and $k_i = a_i k_{i-1} + k_{i-2}$, for $i \geq 0$.

Following the notations in Theorem 2, we have Corollary 3.

Corollary 3. For any $i \geq 1$,

$$\left| \alpha - \frac{h_{i+1}}{k_{i+1}} \right| < \left| \alpha - \frac{h_i}{k_i} \right|. \quad (5)$$

Furthermore, if α is an irrational number, then $\lim_{i \rightarrow \infty} h_i/k_i = \alpha$.

Theorem 4. If a real number α and a reduced fraction a/b satisfy

$$\left| \alpha - \frac{a}{b} \right| < \frac{1}{2b^2}, \quad (6)$$

then a/b equals to one of the convergents of the continued fraction expression of α .

2.2.1. *The Wiener Attack.* The Wiener attack [16] is based on approximations using continued fractions to find the private exponent of RSA-Small- d in polynomial time if $d < N^{1/4}$, where p and q are of the same bit-length. Note that the RSA-key equation, $ed = 1 + k \cdot \varphi(N)$, can be rewritten as

$$\left| \frac{e}{\varphi(N)} - \frac{k}{d} \right| = \left| \frac{1}{d\varphi(N)} \right|, \quad (7)$$

which is similar to the form of the left-hand side of (6). In order to apply Theorem 4, we replace $e/\varphi(N)$ of (7) by e/N , which is known for everyone, and set the difference between e/N and k/d to be smaller than $1/2d^2$; that is,

$$\left| \frac{e}{N} - \frac{k}{d} \right| < \frac{1}{2d^2}. \quad (8)$$

Therefore, according to Theorem 4, k/d can be found by computing one of the convergents of the continued fraction expression of e/N .

The security boundary of the Wiener attack is deduced from the sufficient condition for (8). Since $p \approx q \approx \sqrt{N}$ and $k \approx d$, the left-hand side of (8) is simplified to

$$\left| \frac{e}{N} - \frac{k}{d} \right| = \frac{k(p+q-1)-1}{Nd} \approx \frac{2\sqrt{N}}{N} = \frac{2}{\sqrt{N}}. \quad (9)$$

Hence, (8) is transformed to

$$\frac{2}{\sqrt{N}} < \frac{1}{2d^2}, \quad (10)$$

which gives the security boundary of the Wiener attack (after ignoring the constant term):

$$d < N^{1/4}. \quad (11)$$

2.3. *Verheul and van Tilborg's Extension.* The Wiener attack works very well and efficiently when the private exponent $d < N^{1/4}$. However, what about if the bit-length of d is slightly larger than the bit-length of $N^{1/4}$? In 1997, Verheul and van Tilborg [20] proposed a technique to solve this problem by performing an exhaustive search for $2r + 8$ bits, where $r = \log_2 d - \log_2 N^{1/4}$ means that the bit-length of d is longer than the bit-length of $N^{1/4}$ by r bits.

Verheul and van Tilborg observed that k/d in (8) can be represented as follows:

$$\frac{k}{d} = \frac{p_{j+1}U + (U\Delta + V)p_j}{q_{j+1}U + (U\Delta + V)q_j}, \quad (12)$$

where p_i/q_i is the i th convergent of the continued fraction expression of e/N , $\Delta = 1$ or 2 , and U and V are two unknown integers with upper bounds as follows:

$$\log_2 U \leq r + 4, \quad \log_2 V \leq r + 4. \quad (13)$$

Since Δ is a small integer, we can omit its uncertainty. The unknown parts of (12) are about $2r + 8$ bits, which give the result of Verheul and van Tilborg's extension: extending Wiener's boundary by r bits requires an exhaustive search for about $2r + 8$ bits.

Assume that an exhaustive search for 64 bits is feasible in terms of the current computational capabilities. Solving r for the equation " $2r + 8 = 64$ " yields $r = 28$, which implies that Wiener's boundary can be extended 28 bits over the bit-length of $N^{1/4}$. Therefore, RSA-Small- d with $d < N^{1/4}2^{28}$ can be totally broken by continued fraction attack plus the cost of

performing an exhaustive search for 64 bits. In Section 3, we show that, in order to extend Wiener’s boundary by r bits, it requires only an exhaustive search for $2r + 2$ bits, rather than that from Verheul and van Tilborg’s extension for cost, which requires an exhaustive search for $2r + 8$ bits.

3. Reducing the Cost of Exhaustive Search to $2r+2$ Bits

Our approach contains two steps which are described in Sections 3 and 5, respectively. In this section, we investigate a method for searching as many MSBs (most significant bits) of $p + q$ as possible, which is equivalent to estimating $p + q$ as accurately as possible. With this method, we can reduce the cost of exhaustive search from $2r + 8$ bits (Verheul and van Tilborg’s extension) to $2r + 2$ bits when we extend Wiener’s boundary r bits.

Let A be the estimation of $p + q$. Throughout this paper, we assume $A < p + q$. Thus $\varphi(N) = (N + 1) - (p + q)$ is estimated as $(N + 1) - A$, which implies

$$\frac{e}{\varphi(N)} \approx \frac{e}{(N + 1) - A}. \tag{14}$$

Applying (14) to the Wiener attack, that is, replacing e/N of (8) by $e/((N + 1) - A)$, we have

$$\left| \frac{e}{N + 1 - A} - \frac{k}{d} \right| < \frac{1}{2d^2}. \tag{15}$$

Note that if $A = p + q$, then (15) always holds for any d because

$$\begin{aligned} \left| \frac{e}{N + 1 - (p + q)} - \frac{k}{d} \right| &= \left| \frac{ed - k(N + 1 - (p + q))}{(N + 1 - (p + q))d} \right| \\ &= \frac{1}{\varphi(N)d} < \frac{1}{2d^2}. \end{aligned} \tag{16}$$

Simplifying (15) yields

$$\begin{aligned} \left| \frac{e}{N + 1 - A} - \frac{k}{d} \right| &= \left| \frac{ed - k(N + 1 - A)}{(N + 1 - A)d} \right| \\ &= \frac{k[(p + q) - A] - 1}{(N + 1 - A)d} < \frac{1}{2d^2}, \end{aligned} \tag{17}$$

which is

$$2dk[(p + q) - A] - 2d < N + 1 - A. \tag{18}$$

Solving d in (18), we get the upper bound of the private exponent:

$$d < \frac{N + 1 - A}{2k(p + q - A) - 2}. \tag{19}$$

According to the above inequality, we know that the smaller the difference between $p + q$ and A , the higher the upper bound of d . Consequently, in order to extend the security boundary of RSA-Small- d , we attempt to estimate A as precisely as possible such that $p + q - A$ becomes

small. Equation (19) also shows that the complexity of further extending Wiener’s boundary can be reduced to the complexity of estimating the MSBs of $p + q$. The relation is shown in the following.

Rearranging (18) we have

$$2dk(p + q - 1) - 2d < N + (2dk - 1)(A - 1). \tag{20}$$

Denote Λ as the difference between $p + q$ and A . That is, $\Lambda = p + q - A$. Replacing A in (20) by $p + q - \Lambda$ conducts

$$\begin{aligned} 2dk(p + q - 1) - 2d &< N + (2dk - 1)((p + q - \Lambda) - 1) \\ &= 2dk(p + q - 1) \\ &\quad + \varphi(N) - \Lambda(2dk - 1). \end{aligned} \tag{21}$$

In (21), eliminating $2dk(p + q - 1)$ in both sides we get

$$\Lambda(2dk - 1) - 2d < \varphi(N). \tag{22}$$

Now we consider the bit-length of each side. Assume that the bit-length of d is $n/4 + r$ bits, which is longer than Wiener’s boundary by r bits. Due to the key generation of RSA-Small- d , the parameter k is almost the same size as d with a high probability; that is, $\log_2 k \approx \log_2 d$. In addition, we perform an exhaustive search for the first s MSBs of $p + q$. Thus the difference between $p + q$ and A can be reduced to $(n/2 + 1) - s$ bits; that is, $\log_2 \Lambda \approx (n/2 + 1) - s$. Consequently, The term $\Lambda \cdot 2dk$, which dominates the size in the left-hand side of (22), is about $((n/2 + 1) - s) + 1 + 2 \times (n/4 + r)$ bits long and the sufficient condition of (22) is

$$\frac{((n/2 + 1) - s)}{\text{for } \Lambda} + \frac{1 + 2 \times (n/4 + r)}{\text{for } 2dk} < n, \tag{23}$$

which is simplified to

$$2r + 2 < s. \tag{24}$$

Equation (24) gives the following conclusion. In order to extend Wiener’s boundary by r bits, we have to perform an exhaustive search for the first $2r + 2$ MSBs of $p + q$, where $r = \log_2 d - \log_2 N^{1/4}$. This result is better than that of Verheul and van Tilborg’s cost [20], which requires an exhaustive search for $2r + 8$ bits. Therefore, assume that an exhaustive search for 64 bits is feasible in terms of current computational abilities. Solving r for

$$2r + 2 = 64 \tag{25}$$

yields $r = 31$, which means that RSA-Small- d is insecure when $d < N^{1/4} 2^{31}$.

4. Estimated Prime Factor (EPF)

In this section, a novel approach called Estimated Prime Factor (EPF), which is used to estimate the prime factors of an RSA modulus N , is proposed.

4.1. EPF. Without loss of generality, we assume that $q < p < 2q$, where $N = pq$. Denote D_p and D_q as the distances between \sqrt{N} & p and q & \sqrt{N} , respectively. That is,

$$p = \sqrt{N} + D_p, \quad q = \sqrt{N} - D_q. \quad (26)$$

Applying (26) to $N = pq$ yields

$$N = p \cdot q = (\sqrt{N} + D_p) \cdot (\sqrt{N} - D_q) \quad (27)$$

$$= N + \sqrt{N} \cdot (D_p - D_q) - D_p \cdot D_q. \quad (28)$$

Eliminating N in both sides of (27) we have

$$D_p \cdot D_q = \sqrt{N} \cdot (D_p - D_q), \quad (29)$$

which leads to

$$\frac{1}{\sqrt{N}} = \frac{D_p - D_q}{D_p D_q}. \quad (30)$$

Equation (30) is quite interesting because the irrational fraction $1/\sqrt{N}$ reveals partial information of $D_p - D_q$ and $D_p \cdot D_q$. Note that with $D_p - D_q$ and $D_p \cdot D_q$ we can compute $D_p + D_q$ by

$$(D_p + D_q)^2 = (D_p - D_q)^2 + 4D_p D_q \quad (31)$$

and solve D_p and D_q as follows:

$$D_p = \frac{D_p + D_q}{2} + \frac{D_p - D_q}{2}, \quad (32)$$

$$D_q = \frac{D_p + D_q}{2} - \frac{D_p - D_q}{2}.$$

Now we use continued fractions to construct a rational sequence to approximate $1/\sqrt{N}$. Suppose that the i th convergent of the continued fraction expansion of $1/\sqrt{N}$ is h_i/k_i . According to Theorem 2, we know that

$$\frac{h_i}{k_i} \rightarrow \frac{1}{\sqrt{N}}, \quad \text{as } i \rightarrow \infty. \quad (33)$$

Since the sizes of h_i and k_i grow with increase of the index i (see Theorem 2), there exists an index t such that

$$h_t < D_p - D_q < h_{t+1}. \quad (34)$$

We use h_t and k_t as the estimations of $D_p - D_q$ and $D_p D_q$, respectively, instead of using the larger ones. That is,

$$h_t \approx D_p - D_q, \quad k_t \approx D_p D_q. \quad (35)$$

From (31), $D_p + D_q$ is estimated as

$$D_p + D_q \approx \sqrt{h_t^2 + 4k_t}. \quad (36)$$

And thus D_p and D_q are estimated as

$$D_p \approx \frac{\sqrt{h_t^2 + 4k_t} + h_t}{2}, \quad D_q \approx \frac{\sqrt{h_t^2 + 4k_t} - h_t}{2}. \quad (37)$$

Finally, we define the estimated prime factors of N as

$$p_E := \left\lceil \sqrt{N} + \frac{\sqrt{h_t^2 + 4k_t} + h_t}{2} \right\rceil, \quad (38)$$

$$q_E := \left\lfloor \sqrt{N} - \frac{\sqrt{h_t^2 + 4k_t} - h_t}{2} \right\rfloor.$$

4.2. Theoretical Estimation and Experimental Result on Searching the Index t . The process of computing the convergent of the continued fraction expression of $1/\sqrt{N}$ should be ceased at the index t satisfying (34). Thus, we have to estimate the size of $D_p - D_q$ in order to determine the index t . Since $D_p < p$ and $D_q < q$, h_t should not be set larger than $n/2$ bits at least. Next, we investigate the method to estimate the index t theoretically and experimentally.

4.2.1. Theoretical Estimation. From the definitions of D_p and D_q in (26), we have

$$D_p - D_q = p + q - 2\sqrt{N} = (\sqrt{p} - \sqrt{q})^2, \quad (39)$$

which is equivalent to

$$\log_2(D_p - D_q) = 2\log_2(\sqrt{p} - \sqrt{q}). \quad (40)$$

Equation (40) shows that the bit-length of $D_p - D_q$ is twice the bit-length of $\sqrt{p} - \sqrt{q}$. Consider the following problem.

Problem. Randomly select two prime numbers p and q of $n/2$ bits; what is the expected value of the number of MSBs of \sqrt{p} and \sqrt{q} that are identical?

From our theoretical estimation, the expected value is about 2.6, and it is almost independent of the bit-length of N . This implies that, for any two randomly selected prime numbers p and q of $n/2$ bits each, the first 2.6 MSBs of \sqrt{p} and \sqrt{q} are identical on average. Consequently, according to (40), the size of $D_p - D_q$ is expected to be $2 \times (n/4 - 2.6) = n/2 - 5.2$ bits, which increases linearly with the bit-length of N .

