Wireless Communications and Mobile Computing

Security and Privacy Challenges for Internet-of-Things and Fog Computing

Lead Guest Editor: Ximeng Liu
Guest Editors: Yang Yang, Raymond Kim-Kwang Choo, and Huaqun Wang

Security and Privacy Challenges for Internet-of-Things and Fog Computing

Wireless Communications and Mobile Computing

Security and Privacy Challenges for Internet-of-Things and Fog Computing

Lead Guest Editor: Ximeng Liu
Guest Editors: Yang Yang, Raymond Kim-Kwang Choo, and Huaqun Wang

Editorial Board

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9373961, 3 pages
https://doi.org/10.1155/2018/9373961

Editorial

Security and Privacy Challenges for Internet-of-Things and Fog Computing

Ximeng Liu,[1,2] **Yang Yang**,[2] **Kim-Kwang Raymond Choo**,[3] **and Huaqun Wang**[4]

[1]*School of Information Systems, Singapore Management University, Singapore 188065*
[2]*College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China*
[3]*College of Business, The University of Texas at San Antonio, San Antonio, TX 78249, USA*
[4]*School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*

Correspondence should be addressed to Ximeng Liu; snbnix@gmail.com

Received 5 September 2018; Accepted 5 September 2018; Published 24 September 2018

1. Introduction

Internet-of-Things (IoT) has been considered as a necessary part of our daily life with billions of IoT devices collecting data through wireless technology and can interoperate within the existing Internet infrastructure. The new fog computing paradigm allows storing and processing data at the network edge or anywhere along the cloud-to-endpoint continuum, and it also overcomes the limitations of IoT devices and allows us to design a far more capable architecture. Unfortunately, this new IoT-Fog paradigm faces many new security and privacy issues, such as secure communication, authentication and authorization, and information confidentiality. Although the traditional cloud-based platform can even use heavyweight cryptosystem to enhance the security, it cannot be performed on the resource-constrained fog devices directly. Moreover, millions of smart fog devices are wildly distributed and located in different areas, which increases the risk of being compromised by some malicious parties.

To address these arising challenges and opportunities different from traditional cloud-based architecture, all the papers chosen for this special issue represent recent progress in the field of security and privacy techniques relevant to the convergence of IoT with fog computing, including identity/attribute-based cryptography, system and software security, system and resource optimization, user privacy preservation, and data protection. Overall, our international editorial committee selected 17 papers among 70 submissions from both the theoretical and the practical side. All of these papers in this special issue not only provide novel ideas and state-of-the-art techniques in the field of IoT-Fog computing but also stimulate future research in the IoT-Fog computing environment.

2. Identity/Attribute-Based Cryptography

Identity-based cryptography is an attractive branch of public key cryptography which uses the public known information (such as, an email address or a physical IP address) as the public key. It arises more security and performance issues when meeting the IoT-Fog computing applications. The paper by Q. Wang et al., entitled "An Anonymous Multireceiver with Online/Offline Identity-Based Encryption", presented anonymous multireceiver online/offline identity-based encryption which could reduce the computational cost according to the online/offline encryption method suitable for the sender which had limited resources, such as mobile devices and sensor nodes. The paper by L. Zhu et al., entitled "An Efficient Identity-Based Proxy Blind Signature for Semioffline Services", presented a new proxy blind signature based on the mathematical structure called NTRU lattice, which could be independent of public key infrastructure and secure against quantum computers attack and was suitable for semioffline e-payment system and e-voting in the fog computing scenario. The proposed scheme has proven secure, that is, strongly identifiable and strongly undeniable. The

paper by W. Liu et al., entitled "Strong Identity-Based Proxy Signature Schemes, Revisited", introduced a practical attack; that is, malicious adversary can create a proxy signature on a message, if the adversary had access to the standard signature of the original signer and proxy signer. This attack had not been considered by the existing identity-based proxy signature schemes, which is greatly important for the IoT-Fog computing. Also, the authors proposed a construction that can effectively prevent this attack by transforming "normal" proxy signature scheme into "strong" one. As the extension of the identity-based cryptography, attributed-based cryptosystem can support secure one-to-many message transmission and fine-grained access control. The paper by Q. Li et al., entitled "Traceable Ciphertext-Policy Attribute-Based Encryption with Verifiable Outsourced Decryption in eHealth Cloud", presented a verifiable and traceable CP-ABE scheme in eHealth cloud. The proposed system could support the verifiable outsourced decryption and white-box traceability at the same time and could ensure the privacy of the user's identity. Also, the authors gave a delegation method to let the resource-limited devices (especially fit for the IoT-Fog computing) authorize someone else to interact with the cloud decryption server which is secure, efficient, and practical.