4.2.2. Experimental Results. Table 1 shows the experimental results for the index t in EPF. Suppose that p and q are two randomly generated prime numbers of $n/2$ bits each; we then compute $\log_2(D_p - D_q)$, $\log_2(h_t)$, and $\log_2(h_{t+1})$, which denote the bit-lengths of $D_p - D_q$, h_t , and h_{t+1} , respectively. Each block in the table is evaluated from the average value of 1000 experimental instances. As can be observed from the first row, the bit-length of $D_p - D_q$ is approximately equal to $(n/2 - 7)$ bits long for all n and is greater than that of h_t by at least 1 bit on average. This result is slightly different from the result in the previous version at ACNS'07 [23] due to the reason of using different samples in the experiments. Note that in this paper we implement EPF with uniformly distributed samples which are more objective. Moreover, the values of $\log_2(D_p - D_q)$ in Table 1 are slightly smaller than the theoretical estimation $n/2 - 5.2$ bits; the reason may be that

TABLE 1: The improvement of EPF on $p + q$, where p and q are balanced.

n	512	1024	2048
$\log_2(D_p - D_q)$	248.476	504.626	1016.551
t (in average)	146.229	295.772	594.103
$\log_2(h_t)$	247.161	503.04	1015.201
$\log_2(h_{t+1})$	250.12	506.21	1018.14

TABLE 2: The improvement of EPF on $p + q$, where p and q are balanced.

Balanced Modulus $N = pq$	$n = 512$	$n = 1024$	$n = 2048$
$\log_2((p + q) - 2\sqrt{N})$	248.476	504.626	1016.551
$\log_2((p + q) - (p_E + q_E))$	247.185	503.294	1015.248

we ignore the usage of prime-counting function $\pi(\cdot)$ in the calculation. However, the values in Table 1 actually increase linearly with the bit-length of N .

In EPF, we simply estimate the value of $D_p - D_q$, which is, however, smaller than the actual value. On the other hand, up to now, there is no theory to justify the difference between the bit-lengths of h_t and $D_p - D_q$; in fact, this would be an interesting subject of inquiry.

4.3. Accuracy and Further Improvement. We demonstrate the accuracy of EPF in Table 2. Each entry in the table is the data averaged over 1000 samples. The first row shows the difference of the bit-length between $p+q$ and its estimation by using $2\sqrt{N}$. The second row shows the difference of the bit-length between $p + q$ and its estimation by using EPF. As can be seen in Table 2, using $p_E + q_E$ as the estimation is more accurate than using $2\sqrt{N}$ at least one bit on average. This result shows that EPF is better than the traditional estimation method.

To further raise the accuracy rate of EPF, we may employ the properties of continued fractions. According to Theorem 2, we know that

$$h_{t+1} = a_t h_t + h_{t-1}, \quad k_{t+1} = a_t k_t + k_{t-1}, \quad (41)$$

where a_t is the t th component of the continued fraction expression of $1/\sqrt{N}$ (see Definition in Section 2.2). Consequently, for any real number $\lambda \in [1, a_t]$, we have

$$h_t < \lambda h_t + h_{t-1} < h_{t+1}, \quad k_t < \lambda k_t + k_{t-1} < k_{t+1}. \quad (42)$$

Since $D_p - D_q$ and $D_p \cdot D_q$ are also in the intervals (h_t, h_{t+1}) and (k_t, k_{t+1}) , respectively, $\lambda h_t + h_{t-1}$ and $\lambda k_t + k_{t-1}$ might be better estimations of $D_p - D_q$ and $D_p \cdot D_q$. Hence, an interesting question would be how to find a suitable value of λ that yields better estimations of $D_p - D_q$ and $D_p \cdot D_q$. Note that, from the properties of continued fractions, we have

$$\begin{aligned} \frac{h_{t+1}}{k_{t+1}} &> \frac{1}{\sqrt{N}} > \frac{h_t}{k_t} \quad \text{if } t \text{ is odd,} \\ \frac{h_{t+1}}{k_{t+1}} &< \frac{1}{\sqrt{N}} < \frac{h_t}{k_t} \quad \text{if } t \text{ is even.} \end{aligned} \quad (43)$$

Equation (43) implies that there exists an irrational number λ_1 , such that

$$\frac{\lambda_1 h_t + h_{t-1}}{\lambda_1 k_t + k_{t-1}} = \frac{1}{\sqrt{N}}. \quad (44)$$

To find an appropriate number λ , one method could be to choose λ , which is very close to λ_1 , which might yield better estimations of $D_p - D_q$ and $D_p \cdot D_q$. However, we leave this concept as the subject of future work on EPF.

5. Applying EPF to Reduce the Cost of Exhaustive Search to $2r-6$ Bits

In this section, we apply EPF proposed in Section 4 to further reduce the cost of exhaustive search.

From the results of Section 3, the security boundary of RSA-Small- d depends on the known MSBs of $p + q$. In EPF, the experimental results show that the 1st to the 8th MSB of $p + q$, denoted as $\text{MSB}_{1-8}(p + q)$, can be correctly determined with high probability (see Table 2). Consequently, setting $p_E + q_E = 2^{(n/2+1)-8} A_1 + A_2$, where $A_2 < 2^{n/2-7}$, then

$$(A_1)_2 = \text{MSB}_{1-8}(p + q), \quad (45)$$

where $(A_1)_2$ denotes the binary representation of A_1 . Setting $\Lambda = (p + q) - (p_E + q_E)$, (45) also shows that Λ is about $(n/2 + 1) - 8$ bits long. Hence, representing (22) according to the bit-length of the items, Λ , d , k , and $\varphi(N)$ yields

$$\left(\left(\frac{n}{2} + 1 \right) - 8 \right) + 1 + 2 \left(\frac{n}{4} + r \right) < n. \quad (46)$$

Moreover, by performing an exhaustive search for s bits after the 8th MSB of $p + q$, that is, $\text{MSB}_{9-8+s}(p + q)$, we can further reduce the size of Λ to $(n/2 + 1) - (8 + s)$ bits. This implies that the 1st to the $(8 + s)$ th MSB of $p + q$ can be correctly determined and the size of Λ is reduced to $(n/2 + 1) - (8 + s)$ bits. Hence, (46) is revised to

$$\left(\frac{n}{2} + 1 \right) - (8 + s) + 1 + 2 \left(\frac{n}{4} + r \right) < n, \quad (47)$$

which is simplified to

$$2r - 6 < s. \quad (48)$$

Equation (48) is the improved result when applying EPF to the method presented in Section 3. As a conclusion, extending Wiener's boundary by r bits requires only an exhaustive search for $2r - 6$ bits, which results in a lower computational cost than that with Verheul and van Tilborg's extension. We summarize the improvements in each type of attack in Table 3.

With the progress of technology, the ability of machines to perform exhaustive searches will only increase. Figure 1 shows the relations between the security boundaries of the extensions of the Wiener attack and machines with different computational abilities. The symbol s denotes the required number of bits for an exhaustive search to extend Wiener's boundary, and the symbol $|d|$ denotes the upper

TABLE 3: The improvement between each attack.

Attacks	Boundary	Complexity
Wiener Attack	$d < N^{1/4}$	Polynomial time
V-T Extension	$d < N^{1/4}2^r$	Exhaustive search for U and V of $2r + 8$ bits (see (13))
Proposed Improvement (Step 1)	$d < N^{1/4}2^r$	Exhaustive search for $2r + 2$ bits (see (24))
Applying EPF (Step 2)	$d < N^{1/4}2^r$	Exhaustive search for $2r - 6$ bits (see (48))

bound of the insecure private exponent. In terms of the current computational capabilities, an exhaustive search for 64 bits is feasible. Hence, the lines **L1**, **L2** and **L3** yield the improvements of 28 bits, 31 bits, and 35 bits, respectively, over Wiener’s boundary. The boundaries of the extensions of the Wiener attack (see V-T. Ext., Ext. W., and EPF in Figure 1) can be raised to 284 bits, 287 bits, and 291 bits, respectively, when the RSA modulus N is 1024 bits long. Furthermore, if an exhaustive search for 80 bits is feasible, the upper bound of the extension of the Wiener attack through EPF is raised to $N^{1/4}2^{43}$, which is 299 bits when N is 1024 bits long (see **L3**: EPF). This result is comparable to the boundary of the lattice attack proposed by Boneh and Durfee [19], which has a best upper bound, but heuristic, at the present. Note that there is no guaranty that a heuristic algorithm can output the solution. One may concern whether the assumption that an exhaustive search for 80 bits is feasible or not. In the opinion of current development, it will not be a difficult task to achieve such computational capability in the near future. According to Moore’s Law, computers will double in speed approximately every 18 months, which further supports our assumption. Moreover, paralleling techniques and special-purpose machines can help in speeding-up the computation.

6. Conclusion and Future Works

With the rapid growth of different network environments such as wireless sensor networks [24–27], security is normally the most concerned issue. In this paper, we propose a method, called EPF, to estimate the prime factors of an RSA modulus. With EPF, the cost of exhaustive search can further reduce to $2r - 6$ bits. It means that the cost is 2^{14} times faster than Verheul and van Tilborg’s result and the security boundary is extended 7 bits. It should be noted that their method for an exhaustive search is heuristic since this method is based on the results of distribution of small partial quotient in the continued fraction expansions.

An interesting problem in EPF is whether there exists a deterministic algorithm for finding an index t satisfying $h_t < D_p - D_q < h_{t+1}$. In this paper, we use the theoretical estimation to determine the index t . The success rate is 85.1% according to our experiments. Now, another question arises—how to increase the success rate of the process of finding the index t when the deterministic algorithm is not developed. In addition, the other researchable question is how to improve the accuracy rate of MSBs of $p_E + q_E$, which brings a greater contributive effort of EPF.

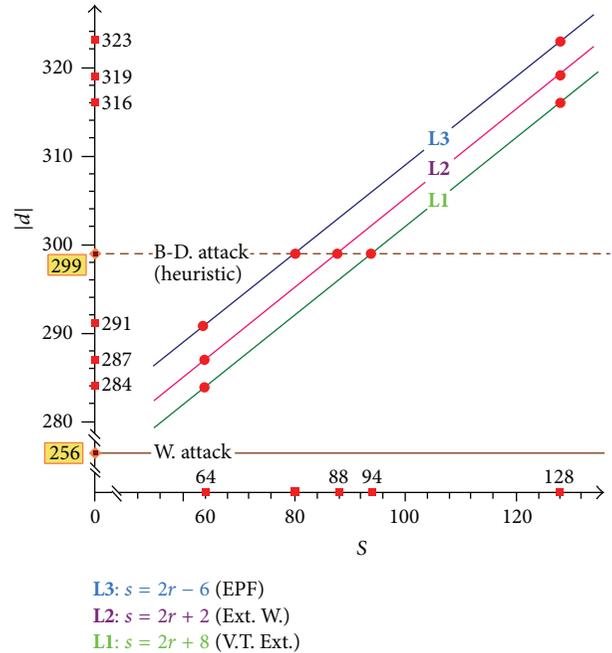


FIGURE 1: The boundaries of the extensions of the Wiener attack under different computational capabilities, where 256 and 299 are the boundaries of the Wiener attack (W. Attack) and Boneh and Durfee’s attack (B-D. Attack), respectively. **L1**, **L2**, and **L3** denote the boundaries of Verheul and Tilborg’s extension (V-T. Ext.) (see [20]), the extension of the Wiener attack (Step 1) (Ext. W.) (see (24)), and the extension of the Wiener attack through EPF (EPF) (see (48)).

We should point out that EPF can be applied to Dujella’s refinement [14] and the generalized Wiener attack [18]. Moreover, we foresee that EPF could be applied to other cryptogrammic aspects, especially to the attacks for cryptosystems based on the integer factorization problem (IFP). For example, the lattice technique is commonly used for the cryptanalysis of RSA [17, 28–30] or for the attacks on RSA with small exponents [15, 18, 19, 21, 22, 31, 32]. We expect EPF to be a supportive tool for assisting the lattice technique to increase the effort on the cryptanalysis of RSA. As a conclusion, we would like to point out that with the continuous improvements in computational capability, the security levels are expected to be higher with the assistance of EPF, and the security analysis should be considered more carefully.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions, which certainly led to improvements of this paper. Chien-Ming Chen was partially supported by the Shenzhen Peacock Project, China, under Contract no. KQC201109020055A and the Shenzhen Strategic Emerging Industries Program under Grant no. ZDSY20120613125016389. Hung-Min Sun was partially supported by the National Science Council, Taiwan, under Grant NSC 100-2628-E-007-018-MY3.

References

- [1] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [2] C. Patsakis, "Number theoretic SETUPS for RSA like factoring based algorithms," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 3, no. 2, pp. 191–204, 2012.
- [3] Q. Kong, P. Li, and Y. Ma, "On the feasibility and security of image secret sharing scheme to identify cheaters," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 4, pp. 2073–4212, 2013.
- [4] N. Peng, G. Luo, K. Qin, and A. Chen, "Query-biased preview over outsourced and encrypted data," *The Scientific World Journal*, vol. 2013, Article ID 860621, 13 pages, 2013.
- [5] H. Lenstra Jr., "Factoring integers with elliptic curves," *Annals of Mathematics*, vol. 126, no. 3, pp. 649–673, 1987.
- [6] J. Pollard, "Theorems on factorization and primality testing," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 76, no. 3, pp. 521–528, 1974.
- [7] D. Boneh and H. Shacham, "Fast variants of RSA," *CryptoBytes*, vol. 5, no. 1, pp. 1–9, 2002.
- [8] S. Galbraith, C. Heneghan, and J. McKee, "Tunable balancing of RSA," in *Information Security and Privacy*, vol. 3574 of *Lecture Notes in Computer Science*, pp. 280–292, Springer, Berlin, Germany, 2005.
- [9] M. Hinek, "Another look at small RSA exponents," in *Topics in Cryptology-CT-RSA 2006*, vol. 3860 of *Lecture Notes in Computer Science*, pp. 82–98, Springer, Berlin, Germany, 2006.
- [10] H. Sun, W. Yang, and C. Lai, "On the design of RSA with short secret exponent," in *Advances in Cryptology-ASIACRYPT '99*, vol. 1716 of *Lecture Notes in Computer Science*, pp. 150–164, Springer, Berlin, Germany, 1999.
- [11] H. Sun and C. Yang, "RSA with balanced short exponents and its application to entity authentication," in *Public Key Cryptography-PKC 2005*, vol. 3386 of *Lecture Notes in Computer Science*, pp. 199–215, Springer, Berlin, Germany, 2005.
- [12] S. Vanstone and R. Zuccherato, "Short RSA keys and their generation," *Journal of Cryptology*, vol. 8, no. 2, pp. 101–114, 1995.
- [13] D. Boneh, R. Rivest, A. Shamir et al., "Twenty years of attacks on the RSA cryptosystem," *Notices of the American Mathematical Society*, vol. 46, no. 2, pp. 203–213, 1999.
- [14] A. Dujella, "Continued fractions and RSA with small secret exponent," *Tatra Mountains Mathematical Publications*, vol. 29, pp. 101–112, 2004.
- [15] E. Jochemsz and B. de Weger, "A partial key exposure attack on RSA using a 2-dimensional lattice," in *Information Security*, vol. 4176 of *Lecture Notes in Computer Science*, pp. 203–216, Springer, Berlin, Germany, 2006.
- [16] M. J. Wiener, "Cryptanalysis of short RSA secret exponents," *IEEE Transactions on Information Theory*, vol. 36, no. 3, pp. 553–558, 1990.
- [17] B. de Weger, "Cryptanalysis of RSA with small prime difference," *Applicable Algebra in Engineering, Communications and Computing*, vol. 13, no. 1, pp. 17–28, 2002.
- [18] J. Blömer and A. May, "A generalized Wiener attack on RSA," in *Public Key Cryptography-PKC 2004*, vol. 2947 of *Lecture Notes in Computer Science*, pp. 1–13, Springer, Berlin, Germany, 2004.
- [19] D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$," in *Advances in Cryptology-EUROCRYPT '99*, vol. 1592 of *Lecture Notes in Computer Science*, pp. 1–11, Springer, Berlin, Germany, 1999.
- [20] E. Verheul and H. van Tilborg, "Cryptanalysis of 'less short' RSA secret exponents," *Applicable Algebra in Engineering, Communications and Computing*, vol. 8, no. 5, pp. 425–435, 1997.
- [21] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, "Low-exponent RSA with related messages," in *Advances in Cryptology-EUROCRYPT '96*, vol. 1070 of *Lecture Notes in Computer Science*, pp. 1–9, Springer, Berlin, Germany, 1996.
- [22] D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1339–1349, 2000.
- [23] H. M. Sun, M. E. Wu, and Y. H. Chen, "Estimating the prime-factors of an rsa modulus and an extension of the wiener attack," in *Applied Cryptography and Network Security*, vol. 4521 of *Lecture Notes in Computer Science*, pp. 116–128, Springer, Berlin, Germany, 2007.
- [24] C. M. Chen, Y. H. Lin, Y. H. Chen, and H. M. Sun, "SASHIMI: secure aggregation via successively hierarchical inspecting of message integrity on WSN," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 1, pp. 57–72, 2013.
- [25] C. M. Chen, Y. H. Lin, Y. C. Lin, and H. M. Sun, "RCDA: recoverable concealed data aggregation for data integrity in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, pp. 727–734, 2012.
- [26] H.-M. Sun, C.-M. Chen, and Y.-C. Hsiao, "An efficient countermeasure to the selective forwarding attack in wireless sensor networks," in *Proceedings of the IEEE Region 10 Conference (TENCON '07)*, pp. 1–4, IEEE, Taipei, Taiwan, November 2007.
- [27] K. Wei-Chi, C. Chien-Ming, and L. Hui-Lung, "Cryptanalysis of a variant of Peyravian-Zunic's password authentication scheme," *IEICE Transactions on Communications*, vol. 86, no. 5, pp. 1682–1684, 2003.
- [28] D. Coppersmith, "Finding a small root of a bivariate integer equation; factoring with high bits known," in *Advances in Cryptology-EUROCRYPT '96*, vol. 1070 of *Lecture Notes in Computer Science*, pp. 178–189, Springer, Berlin, Germany, 1996.
- [29] D. Coppersmith, "Finding a small root of a univariate modular equation," in *Advances in Cryptology-EUROCRYPT '96*, vol. 1070 of *Lecture Notes in Computer Science*, pp. 155–165, Springer, Berlin, Germany, 1996.
- [30] H. Sun, M. Wu, W. Ting, and M. Hinek, "Dual RSA and its security analysis," *IEEE Transactions on Information Theory*, vol. 53, no. 8, pp. 2922–2933, 2007.

- [31] D. Bleichenbacher and A. May, "New attacks on RSA with small secret CRT-exponents," in *Public Key Cryptography-PKC 2006*, vol. 3958 of *Lecture Notes in Computer Science*, pp. 1–13, Springer, Berlin, Germany, 2006.
- [32] D. Boneh, G. Durfee, and Y. Frankel, "An attack on RSA given a small fraction of the private key bits," in *Advances in Cryptology-ASIACRYPT '98*, vol. 1514 of *Lecture Notes in Computer Science*, pp. 25–34, Springer, Berlin, Germany, 1998.

Research Article

Novel Image Encryption Scheme Based on Chebyshev Polynomial and Duffing Map

Borislav Stoyanov and Krasimir Kordov

Faculty of Mathematics and Informatics, Konstantin Preslavski University of Shumen, 9712 Shumen, Bulgaria

Correspondence should be addressed to Borislav Stoyanov; borislav.stoyanov@shu-bg.net

Received 10 December 2013; Accepted 4 March 2014; Published 26 March 2014

Academic Editors: T. Cao, M. K. Khan, and F. Yu

Copyright © 2014 B. Stoyanov and K. Kordov. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a novel image encryption algorithm using Chebyshev polynomial based on permutation and substitution and Duffing map based on substitution. Comprehensive security analysis has been performed on the designed scheme using key space analysis, visual testing, histogram analysis, information entropy calculation, correlation coefficient analysis, differential analysis, key sensitivity test, and speed test. The study demonstrates that the proposed image encryption algorithm shows advantages of more than 10^{113} key space and desirable level of security based on the good statistical results and theoretical arguments.

1. Introduction

In recent years, the dynamical chaotic systems have been commonly used for the design of cryptographic primitives featuring chaotic behaviour and random-like properties. In his seminal work [1], Shannon pointed out the excellent possibilities of the dynamical chaotic maps in the communications. He identified two basic properties that the good data encryption systems should have to prevent (resist) statistical attacks: diffusion and confusion. Diffusion can propagate the change over the whole encrypted data, and confusion can hide the relationship between the original data and the encrypted data. Permutation, which rearranges objects, is the simplest method of diffusion, and substitution, that replaces an object with another one, is the simplest type of confusion. The consistent use of dynamical chaotic system based permutation and substitution methods is in the deep cryptographic fundamental.

The authors of [2] used Chebyshev polynomial to construct secure El Gamal-like and RSA-like algorithms. A new more practical and secure Diffie-Hellman key agreement protocol based on Chebyshev polynomial is presented in [3]. In [4], a stream cipher constructed by Duffing map based message-embedded scheme is proposed. By mixing the Lorenz attractor and Duffing map, a new six-dimensional

chaotic cryptographic algorithm with good complex structure is designed [5]. In [6], an improved stochastic middle multibits quantification algorithm based on Chebyshev polynomial is proposed. Three-party key agreement protocols using the enhanced Chebyshev polynomial are proposed in [7, 8].

Fridrich [9] describes how to adapt Baker map, Cat map, and Standard map on a torus or on a rectangle for the purpose of substitution-permutation image encryption. In [10], a new permutation-substitution image encryption scheme using logistic, tent maps, and Tompkins-Paige algorithm is proposed. In [11], chaotic cipher is proposed to encrypt color images through position permutation part and Logistic map based on substitution. Yau et al. [12] proposed an image encryption scheme based on Sprott chaotic circuit. In [13], Fu et al. proposed a digital image encryption method by using Chirikov standard map based permutation and Chebyshev polynomial based diffusion operations.

In [14], a bit-level permutation scheme using chaotic sequence sorting has been proposed for image encryption. The operations are completed by Chebyshev polynomial and Arnold Cat map. An image encryption algorithm in which the key stream is generated by Chebyshev function is presented in [15]. Simulation results are given to confirm the necessary level of security. In [16], a new image encryption

scheme, based on Chebyshev polynomial, Sin map, Cubic map, and 2D coupled map lattice, is proposed. The experimental results show the security of the algorithm.

In [17], a color image encryption scheme based on skew tent map and hyper chaotic system of 6th-order CNN is presented. An image encryption scheme based on rotation matrix bit-level permutation and block diffusion is proposed in [18].

A new chaos based image encryption scheme is suggested in this paper. The algorithm is a simple improvement of one round substitution-permutation model. The encryption process is divided in two major parts: Chebyshev polynomial based on substitution and substitution and Duffing map based on substitution. In Section 2, we propose two pseudorandom bit generators (PRBGs): one based on Chebyshev polynomial and the other based on Duffing map. In Section 3, in order to measure randomness of the bit sequence generated by the two pseudorandom schemes, we use NIST, DIEHARD, and ENT statistical packages. Section 4 presents the proposed image encryption algorithm, and some security cryptanalysis is given. Finally, the last section concludes the paper.

2. Proposed Pseudorandom Bit Generators

2.1. Pseudorandom Bit Generator Based on the Chebyshev Polynomial. In this section, the real numbers of two Chebyshev polynomials [2, 19] are preprocessed and combined with a simple threshold function to a binary pseudorandom sequence.

The proposed pseudorandom bit generator is based on two Chebyshev polynomials, as described by

$$\begin{aligned} x(n+1) &= T_k(x_n) = \cos(k \times \arccos(x_n)), \\ y(m+1) &= T_l(y_m) = \cos(l \times \arccos(y_m)), \end{aligned} \quad (1)$$

where $(x_n, y_m) \in [-1, 1]$ and $(k, l) \in [2, \infty)$ are control parameters. The initial values $x(0)$ and $y(0)$ and parameters (k, l) are used as the key.

Step 1. The initial values $x(0)$, $y(0)$, k , and l of the two Chebyshev polynomials from (1) are determined.

Step 2. The first and the second Chebyshev polynomials from (1) are iterated for K_0 and L_0 times to avoid the harmful effects of transitional procedures, respectively, where K_0 and L_0 are different constants.

Step 3. The iteration of (1) continues, and, as a result, two decimal fractions $x(n)$ and $y(m)$ are generated.

Step 4. These decimal fractions are preprocessed as follows:

$$\begin{aligned} x(n) &= \text{mod}(\text{floor}(\text{abs}(x(n) \times 10^{14})), 2) \\ y(m) &= \text{mod}(\text{floor}(\text{abs}(y(m) \times 10^{14})), 2), \end{aligned} \quad (2)$$

where $\text{abs}(x)$ returns the absolute value of x , $\text{floor}(x)$ returns the value of x to the nearest integers less than or equal to x , and $\text{mod}(x, y)$ returns the remainder after division.

Step 5. The following threshold function g from (3) is applied:

$$g(x(n), y(m)) = \begin{cases} 1, & \text{if } x(n) > y(m), \\ 0, & \text{if } x(n) \leq y(m), \end{cases} \quad (3)$$

and a pseudorandom bit is produced.

Step 6. Return to Step 3 until pseudorandom bit stream limit is reached.

2.2. Pseudorandom Bit Generator Based on the Duffing Map. In this section, the real numbers of two Duffing maps are preprocessed and combined with a simple threshold function to a binary pseudorandom sequence.

The Duffing map is a 2D discrete dynamical system which takes a point (u_n, v_n) in the plane and maps it to a new point. The proposed pseudorandom bit generator is based on two Duffing maps, given by the following equations:

$$\begin{aligned} u_{1,n+1} &= v_{1,n}, \\ v_{1,n+1} &= -bu_{1,n} + av_{1,n} - v_{1,n}^3, \\ u_{2,m+1} &= v_{2,m}, \\ v_{2,m+1} &= -bu_{2,m} + av_{2,m} - v_{2,m}^3. \end{aligned} \quad (4)$$

The maps depend on the two constants a and b . These are usually set to $a = 2.75$ and $b = 0.2$ to produce chaotic nature. The initial values $u_{1,0}$, $v_{1,0}$, $u_{2,0}$, and $v_{2,0}$ are used as the key.

Step 1. The initial values $u_{1,0}$, $v_{1,0}$, $u_{2,0}$, and $v_{2,0}$ of the two Duffing maps from (4) are determined.

Step 2. The first and the second Duffing maps from (4) are iterated for M_0 and N_0 times, respectively, to avoid the harmful effects of transitional procedures, where M_0 and N_0 are different constants.

Step 3. The iteration of (4) continues, and, as a result, two real fractions $x(n)$ and $y(m)$ are generated.

Step 4. The following threshold function h from (5) is applied:

$$h(v_{1,n}, v_{2,m}) = \begin{cases} 1, & \text{if } v_{1,n} > v_{2,m}, \\ 0, & \text{if } v_{1,n} \leq v_{2,m}, \end{cases} \quad (5)$$

and a pseudorandom bit is produced.

Step 5. Return to Step 3 until pseudorandom bit stream limit is reached.

3. Statistical Test Analysis of the Proposed Pseudorandom Bit Generators

In order to measure randomness of the zero-one sequence generated by the new pseudorandom generators, we used NIST, DIEHARD, and ENT statistical packages.

The Chebyshev polynomial and the Duffing map based pseudorandom bit schemes are implemented by software simulation in C++ language, using the following initial seeds: $x(0) = 0.9798292345345$, $y(0) = -0.4032920230495034$, $k = 2.995$, $l = 3.07$, $u_{1,0} = -0.04$, $v_{1,0} = 0.2$, $u_{2,0} = 0.23$, and $v_{2,0} = -0.13$, stated as a key KI.