3. System and Software Security

System and software security is a crucial component to a device operating at its optimum from authentication and anti-virus protection to vulnerability exploitation and modifications. The system and software contain more security issues when meeting the real IoT-Fog devices. For wildly used Android devices, Z. Peng et al. gave a paper entitled "Hydra-Bite: Static Taint Immunity, Split, and Complot Based Information Capture Method for Android Device." The authors researched the Android system's application layer to use the permission split and reconstruct module to split traditional privacy stealing Trojan, and constructed a collaborative application group. The newly proposed Hydra-Bite could resist the detecting and killing of multiple antivirus programs, which has higher information capture rate and stronger anti-killing performance. To achieve the isolation and access control for virtual machines and IoT devices, J. Dong et al. proposed a paper entitled "Task-Oriented Multilevel Cooperative Access Control Scheme for Environment with Virtualization and IoT." In the scheme, each user of the platform created tasks which could be divided into multiple levels to limit access between virtual machines and IoT terminals. Moreover, the network isolation, process isolation, and shared memory isolation could further enhance security for virtual machines and IoT terminals. The paper by B. Tang et al. entitled "Niffler: A Context-Aware and User-Independent Side-Channel Attack System for Password Inference," presented a novel side-channel attack system according to the user-independent features of tapping consecutive buttons to reconstruct the unlocking passwords on smartphones. Also, the Niffler used a Markov model to model the unlocking process and used the sequences with the highest probabilities as the attack candidates, which achieves high password guessing accuracy with only several attempts and few training samples. The paper by D. Wang et al., entitled "Resetting Your Password Is Vulnerable: A Security Study of Common SMS-Based Authentication in IoT Device," is aimed at gaining the control of IoT devices without firmware analysis. The fundamental idea was based on the observation that most of the official applications (call APP) had a common feature which is using an SMS authentication code sent to client phone to authenticate the client when he forgot his password for the APP. The author implemented a prototype tool to enable performing such brute-force SMS authentication code attack on IoT devices automatically.

4. System and Resource Optimization

The IoT and fog devices are typically deployed in resource (energy, computational, storage) constrained environments. The adoption of fog computing with IoT has a lot of optimization shortcomings such as network reliability optimization, energy balancing, and task allocation. Y. Li et al. in their paper, entitled "Reliable Ant Colony Routing Algorithm for Dual-Channel Mobile Ad Hoc Networks," presented reliable path under dual-channel condition (DSAR) system which contained a dual-channel communication model and a hierarchical network model to improve network bandwidth and to optimize the dual layer network, respectively. Also, the ant colony algorithm was used in the system for changes of network topology adaptability. To solve the IoT energy balancing problem, Y. Wang et al. in their paper, entitled "Gleer: A Novel Gini-Based Energy Balancing Scheme for Mobile Botnet Retopology," presented a novel Gini based energy balancing scheme (Gleer) for the atomic network as the basis of the heterogeneous multi-layer mobile botnet. The authors categorized atomic network into multiple groups with the dynamic energy threshold, estimated botnet energy gap, and regulated the probability for each node with the Gini coefficients to estimate in the Gleer which could significantly reduce user' detection awareness. For the task allocation problem in mobile devices, the paper by W. Zhu et al., entitled "Multitask Allocation To Heterogeneous Participants in Mobile Crowd Sensing", considered a multi-task allocation problem with the heterogeneity of participants (different participants with different devices and tasks). To solve the above problem, the authors proposed a greedy discrete particle swarm optimization with genetic algorithm operation by using heuristic strategies and the random two-point mutation/crossover operations in the genetic algorithm. Aiming to examine the relationship between places of interest and the spatial patterns of mobility flows, the paper by L. Huang et al., entitled "Mining the Relationship between Spatial Mobility Patterns and POIs", modelled a network with each partitioned region as a node and connected between them as links weighted by the mobility flows. The community detection algorithm and logistic regression method were adopted to discover spatial mobility patterns and achieved the classify spatial communities featured by places of interest, respectively. To conserve the energy and wireless communication bandwidth for IoT network, Q. Xu et al. proposed a paper entitled "Cluster-Based Arithmetic Coding for Data

Provenance Compression in Wireless Sensor Networks" and presented a new cluster based arithmetic coding method which could encode and decode the provenance in an incremental manner with a higher compression rate. Also, the authors used a mathematical function of the WSN's size to derive the optimal clustering size, which was greatly useful for IoT-Fog computing environment.

5. User Privacy Preservation and Data Protection

Protecting the user and data privacy is an essential topic in the traditional cloud-based scenario. However, it contains more challenge issues when meeting the distributed IoT-Fog environment. To improve the participation of sensing users and the authenticity of sensing data in the fog computing, J. Xiong et al. proposed a paper entitled "Achieving Incentive, Security, and Scalable Privacy Protection in Mobile Crowdsensing Services" constructing a privacy-preserving data aggregation scheme. The authors used the differential privacy mechanism and homomorphic encryption for protecting the sensing data which could ensure the privacy of the sensing users. Moreover, the authors gave a new auction game theory based secure multi-party auction mechanism to solve the problem of prisoners' dilemma incurred in the sensing data transaction. Aiming to promote quality of service and guarantee security and fairness for wireless sensor network (WSN) in IoT, Z. Lv et al. proposed a paper entitled "A Rational Exchange Protocol under Asymmetric Information in Wireless Sensor Networks." In the paper, the authors presented an entropy-based incentive model and used the model to design an entropy-based rational exchange protocol, which satisfied the correctness, security, fairness, and robustness, respectively. To improve the efficacy of fully homomorphic encryption (FHE) for IoT usage, the paper by W.-T. Song et al., entitled "Privacy Protection of IoT Based on Fully Homomorphic Encryption" improved the bootstrapping technique in the FHE scheme to accelerate the computation. The authors optimized the parameter range, generalized their ciphertext modulus, and introduced SIMD homomorphic computation techniques into the new proposed method to improve the efficiency. To protect the data embedded in electronics, sensors, and software against side-channel attack (like DPA) in IoT scenario, the paper by S. Zhang and W. Zhong entitled "A New Type of Countermeasure against DPA in Multi-Sbox of Block Cipher" gave a new type of a countermeasure scheme against DPA in multi-Sbox of block cipher by converting the multi-Sbox into permutations, reused permutation to turn it into a special reusable Sbox, and made these inputs of permutations random by masking. The new method could successfully prevent the attacker from accurately aligning the power consumption and guarantee the data privacy of the IoT devices.

6. Conclusions

The authors in the special issue highlight both the promise and the challenges faced by this emerging field of security and privacy challenges in IoT and fog computing. Their manuscripts identify the further related research for security and privacy issues in IoT-Fog scenario. Hopefully, the special issue serves as a remarkable source for graduate students, education, professors, researchers, and whoever interested in updating their knowledge of fog computing, IoT, and security and privacy issues for future information services and systems.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Ximeng Liu
Yang Yang
Kim-Kwang Raymond Choo
Huaqun Wang

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 8959635, 12 pages
https://doi.org/10.1155/2018/8959635

Research Article

Achieving Incentive, Security, and Scalable Privacy Protection in Mobile Crowdsensing Services

Jinbo Xiong,[1,2] Rong Ma,[1] Lei Chen,[3] Youliang Tian,[2] Li Lin,[1] and Biao Jin[1]

[1]*College of Mathematics and Informatics, Fujian Normal University, Fuzhou, 350117, China*
[2]*Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang, 550025, China*
[3]*College of Engineering and Computing, Georgia Southern University, GA 30458, USA*

Correspondence should be addressed to Youliang Tian; youliangtian@163.com

Received 9 March 2018; Revised 4 June 2018; Accepted 31 July 2018; Published 12 August 2018

Academic Editor: Kim-Kwang Raymond Choo

Mobile crowdsensing as a novel service schema of the Internet of Things (IoT) provides an innovative way to implement ubiquitous social sensing. How to establish an effective mechanism to improve the participation of sensing users and the authenticity of sensing data, protect the users' data privacy, and prevent malicious users from providing false data are among the urgent problems in mobile crowdsensing services in IoT. These issues raise a gargantuan challenge hindering the further development of mobile crowdsensing. In order to tackle the above issues, in this paper, we propose a reliable hybrid incentive mechanism for enhancing crowdsensing participations by encouraging and stimulating sensing users with both reputation and service returns in mobile crowdsensing tasks. Moreover, we propose a privacy preserving data aggregation scheme, where the mediator and/or sensing users may not be fully trusted. In this scheme, differential privacy mechanism is utilized through allowing different sensing users to add noise data, then employing homomorphic encryption for protecting the sensing data, and finally uploading ciphertext to the mediator, who is able to obtain the collection of ciphertext of the sensing data without actual decryption. Even in the case of partial sensing data leakage, differential privacy mechanism can still ensure the security of the sensing user's privacy. Finally, we introduce a novel secure multiparty auction mechanism based on the auction game theory and secure multiparty computation, which effectively solves the problem of prisoners' dilemma incurred in the sensing data transaction between the service provider and mediator. Security analysis and performance evaluation demonstrate that the proposed scheme is secure and efficient.

1. Introduction

Crowdsensing, also known as crowdsourced sensing, mainly originates from the notion of crowdsourcing. In recent years, crowdsourcing was fused with mobile embedded sensors (such as acceleration sensors, digital compasses, GPS, microphones, and cameras) in an ordinary user mobile device into a powerful sensing unit, which consciously or unconsciously collaborates one another through the mobile Internet to form a mobile crowdsensing network [1, 2]. Compared with the traditionally fixed deployment sensing mode, the cloud-based mobile crowdsourcing has proven to be an attractive solution to provide data storage and share services for resource-limited mobile devices in a privacy preserving manner [3]. Mobile crowdsensing has been widely applied to environmental monitoring [4], intelligent traffic systems [5], social behavior analyses [6, 7], urban management [6], public security [8] and other fields with the advantages of low deployment costs, simple maintenance, and excellent scalability.

As a novel service schema of the IoT, mobile crowdsensing provides an innovative way to implement the ubiquitous social sensing [9]. Despite its innovation, the application of mobile crowdsensing is limited by the insufficient number of perceived participants and the low data quality [10], which may seriously affect the development of mobile crowdsensing for the following reasons. First, sensing users expect to receive actual incentives, rather than providing free sensing data. Without appropriate incentives, sensing users may not be interested at all in the task of data sensing due to the facts that mobile sensing devices have to consume resources, such as battery power, computation and storage resources, and data

traffic. Second, in a mobile crowdsensing network [9], the collected sensing data may contain a significant amount of sensitive and private information with the risk of private data leakage. Therefore, users anticipate that effective measures are taken to protect their privacy when sensing data is uploaded to the service providers [11]. Third, there may be malicious activities in the course of the data transactions between the mediator and the service provider, possibly leading to loss of profits.