3.1. NIST Statistical Test Analysis. The NIST statistical test suite (version 2.1.1) is proposed by the National Institute of Standards and Technology [20]. The suite includes 15 tests, which focus on a variety of different types of non-randomness that could exist in a sequence. These tests are frequency (monobit), block-frequency, cumulative sums, runs, longest run of ones, rank, fast Fourier transform (spectral), nonoverlapping templates, overlapping templates, Maurer’s “universal statistical,” approximate entropy, random excursions, random-excursion variant, and serial and linear complexity. The testing process consists of the following steps:

- (1) state the null hypothesis; assume that the binary sequence is random;
- (2) compute a sequence test statistic; testing is carried out at the bit level;
- (3) compute the P -value; P -value $\in [0, 1]$;
- (4) fix α , where $\alpha \in [0.0001, 0.01]$; compare the P -value to α ; *Success* is declared whenever P -value $\geq \alpha$; otherwise, *failure* is declared.

Given the empirical results from a particular statistical test, the NIST suite computes the proportion of sequences that pass. The range of acceptable proportion is determined using the confidence interval defined as

$$\hat{p} \pm 3 \sqrt{\frac{\hat{p}(1 - \hat{p})}{m}}, \tag{6}$$

where $\hat{p} = 1 - \alpha$ and m is the number of binary tested sequences. In our setup, $m = 1000$. Thus the confidence interval is

$$0.99 \pm 3 \sqrt{\frac{0.99(0.01)}{1000}} = 0.99 \pm 0.0094392. \tag{7}$$

The proportion should lie above 0.9805607.

The distribution of P -values is examined to ensure uniformity. The interval between 0 and 1 is divided into 10 subintervals. The P -values that lie within each subinterval are counted. Uniformity may also be specified through an application of χ^2 test and the determination of a P -values corresponding to the goodness-of-fit distributional test on the P -values obtained for an arbitrary statistical test, P -values of the P -values. This is implemented by calculating

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - s/10)^2}{s/10}, \tag{8}$$

where F_i is the number of P -values in subinterval i and s is the sample size. A P -values is calculated such that

$P\text{-value}_T = IGAMC(9/2, \chi^2/2)$, where $IGAMC$ is the complemented incomplete gamma statistical function. If $P\text{-value}_T \geq 0.0001$, then the sequences can be deemed to be uniformly distributed.

Using the proposed pseudorandom Using the proposed pseudorandom bit generators were generated 1000 sequences of 1000000 bits. The results from all statistical tests are given in Table 1.

The entire NIST test is passed successfully: all the P -values from all 2×1000 sequences are distributed uniformly in the 10 subintervals and the pass rate is also in acceptable range. The minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately 980 for a sample size of 1000 binary sequences for both of pseudorandom generators. The minimum pass rate for the random excursion (variant) test is approximately 589 for a sample size of 603 binary sequences for Chebyshev polynomial based PRBG and 604 for a sample size of 618 binary sequences for Duffing map based PRBG. This shows that the generated pseudorandom sequences feature reliable randomness.

Overall, the results confirm that the novel chaotic cryptographic schemes based on Chebyshev polynomial and Duffing map accomplish a very high level of randomness of the bit sequences.

3.2. DIEHARD Statistical Test Analysis. The DIEHARD suite [21] consists of a number of different statistical tests: birthday spacings, overlapping 5-permutations, binary rank (31×31), binary rank (32×32), binary rank (6×8), bit stream, Overlapping Pairs Sparse Occupancy, Overlapping Quadruples Sparse Occupancy, DNA, stream count-the-ones, byte count-the-ones, 3D spheres, squeeze, overlapping sums, runs up, runs down, and craps. For the DIEHARD tests, we generated two files with 80 million bits each, from the proposed chaotic pseudorandom bit generators. The results are given in Table 2. All P -values are in acceptable range of $[0, 1)$. The proposed pseudorandom bit generators passed all the tests of DIEHARD software.

3.3. ENT Statistical Test Analysis. The ENT package [22] performs 6 tests (entropy, optimum compression, χ^2 distribution, arithmetic mean value, Monte Carlo π estimation, and serial correlation coefficient) to sequences of bytes stored in files and outputs the results of those tests. We tested output of the two strings of 125000000 bytes of the proposed Chebyshev polynomial based pseudorandom bit generator and Duffing map based pseudorandom bit generator, respectively. The results are summarized in Table 3. The proposed pseudorandom bit generators passed all the tests of ENT.

4. Image Encryption Based on Chebyshev polynomial and Duffing Map

Here, we describe an image encryption scheme based on the proposed Chebyshev polynomial and Duffing map based pseudorandom bit generators. We also provide security analysis of the encrypted images.

TABLE 1: NIST statistical test suite results for 2×1000 sequences of size 10^6 -bit each generated by the proposed Chebyshev polynomial based pseudorandom bit generator and Duffing map based pseudorandom bit generator.

NIST statistical test	Chebyshev PRBG		Duffing PRBG	
	P-value	Pass rate	P-value	Pass rate
Frequency (monobit)	0.649612	990/1000	0.490483	989/1000
Block-frequency	0.455937	991/1000	0.777265	992/1000
Cumulative sums (forward)	0.877083	990/1000	0.660012	988/1000
Cumulative sums (reverse)	0.983938	992/1000	0.284024	987/1000
Runs	0.062427	995/1000	0.490983	993/1000
Longest run of ones	0.215574	993/1000	0.612147	992/1000
Rank	0.848027	991/1000	0.212184	988/1000
FFT	0.194813	993/1000	0.013474	993/1000
Nonoverlapping templates	0.504571	990/1000	0.458442	990/1000
Overlapping templates	0.219006	992/1000	0.279844	988/1000
Universal	0.660012	986/1000	0.278461	991/1000
Approximate entropy	0.000478	990/1000	0.363593	991/1000
Random excursions	0.508738	597/603	0.671829	612/618
Random excursions variant	0.614825	596/603	0.490932	612/618
Serial 1	0.585209	991/1000	0.779188	990/1000
Serial 2	0.767582	989/1000	0.713641	993/1000
Linear complexity	0.711601	986/1000	0.699313	991/1000

TABLE 2: DIEHARD statistical test results for two 80 million bits sequences generated by the proposed Chebyshev polynomial based pseudorandom bit generator and Duffing map based pseudorandom bit generator.

DIEHARD statistical test	Chebyshev PRBG	Duffing PRBG
	P-value	P-value
Birthday spacings	0.377207	0.640772
Overlapping 5-permutation	0.410588	0.051538
Binary rank (31×31)	0.551701	0.900609
Binary rank (32×32)	0.940609	0.604265
Binary rank (6×8)	0.530332	0.504383
Bit stream	0.428729	0.461876
OPSO	0.493583	0.498226
OQSO	0.582980	0.478843
DNA	0.632916	0.505181
Stream count-the-ones	0.759561	0.853126
Byte count-the-ones	0.605761	0.479987
Parking lot	0.425621	0.412316
Minimum distance	0.522822	0.486276
3D spheres	0.468043	0.414503
Squeeze	0.236035	0.416625
Overlapping sums	0.543661	0.439732
Runs up	0.234988	0.775408
Runs down	0.527703	0.679825
Craps	0.128550	0.423157

4.1. *Encryption Scheme.* The proposed image encryption algorithm is modification of the classical substitution-permutation scheme [9], column by column shuffling and masking procedures [23], and the diffusion-substitution

model [24]. Here, every single pixel relocation is based on random permutation at once with substitution. The novel derivative scheme has the features of a two-round permutation-substitution color image encryption algorithm. The image encryption method is based on the unique combination of the output bits of the new proposed pseudorandom bit generators.

Without loss of generality, we assume that the dimension of the plain images is $m \times n$ size, where m is the number of rows and n is the number of the columns. The binary lengths of m and n are n_0 and m_0 , respectively. The encryption process is divided into two stages. In the first stage, we generate buffer image B of $m \times n$ size by rearranging and modifying the pixel values of the plain image P by Chebyshev polynomial based PRBG. In the second stage, we generate ciphered image C of $m \times n$ size by modifying the buffer pixel values by Duffing map based PRBG. The encryption process starts with empty buffer image. The plain image pixels are passed sequentially left to right and top to bottom. The entire encryption process is given below.

Step 1. The Chebyshev polynomial based PRBG is iterated continuously to produce m_0 and n_0 bits pseudorandom numbers i' and j' which are transformed modulo m and n , respectively.

Step 2. Repeat Step 1 until an empty pixel with (i', j') coordinates in the buffer image is detected.

Step 3. Continue to do iteration of Chebyshev polynomial based PRBG until 24 bits are produced.

Step 4. To produce buffered image pixel $b(i', j')$, do XOR between a plain image pixel $p(i, j)$ and the last generated 24 bits.

TABLE 3: ENT statistical test results for two 80 million bits sequences generated by the proposed Chebyshev polynomial based pseudorandom bit generator and Duffing map based pseudorandom bit generator, respectively.

ENT statistical test	Chebyshev PRBG results	Duffing PRBG results
Entropy	7.999999 bits per byte	7.999999 bits per byte
Optimum compression	OC would reduce the size of this 125000000 byte file by 0%	OC would reduce the size of this 125000000 byte file by 0%
χ^2 distribution	For 125000000 samples it is 222.98 and randomly would exceed this value 92.68% of the time	For 125000000 samples it is 228.17 and randomly would exceed this value 88.54% of the time
Arithmetic mean value	127.49810 (127.5 = random)	127.5050 (127.5 = random)
Monte Carlo π estimation	3.142062386 (error 0.01%)	3.140968178 (error 0.02%)
Serial correlation coefficient	-0.000026 (totally uncorrelated = 0.0)	0.000018 (totally uncorrelated = 0.0)

Step 5. Repeat Steps 1-4 until all of the plain image pixels are processed.

Step 6. Iterate the Duffing map based PRBG to produce $m \times n \times 24$ bits. Then, do XOR operation between the pseudorandom bit sequence and all of the buffer pixels in the buffered image to produce the encryption image C.

For the reasons of security, we propose several overall rounds of the encryption procedure.

4.2. Security Analysis. The novel image encryption algorithm is implemented in C++ language. All experimental results discussed in the next subsections have been taken by using one iteration of the scheme.

Sixteen 24-bit color images have been encrypted for the security tests. The images are selected from the USC-SIPI image database, miscellaneous volume, available and maintained by the University of Southern California Signal and Image Processing Institute (<http://sipi.usc.edu/database/>). The image numbers are from 4.1.01 to 4.1.08, size 256×256 pixels, from 4.2.01 to 4.2.07, size 512×512 pixels, and House, size 512×512 pixels. The chosen images are currently stored in TIFF format and we have converted them into BMP format (24 bits/pixel).

4.2.1. Key Space Analysis. The key space is the set of all possible keys that can be used in encryption/decryption algorithm. The key of the proposed image encryption scheme is that it is produced by the combination of Chebyshev polynomial based PRBG and Duffing map based PRBG. The novel scheme has eight secret keys $x(0)$, $y(0)$, k , l , $u_{1,0}$, $v_{1,0}$, $u_{2,0}$, and $v_{2,0}$. According to the IEEE floating-point standard [25], the computational precision of the 64-bit double-precision number is about 10^{-15} . If we assume the precision of 10^{-14} , the secret key's space is more than $10^{113} \approx 2^{375}$. This is large enough to defeat brute-force attacks [26] and it is larger than key space size of the image encryption algorithms proposed in [10, 27-29].

Moreover, the initial iteration numbers K_0 , L_0 , M_0 , and N_0 can also be used as a part of the secret key.

4.2.2. Visual Testing. The new algorithm is tested using simple visual inspection of the plain images and corresponding encrypted images. The visual observation does not find convergences between every plain image and its encrypted versions. As an example, Figure 1 shows the plain image 4.2.05 Airplane (F-16), Figure 1(a), and its encrypted version, Figure 1(b). The encrypted image does not contain any constant regions in representative color or texture. The proposed chaos based image encryption breaks any visual data from the plain images.

4.2.3. Histogram Analysis. An image histogram of pixels is a type of a bar graph. It illustrates the visual impact of a distribution of colors that are at certain intensity. We have calculated histograms of red, green, and blue channels of both plain images and their encrypted version by the new image encryption scheme. One representative example among them is shown in Figure 2. The histograms of encrypted image are completely uniformly distributed and considerably different from that of the plain image.

In addition, the average pixel intensity calculations in Table 4, for all of the encrypted images, confirmed the uniformity in distribution of red, green, and blue channels.

4.2.4. Information Entropy. The entropy $H(X)$ is statistical measure of uncertainty in information theory [1]. It is defined as follows:

$$H(X) = -\sum_{i=0}^{255} p(x_i) \log_2 p(x_i), \tag{9}$$

where X is a random variable and $p(x_i)$ is the probability mass function of the occurrence of the symbol x_i . Let us consider that there are 256 states of the information source in red, green, and blue colors of the image with the same probability. We can get the ideal $H(X) = 8$, corresponding to a truly random source.

The information entropy of red, green, and blue channels of the plain images and their corresponding encrypted images are computed and displayed in Table 5. From the obtained values, it is clear that the entropies of red, green, and blue

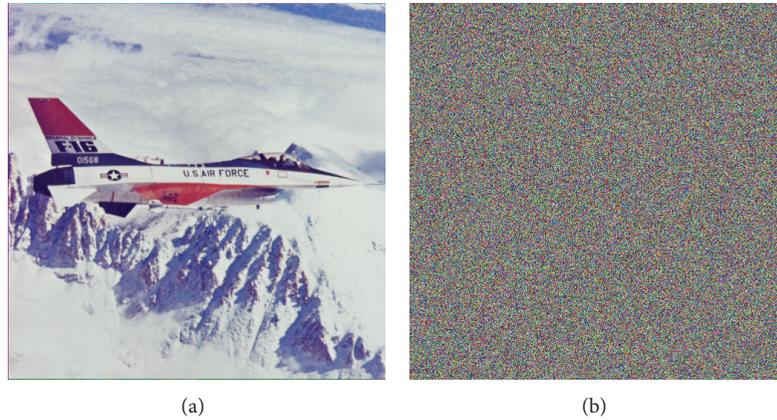


FIGURE 1: Comparison of the plain image and the encrypted image: (a) original picture 4.2.05 Airplane (F-16); (b) encrypted image of 4.2.05 Airplane (F-16).