In order to tackle the aforementioned problems, this paper proposes a hybrid incentive mechanism based on both reputation and service return to motivate users to participate in the sensing tasks. Meanwhile, a privacy preserving data aggregation scheme is proposed to allow sensing users to upload encrypted data to an incompletely trusted mediator, enabling the mediator to acquire the sensing data aggregation for each time interval without decrypting each ciphertext. In this scenario, the mediator cannot derive additional information from its background knowledge or expect statistical data. We further discuss the prisoners' dilemma problem of data transactions between the service provider and the mediator and propose a novel secure multiparty auction mechanism to solve the problem that the service provider lowers the price in data transaction. The main contributions of this paper are as follows:

(i) A reliable hybrid incentive mechanism is proposed based on both reputation and service returns. Additional reputation rewards are given to the sensing users who continue to provide high-quality sensing data, and the quality of the user sensing data reflects the quality of service (QoS) provided by the service provider. For the purpose of obtaining better QoS, sensing users are required to provide more accurate and authentic data. This ensures a sustainable growth of the number of sensing participants and the overall QoS of sensing data.

(ii) A privacy preserving data aggregation scheme is proposed based on differential privacy and homomorphic encryption to solve the problem of private data leakage, where the sensing users can securely contribute their encrypted data. The simulation results indicate that the proposed scheme is effective and efficient, even in the scenario where the mediator has the access to sensing user's auxiliary information while user privacy is still protected.

(iii) The problem of prisoners' dilemma in the transactions between the service provider and mediator is discussed, and a novel secure multiparty auction mechanism is designed based on both auction game theory and secure multiparty computation. In the process of transactions, the parties choose to process the actual value of the transaction data based on the goal of maximizing the profit and implement the privacy preserving for the transaction data.

(iv) The security analysis shows that the encryption algorithm scheme used in this paper is provably secure. The constructed privacy protection scheme meets the security objectives, and the performance analysis and simulation results indicate that the proposed scheme is effective and efficient.

The rest of this paper is organized as follows: Section 2 covers the related work of the incentive mechanism and privacy protection scheme in the mobile crowdsensing systems. The system model, adversary model, and the security requirements of the proposed schemes are described in Section 3. In Section 4, we first describe the hybrid incentive mechanism of mobile crowdsensing system and then introduce a privacy preserving data aggregation scheme by jointly integrating differential privacy and homomorphic encryption between sensing users and the mediator. Finally, the secure multiparty auction model between service providers and the mediator is presented. Sections 5 and 6 analyze the proposed scheme in the aspects of security and performance. Section 7 points out future research directions and summarizes the entire paper.

2. Related Work

2.1. Incentive Mechanism. Various incentives strategies and methods are available in mobile crowdsensing. Generally, in regard of the form of returns, it can be divided into monetary incentives and nonmonetary incentives [12]. Monetary incentives are mainly through reward payments to encourage the sensing users to participate in sensing tacks. Based on the sensing users' quotation of the sensing data, the system selects a subset of the sensing users with lower payment costs to complete the sensing task. Monetary incentives reward the sensing users' with money, a direct and currently the most common form of incentives. One of the most important incentive mechanisms is based on the auction game theory mechanism, including reverse auction, combined auction [13], multiattribute auction, full auction, two-way auction, and vickrey-clarke-groves (VCG) auction. In addition to the above, there are many other incentives methods, such as those based on the Stackelberg game model [14, 15]. The literature [14] proposed a Stackelberg game-based pricing mechanism to inspire core users to distribute videos to the multicast users via device-to-device (D2D) communication. Nonmonetary incentives include entertainment game incentives [16], social relations incentives [15, 17], and virtual integration incentives [18]. The entertainment game incentive [16] uses game entertainment and attractiveness to encourage the sensing users to complete the sensing task. Social relation incentives refer to a type of social networking relationship where the sensing users already exist or server platform is built, and the sensing users are motivated to maintain a sense of belonging in social relations [15]. The literature [17] exploits the coalition game by employing the user's social preference list to dynamically establish virtual communities. Virtual integral incentive [18] refers to the fact that the sensing users will receive the virtual integral from the sensing task. The virtual currency converts into a real currency or some other types of physical or virtual return, which encourages the sensing users to participate in a sensing task. In order to better motivate the sensing users, more recent research

works tend to integrate two or more types of incentives (hybrid incentive). The literatures [19] use reverse auction to select the winner to receive incentives in exchange of the sensing data, while motivating the noncompliant users to remain active in the system through virtual integration. The literature [15] uses the Stackelberg game model and establishes the endorsement relationship among the sensing users to motivate their participation. The literature [20] uses three types of incentives: reverse auction payment method based on game theory, social relation incentive based on the user's reputation level, and entertainment game incentive based on the user's psychological satisfaction. The hybrid incentive method provides satisfactory incentive strategies for various situations. In order to meet the psychological requirements of the sensing users and promote the participation of sensing task, it is encouraged to integrate various incentives, such as reward incentives, entertainment incentives, spiritual incentives, honor incentives, for an optimized solution.

2.2. Privacy Protection Scheme. The security of private information in mobile crowdsensing includes the privacy of the sensing users and the security of the service provider. The sensing users anticipate that their personal data privacy is preserved during the uploading to service provider [21, 22]. Since the sensing users and the mediator may not be completely trusted, the uploaded false data may potentially cause security problems to the service provider, and therefore countermeasures to malicious users and malicious attacks should be considered. Most privacy protection schemes assume that the mediator is fully trusted, and the sensing user takes privacy protection measures in the sensing task. Each user in the t-sensing task provides real-time sensing data and chooses an anonymity level. Data anonymity [23] can be used to add noise to the real data, such as k-anonymity and others. Each user uploads the private data and indicates data anonymity level to the mediator without knowing the privacy preferences of other users. The mediator collects all anonymously processed data sets and the anonymity levels from all sensing users, followed by trading them with the service provider. Wu [9] combined key distribution with trust management to construct a novel dynamic trust relationships-aware data privacy protection (DTRPP) mechanism for mobile crowdsensing. Zhang et al. [24] proposed a novel technique called match-then-decrypt, in which a matching phase is additionally introduced before the decryption phase. Rastogi and Nath [25] considered aggregating the sum statistical sum in the presence of an untrusted mediator, proving that the mediator cannot calculate a linear combination of user values other than the sum. However, this implicit security definition is not complete in a sensing task, and it requires the aggregator to interact with the participants in order to decrypt ciphertext for each time interval. Rieffel et al. [26] considered a specific application scenario in which the manager attempts to decrypt the statistical sum of a group of users on a regular basis without decrypting a single value. Ni et al. [27] proposed a fog-assisted mobile crowdsensing framework, enabling fog nodes to allocate tasks based on users' mobility for improving the accuracy of task assignment.

Table 1: Notations and descriptions.

Notations	Descriptions
g	a random generator
ε	a measure in differential privacy
Δ	the sensitivity
δ	the probability of adding noise
n	the number of sensing users
x_i^t	the sensing data of user i at t-th task
a_i	the data feature set
b_i	the class tag
v	the final transaction price
$\widehat{x}_i$	added noise sensing data
r_i	noise
$H(t)$	anti-collision hash function
sk_i	the sensing user's key
sk_0	the mediator's key
c_i	the ciphertext
V	the aggregation data
$\{sp_1, sp_2, \ldots, sp_m\}$	a list of service providers
r_0	a random number
$\{price_1, price_2, \ldots, price_m\}$	a list of the expected cost price
Z	the mediator

But their construction falls short in completely resisting against the collusion attack; in other words, users may collude with managers to decrypt the victim's data. Therefore, it is necessary to design a novel privacy protection scheme immune from collusion attacks.

In summary, limitations exist in the existing privacy protection schemes in mobile crowdsensing systems, mainly due to neglecting the consideration of mediator being not completely trusted. There exist security threats to the privacy protection methods used by sensing users (e.g., an anonymous antibackground knowledge attack). They also lack the privacy protection solutions against malicious users colluding with one another [28]. At the same time, the incentive mechanism of the mobile crowdsensing system is too simple to be effective. How to design an effective hybrid incentive mechanism to ensure high participation of sensing users in the sensing task while providing long-term high-quality data is of great importance.

3. Problem Description

This section first gives the notations and descriptions in Table 1 and then introduces our system model, followed by an adversary model, security requirements, and our design goals.

3.1. System Model. The mobile crowdsensing system consists of the following three main entities, as shown in Figure 1.

Crowdsensing Users. They are the participants who collect the sensing data using their personal mobile-aware devices (such as intelligent terminal equipment, wearable equipment, and automotive equipment). Crowdsensing users participate