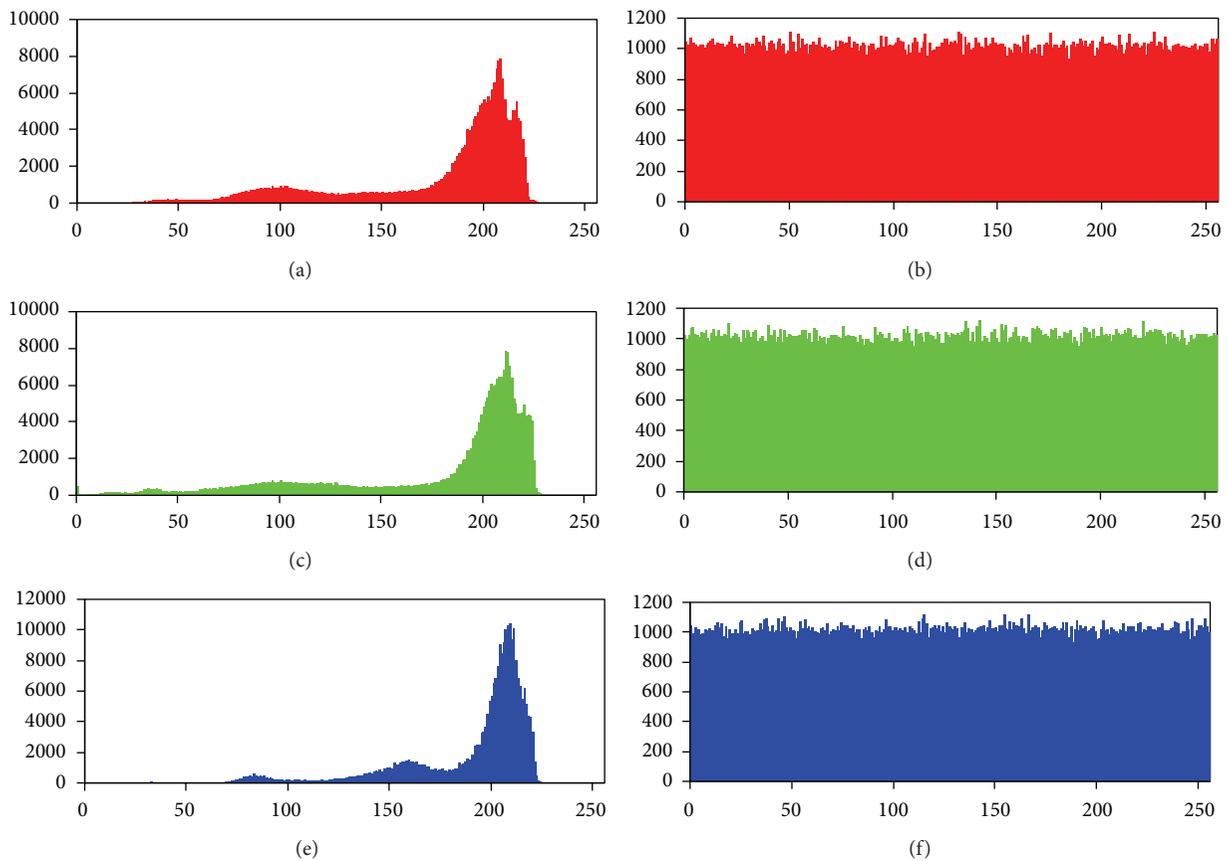


FIGURE 2: Histogram analysis of plain image and encrypted image: (a), (c), and (e) show the histograms of red, green, and blue channels of plain picture 4.2.05 Airplane (F-16); (b), (d), and (f) show the histograms of red, green, and blue channels of encrypted picture 4.2.05 Airplane (F-16).

colors of the encrypted images are very close to the best possible theoretical value, which is an indication that the new chaos based image encryption scheme is trustworthy and secure upon information entropy attack.

4.2.5. Correlation Coefficient Analysis. The adjacent pixels in plain images are strongly correlated in either horizontal,

vertical, or diagonal direction. The correlation coefficient r between two adjacent pixels (a_i, b_i) is computed in accordance with the way described in [30]. Consider

$$r = \frac{\text{cov}(a, b)}{\sqrt{D(a)} \sqrt{D(b)}}, \quad (10)$$

TABLE 4: Average pixel intensity of plain image colors and encrypted image colors.

File name	Plain image color			Encrypted image color		
	Red	Green	Blue	Red	Green	Blue
4.1.01	75.827	52.559	46.305	127.661	127.203	127.458
4.1.02	42.075	30.086	27.540	128.033	127.233	127.413
4.1.03	137.603	139.958	144.018	127.265	127.394	127.697
4.1.04	129.218	99.267	125.199	127.393	127.275	127.021
4.1.05	146.564	133.000	142.023	127.779	127.268	126.878
4.1.06	132.202	124.902	143.263	127.563	127.377	127.346
4.1.07	179.204	180.650	142.348	127.152	127.325	127.378
4.1.08	174.897	170.866	128.346	126.856	127.426	127.563
4.2.01	176.270	70.494	108.898	127.562	127.118	127.468
4.2.02	234.195	208.644	163.552	127.101	127.729	127.722
4.2.03	137.391	128.859	113.117	127.199	128.126	127.392
4.2.04	180.224	99.051	105.410	127.266	127.311	127.519
4.2.05	177.577	177.852	190.214	127.331	127.565	127.654
4.2.06	131.007	124.304	114.893	127.254	127.850	127.450
4.2.07	149.821	115.568	66.534	127.368	127.379	127.257
House	155.436	168.226	142.209	127.419	127.659	127.525

TABLE 5: Entropy results of plain images and encrypted images.

File name	Plain image color			Encrypted image color		
	Red	Green	Blue	Red	Green	Blue
4.1.01	6.42005	6.44568	6.38071	7.96418	7.96805	7.96648
4.1.02	6.24989	5.96415	5.93092	7.96622	7.96734	7.96629
4.1.03	5.65663	5.37385	5.71166	7.96606	7.96740	7.96398
4.1.04	7.25487	7.27038	6.78250	7.96692	7.96552	7.96741
4.1.05	6.43105	6.53893	6.23204	7.96587	7.96618	7.96776
4.1.06	7.21044	7.41361	6.92074	7.96598	7.96697	7.96786
4.1.07	5.26262	5.69473	6.54641	7.96392	7.96634	7.96632
4.1.08	5.79199	6.21951	6.79864	7.96515	7.96651	7.96782
4.2.01	6.94806	6.88446	6.12645	7.96799	7.96762	7.96848
4.2.02	4.33719	6.66433	6.42881	7.96825	7.96838	7.96582
4.2.03	7.70667	7.47443	7.75222	7.96999	7.96778	7.96859
4.2.04	7.25310	7.59404	6.96843	7.96777	7.96932	7.96998
4.2.05	6.71777	6.79898	6.21377	7.96715	7.96807	7.96883
4.2.06	7.31239	7.64285	7.21364	7.96799	7.96749	7.96791
4.2.07	7.33883	7.49625	7.05831	7.96864	7.96756	7.96730
House	7.41527	7.22948	7.43538	7.96849	7.96825	7.96735

where

$$\begin{aligned}
 D(a) &= \frac{1}{M} \sum_{i=1}^M (a_i - \bar{a})^2, \\
 D(b) &= \frac{1}{M} \sum_{i=1}^M (b_i - \bar{b})^2, \\
 \text{cov}(a, b) &= \sum_{i=1}^M (a_i - \bar{a})(b_i - \bar{b}),
 \end{aligned}
 \tag{11}$$

M is the total number of couples (a_i, b_i) , obtained from the image, and \bar{a}, \bar{b} are the mean values of a_i and b_i , respectively. Correlation coefficient can range in the interval $[-1.00; +1.00]$.

Table 6 shows the results of horizontal, vertical, and diagonal adjacent pixels correlation coefficients calculations of the plain images and the corresponding encrypted images. It is clear that the novel image encryption scheme does not retain any linear dependencies between observed pixels in all three directions: the inspected horizontal, vertical, and diagonal correlation coefficients of the encrypted images are very close to zero. Overall, the correlation coefficients of

TABLE 6: Horizontal, vertical and diagonal correlation coefficients of adjacent pixels in plain images and encrypted images.

File name	Plain image correlation			Encrypted image correlation		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
4.1.01	0.956725	0.952503	0.937836	0.001274	0.001785	0.003044
4.1.02	0.908923	0.944135	0.889084	-0.007292	0.007162	0.004493
4.1.03	0.970861	0.916864	0.895799	-0.005882	-0.004236	0.002140
4.1.04	0.956759	0.964448	0.930833	-0.004143	-0.006414	0.008894
4.1.05	0.982138	0.974908	0.962532	0.005939	-0.001269	-0.002035
4.1.06	0.959183	0.934498	0.926566	0.003809	0.011929	-0.002274
4.1.07	0.988603	0.987932	0.979855	-0.008391	0.001379	-0.000308
4.1.08	0.977248	0.979839	0.958275	0.000991	-0.000089	-0.002968
4.2.01	0.978507	0.970863	0.964947	0.001250	-0.000860	0.001454
4.2.02	0.896888	0.909936	0.863983	0.000449	0.001230	-0.000765
4.2.03	0.907119	0.877498	0.839639	-0.001320	-0.000628	-0.000366
4.2.04	0.933223	0.958036	0.918587	-0.004386	0.000342	0.000569
4.2.05	0.962496	0.915378	0.914867	0.004689	0.000547	0.000136
4.2.06	0.969769	0.968659	0.953038	0.000850	0.005358	0.003821
4.2.07	0.964885	0.961169	0.948114	0.001554	-0.001897	0.002504
House	0.975076	0.959036	0.944382	0.001099	-0.002301	0.001799

TABLE 7: NPCR and UACI results of encrypted plain images and encrypted with one pixel difference plane images.

File name	NPCR test			UACI test		
	Red	Green	Blue	Red	Green	Blue
4.1.01	99.5701	99.5911	99.6155	33.6394	33.3493	33.4648
4.1.02	99.6613	99.5743	99.5804	33.4397	33.3669	33.4438
4.1.03	99.6094	99.6216	99.6109	33.3171	33.5476	33.3226
4.1.04	99.6323	99.6384	99.6155	33.4149	33.5338	33.3298
4.1.05	99.5987	99.6201	99.5743	33.4149	33.5670	33.4883
4.1.06	99.5693	99.5972	99.5705	33.4311	33.4601	33.4934
4.1.07	99.5941	99.6094	99.5972	33.5586	33.4576	33.4826
4.1.08	99.6490	99.6414	99.6536	33.4083	33.5591	33.4937
4.2.01	99.6346	99.6033	99.6311	33.4646	33.4548	33.4644
4.2.02	99.6120	99.6056	99.5894	33.4073	33.5304	33.4401
4.2.03	99.6197	99.6021	99.6273	33.4377	33.3873	33.4169
4.2.04	99.6109	99.6094	99.6185	33.4668	33.5337	33.3924
4.2.05	99.6025	99.5975	99.6101	33.4869	33.4080	33.4413
4.2.06	99.6136	99.5953	99.6361	33.3937	33.5238	33.5039
4.2.07	99.6426	99.6231	99.6300	33.4659	33.5129	33.4368
House	99.6357	99.6082	99.6220	33.3519	33.4210	33.4300

the proposed algorithm are similar to results of four other image encryption schemes [27–30].

4.2.6. *Differential Analysis.* In the main, a typical property of an image encryption scheme is to be sensitive to minor alterations in the plain images. Differential analysis supposes that an enemy is efficient to create small changes in the plain image and inspect the encrypted image. The alteration level can be measured by means of two metrics, namely, the number of pixels change rate (NPCR) and the unified average changing intensity (UACI) [30, 31].

Suppose encrypted images before and after one pixel change in plain image are C_1 and C_2 . The NPCR and UACI are defined as follows:

$$\begin{aligned}
 \text{NPCR} &= \frac{\sum_{i=0}^{W-1} \sum_{j=0}^{H-1} D(i, j)}{W \times H} \times 100\%, \\
 \text{UACI} &= \frac{1}{W \times H} \left(\sum_{i=0}^{W-1} \sum_{j=0}^{H-1} \frac{|C_1(i, j) - C_2(i, j)|}{255} \right) \times 100\%, \tag{12}
 \end{aligned}$$

where D is a two-dimensional array, having the same size as image C_1 or C_2 , and W and H are the width and height of the

TABLE 8: Correlation coefficients between the corresponding pixels of the 48 different encrypted images obtained from the 16 plain images by using the three slightly different secret keys: K1, K2, and K3.

Encrypted image 1	Encrypted image 2	Correlation coefficient	Encrypted image 1	Encrypted image 2	Correlation coefficient
4.1.01.K1	4.1.01.K2	0.006252	4.2.01.K1	4.2.01.K3	0.001802
4.1.02.K1	4.1.02.K2	0.002129	4.2.02.K1	4.2.02.K3	0.000867
4.1.03.K1	4.1.03.K2	0.006434	4.2.03.K1	4.2.03.K3	0.001430
4.1.04.K1	4.1.04.K2	0.001634	4.2.04.K1	4.2.04.K3	0.000064
4.1.05.K1	4.1.05.K2	-0.001745	4.2.05.K1	4.2.05.K3	0.003107
4.1.06.K1	4.1.06.K2	-0.005686	4.2.06.K1	4.2.06.K3	-0.001260
4.1.07.K1	4.1.07.K2	0.000907	4.2.07.K1	4.2.07.K3	-0.001401
4.1.08.K1	4.1.08.K2	-0.003864	House.K1	House.K3	-0.001986
4.2.01.K1	4.2.01.K2	0.000299	4.1.01.K2	4.1.01.K3	-0.002088
4.2.02.K1	4.2.02.K2	0.001053	4.1.02.K2	4.1.02.K3	-0.001454
4.2.03.K1	4.2.03.K2	0.000103	4.1.03.K2	4.1.03.K3	-0.003497
4.2.04.K1	4.2.04.K2	0.001290	4.1.04.K2	4.1.04.K3	-0.002121
4.2.05.K1	4.2.05.K2	0.000557	4.1.05.K2	4.1.05.K3	-0.002167
4.2.06.K1	4.2.06.K2	-0.000290	4.1.06.K2	4.1.06.K3	0.000598
4.2.07.K1	4.2.07.K2	0.001601	4.1.07.K2	4.1.07.K3	0.004454
House.K1	House.K2	0.000905	4.1.08.K2	4.1.08.K3	0.001396
4.1.01.K1	4.1.01.K3	-0.001953	4.2.01.K2	4.2.01.K3	0.004092
4.1.02.K1	4.1.02.K3	0.002054	4.2.02.K2	4.2.02.K3	-0.000099
4.1.03.K1	4.1.03.K3	0.004989	4.2.03.K2	4.2.03.K3	0.000007
4.1.04.K1	4.1.04.K3	0.001796	4.2.04.K2	4.2.04.K3	0.000170
4.1.05.K1	4.1.05.K3	-0.000826	4.2.05.K2	4.2.05.K3	0.002596
4.1.06.K1	4.1.06.K3	0.004114	4.2.06.K2	4.2.06.K3	0.003894
4.1.07.K1	4.1.07.K3	-0.000977	4.2.07.K2	4.2.07.K3	-0.001332
4.1.08.K1	4.1.08.K3	-0.000203	House.K2	House.K3	-0.000282

image. The array $D(i, j)$ is defined by $C_1(i, j)$ and $C_2(i, j)$; if $C_1(i, j) = C_2(i, j)$, then $D(i, j) = 1$; otherwise, $D(i, j) = 0$. The NPCR and UACI test results from the proposed chaos based algorithm are shown in Table 7.

The obtained NPCR values for the images from 4.1.01 to 4.1.08 are larger than critical values $N_{0.05}^* = 99.5693$, $N_{0.01}^* = 99.5527$, and $N_{0.001}^* = 99.5341$ and, for the images from 4.2.01 to 4.2.07 and House image, are larger than critical values $N_{0.05}^* = 99.5893$, $N_{0.01}^* = 99.5810$, and $N_{0.001}^* = 99.5717$ [31].

The obtained UACI values for the images from 4.1.01 to 4.1.08 are in the intervals from $N_{0.05}^{*-} = 33.2824$ to $N_{0.05}^{*-} = 33.6447$, from $N_{0.01}^{*-} = 33.2255$ to $N_{0.01}^{*-} = 33.7016$, and from $N_{0.001}^{*-} = 33.1594$ to $N_{0.001}^{*-} = 33.7677$. The obtained UACI values for the images from 4.2.01 to 4.2.07 and House image are in the intervals from $N_{0.05}^{*-} = 33.3730$ to $N_{0.05}^{*-} = 33.5541$, from $N_{0.01}^{*-} = 33.3445$ to $N_{0.01}^{*-} = 33.5826$, and from $N_{0.001}^{*-} = 33.3115$ to $N_{0.001}^{*-} = 33.6156$ [31].

The results from NPCR and UACI computations indicate that the new image encryption scheme is highly sensitive with respect to small changes in the plain images and has a strong ability of resisting differential cryptanalysis.

4.2.7. Key Sensitivity Test. Another important component of correlation analysis is the key sensitivity test. A good image encryption algorithm should be sensitive with respect to the secret key, that is, a slight modification of the secret key. We encrypted the 16 images with three similar secret keys: K1,

TABLE 9: Time test (seconds).

Image size	Reference [24]	Reference [32]	Reference [33]	Our scheme
256 × 256	0.22	1.34	0.35	0.19
512 × 512	1.04	5.26	0.72	0.61

K2 ($x(0) = 0.9798292345346$, $y(0) = -0.4032920230495034$, $k = 2.995$, $l = 3.07$, $u_{1,0} = -0.04$, $v_{1,0} = 0.2$, $u_{2,0} = 0.23$, and $v_{2,0} = -0.13$), and K3 ($x(0) = 0.9798292345347$, $y(0) = -0.4032920230495034$, $k = 2.995$, $l = 3.07$, $u_{1,0} = -0.04$, $v_{1,0} = 0.2$, $u_{2,0} = 0.23$, and $v_{2,0} = -0.13$). The results are shown in Table 8. It is evident that the proposed image encryption is highly key sensitive: the calculated correlation coefficients are very close to 0.00.

Moreover, in Figure 3, the results of two tests are shown to decrypt the Figure 1(b), with the secret keys K2 and K3.

We observed that the two decrypted images (Figure 3(a) and Figure 3(b)) have no relation with the plain image 4.2.05, Figure 1(a).

4.2.8. Speed Test. We have measured the encryption time for 256 × 256 and 512 × 512 sized images by using the novel image encryption algorithm. Speed analysis has been done on 2.8 GHz Pentium IV personal computer. In Table 9, we compared the speed of our method with [24, 32, 33]. The

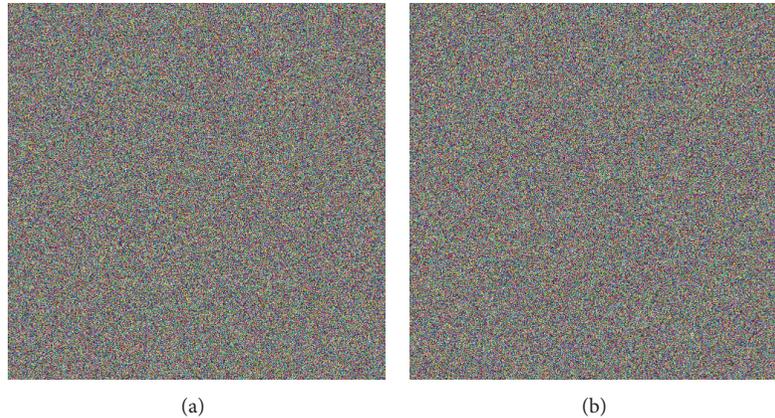


FIGURE 3: Decryption of Image 4.2.05 Airplane (F-16): (a) encrypted with key K1 and decrypted using key K2 and (b) encrypted with key K1 and decrypted using key K3.

data show that the proposed image encryption scheme has a satisfactory speed.

5. Conclusions

A novel image encryption algorithm based on dynamical chaotic systems is proposed in this paper. The developed encryption scheme combines Chebyshev polynomial based permutation and substitution and Duffing map based substitution. A precise security analysis on the novel encryption algorithm is given. Based on the experimental results of our computations, we conclude that the proposed chaos based image encryption technique is perfectly suitable for the practical image encryption.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the Project BG05IPO001-3.3.06-0003 “Building and steady development of PhD students, post-PhD and young scientists in the areas of the natural, technical and mathematical sciences.” The project is realized by the financial support of the Operative Program “Development of the human resources” of the European social fund of the European Union.

References

- [1] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.
- [2] L. Kocarev, J. Makraduli, and P. Amato, “Public-key encryption based on Chebyshev polynomials,” *Circuits, Systems, and Signal Processing*, vol. 24, no. 5, pp. 497–517, 2005.
- [3] E. Yoon and I. Jeon, “An efficient and secure Diffie-Hellman key agreement protocol based on Chebyshev chaotic map,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 6, pp. 2383–2389, 2011.
- [4] M. Mishra and V. H. Mankar, “Chaotic cipher using Arnolds and Duffings map,” in *Advances in Computer Science, Engineering & Applications*, D. Wyld, J. Zizka, and D. Nagamalai, Eds., vol. 167 of *Advances in Intelligent Systems and Computing*, pp. 529–539, Springer, Berlin, Germany, 2012.
- [5] J. Liu and H. A. Lv, “New Duffing-Lorenz chaotic algorithm and its application in image encryption,” in *Proceedings of the IEEE International Conference on Control Engineering and Communication Technology*, pp. 1022–1025, Liaoning, China, 2012.
- [6] C. Fu, P. Wang, X. Ma, Z. Xu, and W. A. Zhu, “Fast pseudo stochastic sequence quantification algorithm based on Chebyshev map and its application in data encryption,” in *Computational Science—ICCS 2006*, V. Alexandrov, G. Albada, P. Sloot, and J. Dongarra, Eds., vol. 3991 of *Lecture Notes in Computer Science*, pp. 826–829, Springer, Berlin, Germany, 2006.
- [7] H. Lai, J. Xiao, L. Li, and Y. Yang, “Applying semigroup property of enhanced Chebyshev polynomials to anonymous authentication protocol,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 454823, 17 pages, 2012.
- [8] F. Zhao, P. Gong, S. Li, M. Li, and P. Li, “Cryptanalysis and improvement of a three-party key agreement protocol using enhanced Chebyshev polynomials,” *Nonlinear Dynamics*, vol. 74, no. 1-2, pp. 419–427, 2013.
- [9] J. Fridrich, “Symmetric ciphers based on two-dimensional chaotic maps,” *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 8, no. 6, pp. 1259–1284, 1998.
- [10] S. Etemadi Borujeni and M. Eshghi, “Chaotic image encryption design using Tompkins-Paige algorithm,” *Mathematical Problems in Engineering*, vol. 2009, Article ID 762652, 22 pages, 2009.
- [11] X. Wang, L. Teng, and X. Qin, “A novel colour image encryption algorithm based on chaos,” *Signal Processing*, vol. 92, no. 4, pp. 1101–1108, 2012.
- [12] H. Yau, T. Hung, and C. Hsieh, “Bluetooth based chaos synchronization using particle swarm optimization and its applications to image encryption,” *Sensors*, vol. 12, pp. 7468–7484, 2012.
- [13] C. Fu, J. J. Chen, H. Zou, W. H. Meng, Y. F. Zhan, and Y. W. Yu, “A chaos-based digital image encryption scheme with an improved diffusion strategy,” *Optics Express*, vol. 20, no. 3, pp. 2363–2378, 2012.
- [14] C. Fu, B. B. Lin, Y. S. Miao, X. Liu, and J. J. Chen, “A novel chaos-based bit-level permutation scheme for digital image

- encryption,” *Optics Communications*, vol. 284, no. 23, pp. 5415–5423, 2011.
- [15] X. Huang, “Image encryption algorithm using chaotic Chebyshev generator,” *Nonlinear Dynamics*, vol. 67, no. 4, pp. 2411–2417, 2012.
- [16] N. Lin, X. Guo, P. Xu, and Y. A. Wang, “New multi-chaos based image encryption algorithm,” in *Advances in Intelligent Systems and Computing*, Z. Du, Ed., pp. 215–221, Springer, Berlin, Germany, 2013.
- [17] A. Kadir, A. Hamdulla, and W. Guo, “Color image encryption using skew tent map and hyper chaotic system of 6th-order CNN,” *International Journal For Light and Electron Optics*, vol. 125, pp. 1671–1675, 2014.
- [18] Y. Zhang and D. Xiao, “An image encryption scheme based on rotation matrix bit-level permutation and block diffusion,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, pp. 74–82, 2014.
- [19] L. Zhang, “Cryptanalysis of the public key encryption based on multiple chaotic systems,” *Chaos, Solitons and Fractals*, vol. 37, no. 3, pp. 669–674, 2008.
- [20] A. Rukhin, J. Soto, J. Nechvatal et al., A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Application. NIST Special Publication 800-22, Revision 1a, Lawrence E. Bassham III, 2010, <http://csrc.nist.gov/rng/>.
- [21] G. Marsaglia, The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness. Florida State University, 1995, <http://www.stat.fsu.edu/pub/diehard/>.
- [22] J. Walker, ENT: A Pseudorandom Number Sequence Test Program. 2008, <http://www.fourmilab.ch/random/>.
- [23] T. Gao and Z. Chen, “Image encryption based on a new total shuffling algorithm,” *Chaos, Solitons and Fractals*, vol. 38, no. 1, pp. 213–220, 2008.
- [24] N. Pareek, V. Patidar, and K. Sud, “Diffusion-substitution based gray image encryption scheme,” *Digital signal processing*, vol. 23, pp. 894–901, 2013.
- [25] IEEE Computer Society. IEEE 754: standard for binary floating-point arithmetic, 1985.
- [26] G. Alvarez and S. Li, “Some basic cryptographic requirements for chaos-based cryptosystems,” *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.
- [27] J. Chen, Z. Zhu, C. Fu, and H. Yu, “An improved permutation-diffusion type image Cipher with a Chaotic orbit perturbing,” *Optics Express*, vol. 21, pp. 27873–27890, 2013.
- [28] S. Al-Maadeed, A. Al-Ali, and T. Abdalla, “A new chaos-based image-encryption and compression algorithm,” *Journal of Electrical and Computer Engineering*, vol. 2012, Article ID 179693, 11 pages, 2012.
- [29] A. Diaconu and K. Loukhaoukha, “An improved secure image encryption algorithm based on Rubik’s cube principle and digital Chaotic Cipher,” *Mathematical Problems in Engineering*, vol. 2013, Article ID 848392, 10 pages, 2013.
- [30] G. Chen, Y. Mao, and C. K. Chui, “A symmetric image encryption scheme based on 3D chaotic cat maps,” *Chaos, Solitons and Fractals*, vol. 21, no. 3, pp. 749–761, 2004.
- [31] Y. Wu, J. P. Noonan, and S. Agaian, “NPCR and UACI randomness tests for image encryption,” *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications*, vol. 2, pp. 31–38, 2011.
- [32] X. Tong and M. Cui, “Image encryption with compound chaotic sequence cipher shifting dynamically,” *Image and Vision Computing*, vol. 26, no. 6, pp. 843–850, 2008.
- [33] C. Fu, W. Meng, Y. Zhan et al., “An efficient and secure medical image protection scheme based on chaotic maps,” *Computers in Biology and Medicine*, vol. 43, pp. 1000–1010, 2013.