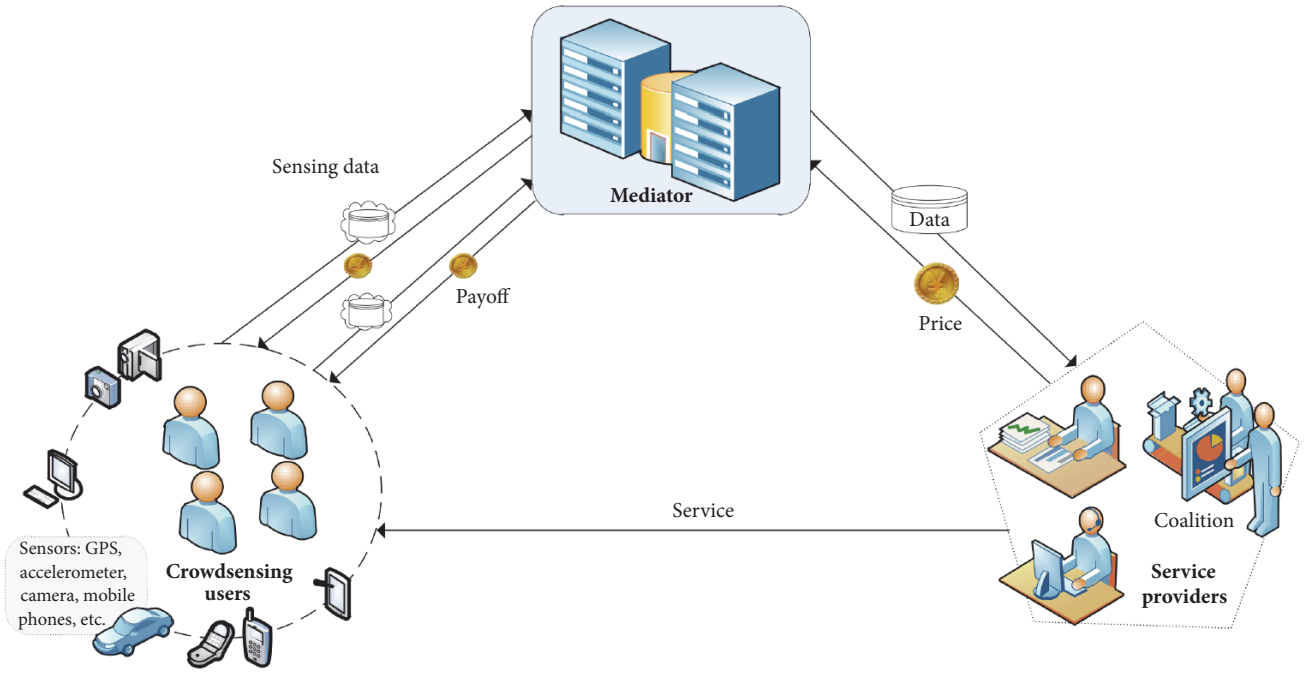


Figure 1: System model of mobile crowdsensing.

in sensing tasks and receive the corresponding maximal revenue via the mobile network by uploading the sensing data to the mediator. Continuous participation will help acquire additional reputation incentives. In order to implement privacy protection for the sensing data, crowdsensing users usually encrypt the data using the homomorphic encryption algorithm [22] with added noise, followed by uploading the ciphertext to the mediator. The quality of the sensing data provided by sensing user is reflected in the final quality of service (QoS) provided by the service provider: the higher the quality of the data provided by the sensing user, the higher the QoS be returned to the customers.

Service Providers. Due to the different requirements of service customers, service providers are responsible for participating in the final aggregated data transactions and providing various services to the customers. The aggregated sensing data received by the service provider is then used for machine learning, data visualization and other studies. A rational service provider aims to acquire higher value data from a mediator at a reasonable price. To reduce the cost of purchase, service provider may choose to share the data with shared other providers to average the total costs.

Mediators. They interact with both the sensing users and the service providers. Under the privacy protection mechanism though combining the differential privacy with homomorphic encryption, the mediator advertises the sensing task to the mobile crowdsensing users and adopts the hybrid incentive mechanism to attract more users to upload their encrypted sensing data. The mediator aggregates and sells the sensing data to the service providers to receive the corresponding rewards.

3.2. Adversary Model. Cryptographic Hash function is used for securing both the sensing user key and the mediation key of the homomorphic encryption algorithm. We assume that the Hash function is cryptologically secure. More specifically, it is assumed that the Hash functions used in our schemes are resistant against the weak collision attacks and the strong collision attacks [29]. Moreover, we assume that the mediator is a semitrusted data aggregator.

In our adversary model, we consider a strong attacker $\mathcal{A}$, who cannot only listen to all the communication data in our system model but also launches the following attacks:

(i) Attacker $\mathcal{A}$ may intercept one single sensing user's private data in the process of uploading data by consuming a substantial amount of cost. However, there are a large number of sensing users in the system, it will be extremely high cost for attacker $\mathcal{A}$ to intercept private data from each individual sensing user. Therefore, the attacker $\mathcal{A}$ may attempt to analyze the privacy of other sensing user's by using the user's private key which has already intercepted [30].

(ii) Attacker $\mathcal{A}$ may obtain private data by colluding with an incompletely trusted mediator. By doing this, attacker $\mathcal{A}$ will be able to access a large amount of sensing data stored at the mediator.

(iii) Attacker $\mathcal{A}$ can break a small number of sensing users, and attempt to obtain the other users' private keys and decrypt the ciphertext of the sensing data.

3.3. Security Requirements. The reliability and efficiency of the mobile crowdsensing system depends on the security of the communication system. Mobile crowdsensing systems are increasingly complex, interactive, and dynamic and therefore require advanced network technologies and complex security protocols to address potential security vulnerabilities. The design of a privacy protection mechanism in mobile crowdsensing systems needs serious and comprehensive consideration of the security of communications. In order to prevent attacker $\mathscr{A}$ from obtaining user's private data, it is desired to achieve the following security requirements:

(i) Even if $\mathscr{A}$ can listen to the communication data flow, he still could not obtain the private data from any sensing user.

(ii) Even if $\mathscr{A}$ can break into individual sensing user device, he still could not acquire other sensing users' private data.

(iii) Even if $\mathscr{A}$ can collude with the mediator to access the aggregation results of the sensing data, $\mathscr{A}$ still could not obtain the sensing user's personal private data.

(iv) Even if $\mathscr{A}$ can break into a small number of sensing users to access their private key, $\mathscr{A}$ still could not get the sensing user's original personal data.

3.4. Design Goals. With the above system model, adversary model, and security requirements, our goals are to propose a reliable hybrid incentive mechanism, an efficient privacy preserving data aggregation scheme, and a secure multiparty auction mechanism in mobile crowdsensing system. Specifically, the following objectives should be achieved:

(i) The proposed incentive mechanism should be reliable and effective. The sensing effect is closely related to the number of participants and the quality of data provided by the sensing users in mobile crowdsensing. Incentive mechanism must ensure that a sufficient number of sensing users have long-term involvement in sensing tasks and provide reliable data.

(ii) The proposed privacy preserving data aggregation scheme should meet the security requirements. As mentioned above, if the security and privacy issues are not considered in the mobile crowdsensing system, the privacy of the individual sensing user will be disclosed, hindering the further development and application of mobile crowdsensing system. Therefore, the proposed scheme must be able to meet the above security requirements.