Research Article

Privacy-Preserving Location-Based Query Using Location Indexes and Parallel Searching in Distributed Networks

Cheng Zhong, Lei Liu, and Jing Zhao

School of Computer and Electronics and Information, Guangxi University, Nanning, Guangxi 530004, China

Correspondence should be addressed to Cheng Zhong; chzhong@gxu.edu.cn

Received 5 January 2014; Accepted 26 February 2014; Published 25 March 2014

Academic Editors: T. Cao, M. Ivanovic, and F. Yu

Copyright © 2014 Cheng Zhong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An efficient location-based query algorithm of protecting the privacy of the user in the distributed networks is given. This algorithm utilizes the location indexes of the users and multiple parallel threads to search and select quickly all the candidate anonymous sets with more users and their location information with more uniform distribution to accelerate the execution of the temporal-spatial anonymous operations, and it allows the users to configure their custom-made privacy-preserving location query requests. The simulated experiment results show that the proposed algorithm can offer simultaneously the location query services for more users and improve the performance of the anonymous server and satisfy the anonymous location requests of the users.

1. Introduction

Recently, with the development of the mobile wireless communication location technology, the location-based service is emerged. The location information of the users is made of their identifiers and temporal and spatial information [1]. Another important problem related to the location information services is to preserve the location privacy of the user [2]. Using the anonymity on the location-based services [3] is a direct and effective method to prevent the quasi identifiers of the users. Gruteser et al. [4] introduced the k -anonymity model to investigate the problem of preserving the location privacy of the users. Kido et al. [5] used the dummies to study the anonymous communication technique for the location-based services.

One of the key issues for the privacy-preserving location-based services in the distributed networks is to balance the quality of query services and the privacy protection of the users. In this paper, we will propose an efficient privacy-preserving location-based query algorithm using parallel searching to improve the efficiency of the anonymous server, which not only can protect the location privacy of the user but also obtain the location query services. The remainder of this paper is organized as follows. In Section 2, we give the related work about the privacy-preserving location-based

techniques. In Section 3, we propose an efficient privacy-preserving location-based query algorithm using location indexes and parallel searching in the distributed networks. Section 4 reports the simulated experimental results. Section 5 concludes the paper.

2. Related Work

By applying the personalized k -anonymity model, Gedik and Liu [6] proposed the architecture and the algorithms to protect the location privacy of the user. Chow et al. [7] proposed a distributed k -anonymity model and a peer-to-peer spatial cloaking algorithm for the anonymous location-based services. Ghinita et al. [8] investigated the anonymous location-based query method in the distributed mobile systems. By using the distributed hash table to select the anonymous set of the users, Ghinita et al. [9] implemented the anonymous location-based query services in the mobile P2P system. Zhong and Hengartner [10] used the secure multiparty computation protocol to design a distributed k -anonymity protocol for protecting the location privacy. By using the obfuscation method and vague location information of the user, Duckham and Kulik [11] presented a privacy-preserving location query algorithm. Mokbel [12] proposed a location-obfuscation method which allows the

server to record the real identifier of the user but decreases the precision of the location information to protect the location privacy. By introducing the trusted third party, Mokbel et al. [13] proposed a location service query method without compromising privacy.

By using the space transformation, Khoshgozaran and Shahabi [14] gave a blind evaluation of the nearest neighbor query to protect the location privacy. Ghinita et al. [15] studied the private query method in the location-based services by partitioning the space into several areas and mapping these areas into the points in Hilbert curve. Pietro and Viejo [16] developed a probabilistic and scalable protocol which guarantees the location privacy of the sensors replying to the query. Raj et al. [17] proposed a realistic semiglobal eavesdropping attack model and showed its effectiveness in compromising an existing source-location preserving technique and designed a new protocol which preserves α -angle anonymity by adapting the conventional function of data mules. Zhao et al. [18] developed the optimal solutions to some special cases through dynamic programming and several heuristics for the general case to the location privacy-preserving problem. Pingley et al. [19] implemented a context-aware privacy-preserving location-based services system with integrated protection for both data privacy and communication anonymity and integrated it with Google Maps. Tan [20] proposed a conditional privacy-preserving authentication and access control scheme for the pervasive computing environments, in which the registration servers and authentication servers do not need to maintain any sensitive verification tables. Xi et al. [21] showed that the privacy-preserving shortest path routing problem can be solved with the private information retrieval techniques without disclosing the origin or the destination.

By introducing local suppression to trajectory data anonymization to enhance the resulting data utility, Chen et al. [22] obtained a $(K, C)_L$ -privacy model on trajectory data without paying extra utility and computation cost and proposed an anonymization framework that is independent of the underlying data utility metrics and is suitable for different trajectory data mining workloads. Based on extending the private equality primitive, Buchanan et al. [23] presented a novel encryption method for preserving the location and trajectory path of a user by privacy-enhancing technologies, which has significant improvement in the computation speed. Cicek et al. [24] grouped the points of interest to create obfuscation areas around sensitive locations and used the map anonymization as a model to anonymize the trajectories and proposed a new privacy metric p -confidentiality that ensures location diversity by bounding the probability of a user visiting a sensitive location with the p input parameters. Li and Jung [25] proposed a fine-grained privacy-preserving location query protocol (PLQL) to solve the privacy issues in existing LBS applications and provide various location-based queries. The protocol PLQL can implement semi-functional encryption by novel distance computation and comparison protocol and support multilevel access control. Dewri and Thurimella [26] proposed a user-centric location-based service architecture, that the users can observe the impact of location inaccuracy on the service accuracy, and

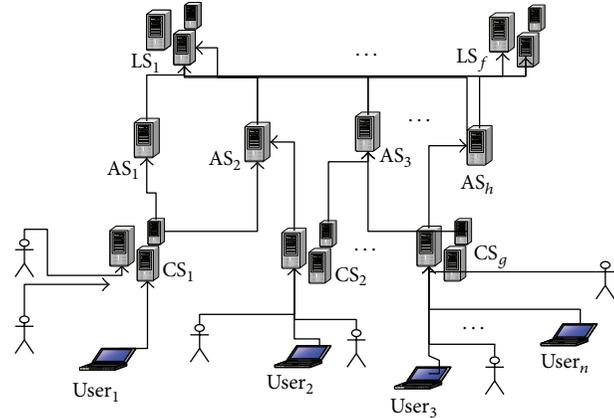


FIGURE 1: Location-based services system with the anonymous server.

constructed a local search application and demonstrated how the meaningful information can be exchanged between the user and the service provider to allow the inference of contours depicting the change in the query results across a geographic area.

3. Privacy-Preserving Location-Based Services System and Algorithm

3.1. Anonymous Location Query Services System. The privacy-preserving location-based services system in the distributed networks includes mobile users, communication services providers CS, and location service providers LS, in which the independent trusted third party will provide the anonymous servers AS [6], which is described in Figure 1.

The anonymous location-based query process is as follows.

Step 1. The users acquire their locations (x, y, r) via the communication services provider, where x and y are the two-dimensional location coordinates of the users, respectively, and r represents the location precision.

Step 2. The users send the service request information $(\text{Uid}, (x, y, r), \text{profile}(A_{\min}, A_{\max}, K_s, K_t, t, \text{Pre}), \text{Cont})$ to the anonymous server, where Uid is the identifier of the user, (x, y, r) is the current location information of the user, $\text{profile}(A_{\min}, A_{\max}, K_s, K_t, t, \text{Pre})$ represents the configuration file of the users, A_{\min} and A_{\max} denote the minimum and maximum requirements for anonymous areas, K_s and K_t are the temporal and spatial anonymous requests, respectively, t is the service time demand, Pre represents the set to the anonymity priority or services priority, and Cont is the content of the query.

Step 3. The anonymous server receives the request from the user, generates the anonymous sets, and sends the information $((X, Y, R), (\text{zid}_1, \text{Cont}_1), \dots, (\text{zid}_k, \text{Cont}_k))$ to the location service server, where (X, Y, R) is the anonymous area and zid_i is the i th anonymous identifier of the user and Cont_i

represents the content of the i th request from the user, $i = 1 \sim k$.

Step 4. The location service server receives the requests of the user and returns the processed results $(\text{zid}_1, \text{result}_1), \dots, (\text{zid}_k, \text{result}_k)$ to the anonymous server, and the anonymous server sends the transformed ID result to the user.

3.2. Privacy-Preserving Location Query Algorithm. Assume that the k users want to request location-based query services and the i th anonymous request is $k_i, k \geq \max\{k_i\}$. If the users are evenly distributed in the space range, the probability that their request information can be guessed will be $1/k$ and the probability that the actual locations of the users can be guessed will be $1/(\pi R^2)$, respectively. We know the more the users in the space range, the more the anonymous requests and the larger the generated anonymous area, the better the anonymous effect. But the computational cost to search the anonymous space will increase, and the quality of obtained location-based services may be relatively poor. The multiple searching threads are executed in parallel to accelerate the generation of the candidate anonymous set for each request queue and compute the density $\rho = k/(\pi R^2)$ of the user for all the candidate anonymous sets and the distribution of the users in the anonymous sets $C = |(N_1 + N_2) - (N_3 + N_4)| + |(N_1 + N_4) - (N_3 + N_2)|$, where N_j is the number of the users in the j th quadrant among the four partitioned quadrants, $j = 1 \sim 4$.

When the location anonymous server has received the request from the user, it searches the location indexes in B-tree and inserts the location information into the request queue. If it is necessary to establish a new request queue, the location indexes will be updated. The multiple threads search in parallel and select quickly the anonymous areas in the request queues. The anonymous server handles the selected anonymous areas and provides the appropriate location services for the users.

To establish the bidirectional indexes, each element in the request queues is arranged into the form $(\text{Uid}, \langle x, y, r \rangle, \langle R_{\min}, R_{\max}, k_s, t, k_t, \text{pre} \rangle, \text{Cont}, *next, *pre)$. The performance of the anonymous server is directly affected by the number of the request queues on the anonymous server. We assume that there are n location query requests and n request queues on the anonymous server; the n location query requests are evenly distributed in the range with area S and the maximum anonymous radius R , and the number of the request queues is $S/(\pi R^2)$. B-tree is used to construct the location indexes with the directions X and Y on the anonymous server. The two main algorithms running on the anonymous server are the Request Enqueue Algorithm and Anonymous Set Generation Algorithm, which are described as follows.

Algorithm 1. Request Enqueue Algorithm.

Begin

- (1) The request $(\text{Uid}, \langle x, y, r \rangle, \langle R_{\min}, R_{\max}, k_s, t, k_t, \text{pre} \rangle, \text{Cont})$ is received and it is expanded to $(\text{Uid}, \langle x, y, r \rangle, \langle R_{\min}, R_{\max}, k_s, t, k_t, \text{pre} \rangle, \text{Cont}, *next, *pre)$.

- (2) B-tree indexes with the condition $|X - x| < 2R_{\max}$ along the direction X is searched.

- (2.1) If the searching in the direction X is unsuccessful, the request is inserted into the queue, the indexes in the direction X are updated, and the indexes in the direction Y are added.

- (2.2) If the searching in the direction X is successful, B-tree index with the condition $|Y - y| < 2R_{\max}$ along the direction Y is searched.

- (2.2.1) If the searching in the direction Y is unsuccessful, the current request is inserted into the request queue and the indexes in the direction Y are updated.

- (2.2.2) If the searching in the direction Y is successful, the current request is inserted into the request queue in the chronological order.

End.

Algorithm 2. Anonymous Set Generation Algorithm.

Begin

- (1) The temporal-spatial queue L_T is constructed, which each element in L_T links a request queue.

- (2) The Request Enqueue Algorithm is executed to generate a new request queue $(\text{Uid}, \langle x, y, r \rangle, \langle R_{\min}, R_{\max}, k_s, t, k_t, \text{pre} \rangle, \text{Content}, *next, *pre)$ and this request queue is inserted into queue L_T in the chronological order, where s and t represent the space and time respectively.

- (3) Each element in queue L_T is searched, and multiple threads are generated according to the condition $|T - T_s| < \delta$, where T is the time when the element L in queue L_T wants to generate the request queue, T_s is the current time of the running system, and δ is the threshold. Each request queue is assigned to a thread.

- (4) Multiple threads are run in parallel, and each thread is responsible for the following operations.

- (4.1) If the number of the elements in the request queue is smaller than k , the elements which satisfy the condition $|T - T_s| < \delta$ are searched and those elements with priority pre are deleted to form request queue set Ω . When Ω is not empty, the density ρ of the user is queried by the communication service provider, anonymous area A and radius R_{xm} with the minimum anonymous request are computed, and the anonymous request set is generated by the radius R_{xm} and the centroid of all elements in set Ω . The ID disturbing algorithm is executed to disturb the ID of the user and the anonymous request set is submitted to the location services server.

(4.2) If the number of the elements in the request queue is larger than k , m threads are generated, where m is the number of elements in the request queue. Each thread executes the following operations.

(4.2.1) If the three points (x_1, y_1) , (x_2, y_2) and (x_3, y_3) in the anonymous range are located in a straight line, the coordinate of the center in the anonymous request set is $(x_0 = (x_1 + x_2 + x_3)/3, y_0 = (y_1 + y_2 + y_3)/3)$; if not, the center of the circum of the triangle with coordinates (x_1, y_1) , (x_2, y_2) and (x_3, y_3) is the center in the anonymous request set.

(4.2.2) If $R_{xm} \leq \min\{R_{\max}\}$, a candidate anonymous area with circle center (x_0, y_0) and radius $\min\{R_{\max}\}$ is generated.

(4.2.3) Number s of the elements in the circle is computed, and the farthest point from the circle center and its distance D_{\max} are recorded. If $s < k$, then report failure.

(4.2.4) If the number of the elements which satisfy the anonymous request is also smaller than k , then report failure.

(4.2.5) If $x_j - x_0 > 0$ and $y_j - y_0 > 0$ then $N_1 = N_1 + 1$, if $x_j - x_0 < 0$ and $y_j - y_0 > 0$ then $N_2 = N_2 + 1$, if $x_j - x_0 < 0$ and $y_j - y_0 < 0$ then $N_3 = N_3 + 1$, and if $x_j - x_0 > 0$ and $y_j - y_0 < 0$ then $N_4 = N_4 + 1$, $j = 1 \sim 3$.