(iii) The proposed aggregation scheme should be highly efficient in communication. While the sensing user and the mediator communicate via high-bandwidth, low-latency wired/wireless connections, it is essential to support a large number of sensing users simultaneously sending data to the mediator. The proposed scheme should consider the efficiency of communication, so that real-time sensing data can be sent to the intermediary in a timely manner.

(iv) The proposed data transaction mechanism should ensure the security of the service providers and the mediator. Service providers participate in data transaction process with rational choices to solve the problem of the prisoners' dilemma and achieve the goal of maximizing payoff without unnecessary disclosure of the parties' private information.

4. Construction

This section elaborates the details of the proposed reliable hybrid incentive mechanism for mobile crowdsensing, the privacy preserving data aggregation scheme combining differential privacy with homomorphic encryption, and the secure multiparty auction mechanism based on auction game theory.

4.1. Reliable Hybrid Incentive Mechanism. The data $X_i^t = \{(a_i, b_i)\}_{i=1}^L$ perceived by the sensing user i in the tth perceptual task is a tuples containing the data feature set $a_i \in \mathbb{R}^M$ and the class tag $b_i \in \mathbb{R}$, where L is the number of data tuples and M is the number of data attributes [31]. Feature set a_i consists of sensing data, such as GPS data in smart trip services and personalized recommended social network behavior data. Class tag b_i contains human inputs and is only available in supervised data analytics. After collecting sufficient sensing data, the service provider analyzes the sensing data and builds data-based services. For example, in intelligent transportation services, the use of mobile sensing equipment and urban traffic information collection analyses can provide consumers with more efficient and convenient travel route planning and auxiliary driving information support. We define the expression $U(X_i^t) = u(X_i^t) + R^t + Q$ to represent all of the final proceeds obtained by the sensing data X_i^t in the tth sensing task, which consists of the following three parts:

(i) *Participation income* $u(X_i^t)$. Sensing users choose to participate in a sensing task to obtain their participation income, which depends on the price paid by the service provider in the final transaction with the mediator. Assuming that N sensing users participate in the same sensing task, the transaction price of the final transaction service provider is v; then each participant's revenue in the tth sensing task is $u(X_i^t) = v/N$.

(ii) *Reputation points* R^t. Sensing user participating in the tth sensing task will earn their reputation points. The user's reputation points with continued participation in the sensing task can be accumulated to encourage long-term participation in the sensing task. However, the reputation points will be cleared when the sensing task is interrupted or revoked.

(iii) *Feedback service quality* Q. The quality of the sensing data will reflect the QoS provided by the final service provider to the consumers. Therefore, the system encourages the sensing users to provide data with

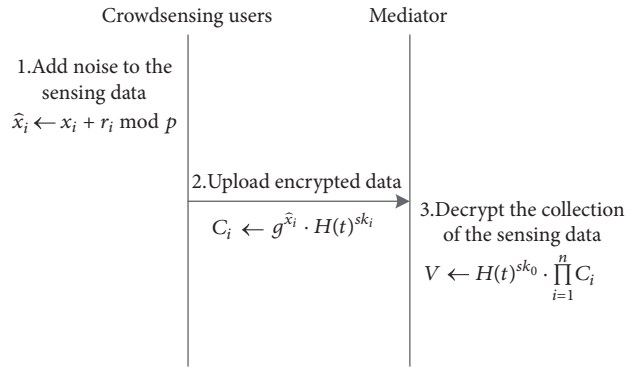


Figure 2: The interactive process of privacy protection.

high level of authenticity to obtain better quality service.

The sensing users can obtain three kinds of incentive income in the hybrid incentive mechanism: reward payment, reputation integral incentive, and service quality incentive. In order to meet the sensing users' psychological expectation and material requirements, the system promotes long-term access to the perceptual task with highly accurate sensing data.

4.2. Privacy Preserving Data Aggregation Scheme. The privacy preserving data aggregation scheme allows the sensing users to upload the encrypted data to an incompletely trusted mediator. The mediator obtains the aggregation of sensing data for each time interval without decrypting each ciphertext. The interaction between the sensing user and mediator is illustrated in Figure 2. The marriage of the differential privacy and homomorphic encryption in privacy preserving data aggregation scheme includes the following three phases: adding noise data, data encryption, and data decryption.

4.2.1. Adding Noise Data. In a mobile crowdsensing system, it is assumed that the sensing user is capable of arbitrarily adding noise to sensing data based on privacy preferences before uploading. This may lead to a serious deviation between the final data aggregation statistics and the actual results, thereby significantly reducing the availability of data collection. For this reason, we introduce a distributed differential privacy protection mechanism [32], where each sensing user only needs to add a small amount of noise with randomness, which still ensures that data privacy is well preserved. The specific description of this phase is shown in Algorithm 1.