(4.2.6) If each element within the circle satisfies the anonymous request, then $\Delta = \min\{R_{\max}\} - D_{\max}$ is computed. If Δ goes beyond the threshold, the radius of the circle is reduced until Δ is in the threshold. Finally, the new radius R_0 is obtained.

(4.2.7) The anonymous area $A((x_0, y_0), R_0)$, the set Q including all the request elements in this area and number N_Q of the elements in set Q are returned, and the density $\rho = N_Q/(\pi R_0^2)$ of the users in the anonymous set is computed.

(4.3) The ID disturbing algorithm is executed to disturb the ID of the user, set Q is submitted to the location service server, and queue L_T is renewed by the elements which are not in set Q and the location indexes are updated.

End.

4. Experiment

We used a multicore computer to simulate the anonymous server and the PC computers to simulate the users to request concurrently the location services. Redhat 5.1 and MySQL 5.5 are run on the anonymous server, respectively, and Ubuntu 10.04 is run on the clients. The presented algorithms are implemented by Java programming with JDK7.0 and socket communication.

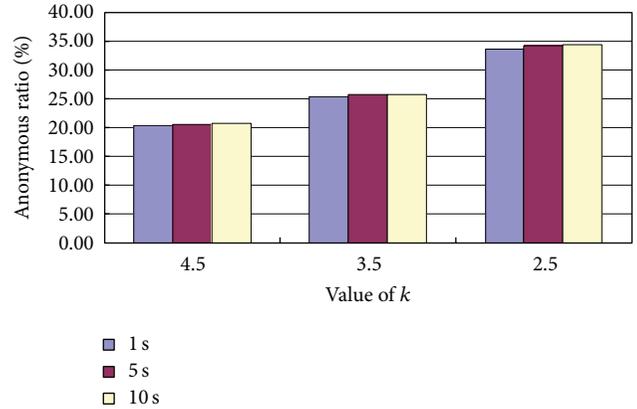


FIGURE 2: Ratio of the temporal-spatial anonymity with different waiting time and average anonymous request.

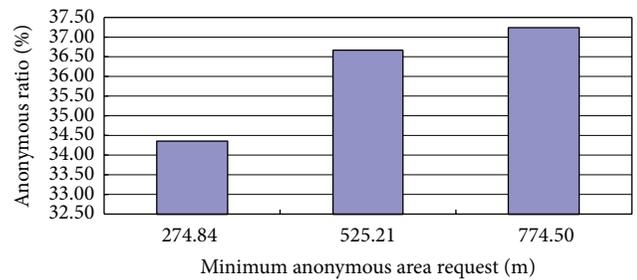


FIGURE 3: Ratio of the temporal-spatial anonymity with different anonymous space requests.

The Thomas Brinkhoff road network data generator is applied to produce the location service requests, and the OldenBurg urban communication network information is used as the input data of the road network data generator. The anonymous server deals with the location query and the anonymous requests from the users. The value of pre is set to service priority. The values of the relative experimental parameters are listed in Table 1.

We first test that the waiting time of the user and the average anonymous value of k are how to impact the ratio of the temporal-spatial anonymity. The obtained simulation experimental result is given in Figure 2.

From Figure 2, we can see that the longer the waiting time of the user, the higher the ratio of the temporal-spatial anonymity and the smaller the average anonymous request, the higher the ratio of the temporal-spatial anonymity.

The result in Figure 3 shows that the larger the anonymous space request, the higher the ratio of the temporal-spatial anonymity and the ratio of the temporal-spatial anonymity changes significantly along with the increase of the anonymous space request. This illustrates that the different anonymous space requests will affect remarkably the ratio of the temporal-spatial anonymity.

The required processing time and the anonymous area about our algorithm and the Bottom_up algorithm [13] are shown in Figures 4 and 5, respectively, where the minimum anonymous range partitioned some small square areas with

TABLE 1: Experimental parameters.

Average service delay request (second)	Average location precision (mile)	Request amount	Average spatial request (k_s)	Average temporal request (k_t)	Minimum radius R_{\min} in average anonymous area (mile)	Maximum radius R_{\max} in average anonymous area (mile)
1	50.03	466034	4.49	4.50	274.94	637.11
1	50.04	503432	3.50	3.50	275.43	637.43
1	50.08	453293	2.50	2.50	275.10	636.67
5	50.06	457712	4.50	4.50	275.08	637.25
5	49.98	446796	3.50	3.50	275.19	636.99
5	50.01	442778	2.50	2.50	275.12	637.86
10	50.08	456924	4.50	4.50	274.79	637.74
10	49.93	697428	3.50	3.49	275.01	637.54
10	49.98	681940	2.50	2.49	274.84	637.43
10	50.00	493648	2.50	2.50	525.21	1263.51
10	49.97	455932	2.49	2.50	774.50	1387.06
20	49.97	448366	2.50	2.51	275.27	637.64
30	50.03	472418	2.50	2.50	275.37	637.87

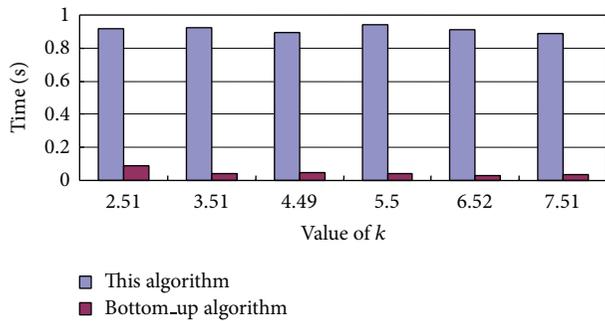


FIGURE 4: Required processing time for our algorithm and the Bottom_up algorithm.

a length of 300 m of a side and 3 users are initially contained within the minimum anonymous range.

We can see from Figure 4 that the required processing time for the Bottom_up algorithm is much less than the required time for our algorithm. This is because our algorithm wants to process more location service requests than the Bottom_up algorithm in order to achieve better privacy-preserving effect.

The results in Figure 5 show that along with the increase of the value of k , the anonymous area for the Bottom_up algorithm is increased, but the anonymous area for our algorithm is relatively stable; when the value of k is larger than 5.5, the anonymous area for our algorithm is smaller and the quality of the anonymous location service is better; in other words, the degree of privacy protection for our algorithm is higher.

Figure 6 gives the size of the processed anonymous data, in which our algorithm and the Bottom_up algorithm are executed in 20 minutes.

We can see from Figure 6 that, if there are adequate location service requests, our presented algorithm executes

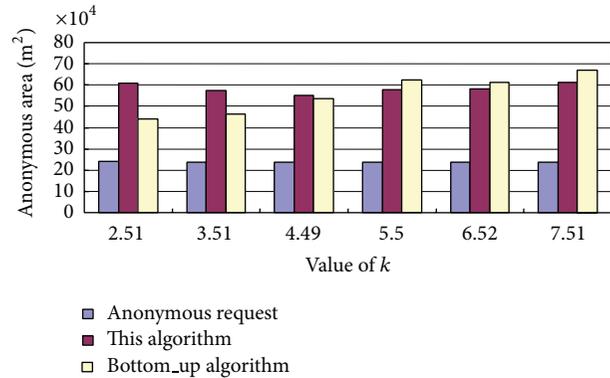


FIGURE 5: Anonymous area of our algorithm and the Bottom_up algorithm.

multiple parallel threads to search quickly the candidate anonymous sets and it can process more location service requests than the Bottom_up algorithm. That is to say, our algorithm can offer simultaneously services for more users.

5. Conclusion

The main contribution of this paper is to establish the location request queues according to the location indexes of the users such that the size of searching information can be remarkably reduced when the anonymous operations are executed and the selection of the anonymous sets on the anonymous server can be speeded up by executing multiple threads to search in parallel the candidate anonymous sets. The presented efficient privacy-preserving location-based query algorithm can obtain better location information services. The next work is to integrate the anonymous locations and the trajectory services into cartographic information and history data

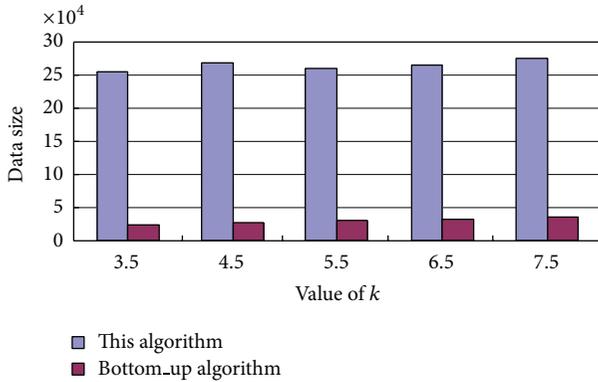


FIGURE 6: Size of processed anonymous data for our algorithm and the Bottom_up algorithm.

to develop the trajectory privacy-preserving method in the distributed networks.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This paper is supported by Guangxi Natural Science Foundation under Grant no. 2011GXNSFA018152.

References

- [1] J. P. Baugh and J. Guo, "Location privacy in mobile computing environments," in *Ubiquitous Intelligence and Computing*, J. Ma, H. Jin, L. T. Yang, and J. J. P. Tsai, Eds., vol. 4159 of *Lecture Notes in Computer Science*, pp. 936–945, Springer, Berlin, Germany, 2006.
- [2] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.
- [3] A. Pfitamann and M. Köhntopp, "Anonymity, unobservability, and pseudonymity—a proposal for terminology," in *Designing Privacy Enhancing Technologies*, vol. 2009 of *Lecture Notes in Computer Science*, pp. 1–9, Springer, Berlin, Germany, 2001.
- [4] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services*, pp. 31–42, San Francisco, Calif, USA, May 2003.
- [5] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of the 2nd International Conference on Pervasive Services (ICPS '05)*, pp. 88–97, Santorini, Greece, July 2005.
- [6] B. Gedik and L. Liu, "Protecting location privacy with personalized k -anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 27, no. 1, pp. 1–18, 2008.
- [7] C.-Y. Chow, M. F. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS '06)*, pp. 171–178, Arlington, Va, USA, November 2006.
- [8] G. Ghinita, P. Kalnis, and S. Skiadopoulos, "PRIVE: anonymous location-based queries in distributed mobile systems," in *Proceedings of the 16th International World Wide Web Conference (WWW '07)*, pp. 371–380, Banff, Canada, May 2007.
- [9] G. Ghinita, P. Kalnis, and S. Skiadopoulos, "MobiHide: a mobile peer-to-peer system for anonymous location-based queries," in *Advances in Spatial and Temporal Databases*, vol. 4605 of *Lecture Notes in Computer Science*, pp. 221–238, Springer, Berlin, Germany, 2007.
- [10] G. Zhong and U. Hengartner, "A distributed k -anonymity protocol for location privacy," in *Proceedings of the 7th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '09)*, pp. 1–10, Galveston, Tex, USA, March 2009.
- [11] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proceedings of the 3rd International Conference on Pervasive Computing*, pp. 152–170, Munich, Germany, May 2005.
- [12] M. F. Mokbel, "Towards privacy-aware location-based database servers," in *Proceedings of the of 22nd International Conference on Data Engineering Workshops*, p. 93, Atlanta, Ga, USA, April 2006.
- [13] M. F. Mokbel, C. Y. Chow, and W. G. Aref, "The new casper: query processing for location services without compromising privacy," in *Proceedings of the of the 32nd International Conference on Very Large Data Bases*, pp. 763–774, Seoul, Republic of Korea, September 2006.
- [14] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in *Advances in Spatial and Temporal Databases*, vol. 4605 of *Lecture Notes in Computer Science*, pp. 239–257, Springer, Berlin, Germany, 2007.
- [15] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 121–132, Vancouver, Canada, June 2008.
- [16] R. D. Pietro and A. Viejó, "Location privacy and resilience in wireless sensor networks querying," *Computer Communications*, vol. 34, no. 3, pp. 515–523, 2011.
- [17] M. Raj, N. Li, D. Liu, M. Wright, and S. K. Das, "Using data mules to preserve source location privacy in Wireless Sensor Networks," *Pervasive and Mobile Computing*, 2012.
- [18] B. Zhao, D. Wang, Z. Shao, J. Cao, and J. Su, "Privacy aware publishing of successive location information in sensor networks," *Future Generation Computer Systems*, vol. 28, no. 6, pp. 913–922, 2012.
- [19] A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao, "A context-aware scheme for privacy-preserving location-based services," *Computer Networks*, vol. 56, no. 11, pp. 2551–2568, 2012.
- [20] Z. Tan, "A lightweight conditional privacy-preserving authentication and access control scheme for pervasive computing environments," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1839–1846, 2012.
- [21] Y. Xi, L. Schwiebert, and W. Shi, "Privacy preserving shortest path routing with an application to navigation," *Pervasive and Mobile Computing*, 2013.
- [22] R. Chen, B. C. M. Fung, N. Mohammed, B. C. Desai, and K. Wang, "Privacy-preserving trajectory data publishing by local

- suppression,” *Information Sciences*, vol. 231, no. 1, pp. 83–97, 2013.
- [23] W. J. Buchanan, Z. Kwecka, and E. Ekonomou, “A privacy preserving method using privacy enhancing techniques for location based services,” *Mobile Networks and Applications*, vol. 18, no. 5, pp. 728–737, 2013.
- [24] A. E. Cicek, M. E. Nergiz, and Y. Saygin, “Ensuring location diversity in privacy-preserving spatio-temporal data publishing,” *The VLDB Journal*, 2013.
- [25] X. Y. Li and T. Jung, “Search me if you can: privacy- preserving location query service,” in *Proceedings of the 32nd IEEE International Conference on Computer Communications (IEEE INFOCOM '13)*, pp. 2760–2768, Turin, Italy, April 2013.
- [26] R. Dewri and R. Thurimella, “Exploiting service similarity for privacy in location-based search queries,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 374–383.