4.2.2. Data Encryption. In the proposed model, it is assumed that the mediator is not fully trusted. The sensing user uploads the ciphertext encrypted though the homomorphic encryption [22] data to the mediator, who can only obtain the data aggregation result without knowing the individual private information of any sensing user. In order to counter against background knowledge attack, data encryption is discussed using data collection as an example. In the sensing

input: $\alpha = \exp(\varepsilon/\Delta)$, $\beta = (1/r^n)\log(1/\delta)$, $X = \{x_1, \ldots, x_n\}$
output: $\widehat{X} = \{\widehat{x}_1, \ldots, \widehat{x}_n\}$
(1) **for** each $i \in [n]$ **do**;
(2) Sample noise r_i according to the following;
(3)

$$r_i \longleftarrow \begin{cases} Geom(\alpha) & \text{with probability } \beta \\ 0 & \text{with probability } 1-\beta \end{cases}$$

(4) Randomize data by computing $\widehat{x}_i \longleftarrow x_i + r_i$ **mod** p;
return $\widehat{X}$

Algorithm 1: Adding noise data.

input: $\widehat{X} = \{\widehat{x}_1, \ldots, \widehat{x}_n\}$, $SK = \{sk_1, \ldots, sk_n\}$
output: $C_i = \{c_1, \ldots, c_n\}$
(1) **for** each $i \in [n]$ **do**;
(2) $M_{i,t} = H(t)^{sk_i}$;
(3) Encrypted message by computing $c_i \longleftarrow g^{\widehat{x}_i} \cdot M_{i,t}$;
return C_i

Algorithm 2: Data encryption.

task, the random sequence generator assigns the encryption keys $sk_1, \ldots, sk_n$ to the different sensing users and the decryption key sk_0 to the mediator, which satisfies $\sum_{i=0}^{n} sk_i = 0$. Each sensing user calculates the function $M_{i,t} = H(t)^{sk_i}$, $i \in [n]$ according to its encryption key and the hash function $H(x)$. Then the mediator calculates function $M_{0,t} = H(t)^{sk_0}$. Since $\sum_{i=0}^{n} sk_i = 0$, therefore $\prod_{i=0}^{n} M_{i,t} = 1$. Using this property, the homomorphic addition encryption of the perceptual data can be implemented. The users encrypt the sensing data using the respective encryption keys to obtain the ciphertext c_i:

$$c_i \longleftarrow g^{\widehat{x}_i} \cdot H(t)^{sk_i}. \tag{1}$$

The specific description of the data encryption phase is shown in Algorithm 2.

4.2.3. Data Decryption. The mediator receives the user's uploaded ciphertext and decrypts it with the corresponding decryption key sk_0 by calculating

$$V \longleftarrow H(t)^{sk_0} \cdot \prod_{i=1}^{n} C_i. \tag{2}$$

The specific description of the data decryption phase is shown in Algorithm 3.

In this privacy preserving data aggregation scheme, the mediator cannot decrypt the private data of a particular sensing user even with potential assistance information. In addition to knowing whether a particular user participates in a perceptual task, the mediator only obtains the statistical results of the data aggregation.

input: $sk_0, C_i = \{c_1, \ldots, c_n\}$
output: V
(1) Aggregation by computing $\prod_{i=1}^{n} C_i = c_1 \times c_2 \times, ..., \times c_n$;
(2) $M_{0,t} = H(t)^{sk_0}$;
(3) Decrypted message by computing $V \longleftarrow H(t)^{sk_0} \cdot \prod_{i=1}^{n} C_i$;
(4) $V = g^{\sum_{i=1}^{n} \tilde{x}_i}$;
return V

Algorithm 3: Data decryption.

4.3. Secure Multiparty Auction Mechanism. In addition to relying on sensing users to actively provide high-quality data, mobile crowdsensing system also depends on the mutual trust transactions between the service providers and mediator. However, in the actual transaction process, due to the unequal information accessibility, the data transaction process can be seen as two incomplete information static games [33]. In order to maximize their own profits, service providers tend to purchase data at a lower price, regardless of the quality of the data provided by the mediator. For the mediator, regardless of the price provided by the service provider, providing low-quality data may lead to more revenue. Based on this, the Nash equilibrium solution of the game for the service provider and mediator is <low quality data, low price>, which falls into the prisoner's dilemma. Obviously, this is the worst result, which will lead to lower yields on the mediator, resulting in less revenue for the sensing users. In the proposed privacy preserving data aggregation scheme, since the sensing data is uploaded to the mediator in ciphertext, data accuracy cannot be evaluated. In other words, the real price/value of the data is unknown. This is even more detrimental to data transactions between service providers and mediator.

4.3.1. Secure Multiparty Computation. Using the secure multiparty computation, each service provider can enter the expected price of the data provided by the mediator, respectively. The computational functions $f(x_1), f(x_2), \ldots, f(x_n)$ negotiate the true price of the obtained data. At the end of the protocol, each service provider cannot receive any other information except the value of $f(x_i)$.

Suppose there are more than three service providers $sp_1, sp_2, \ldots, sp_m$ and a mediator Z to provide data for the expected cost price $price_1, price_2, \ldots, price_m$, and $price_0$, respectively. Since the service providers are mutual competitors, it is safe to assume that there is no collusion among them.

To calculate the starting price $price = (\sum_{i=0}^{m} price_i)/(m + 1)$, let the mediator Z generate a random number r_0 and passes $r_0 + price_0$ to service provider sp_1. Then sp_1 calculates $r_0 + \sum_{i=0}^{1} price_i$ and passes it to sp_2, and this process continues. The mediator Z finally obtains $r_0 + \sum_{i=0}^{m} price_i$ and subtracts r_0, where the mediator and its service providers obtain the expected cost of the sum of the price. The mediator Z divides the total price with the number of participants in the calculation to obtain the starting price and publishes it to each service provider.

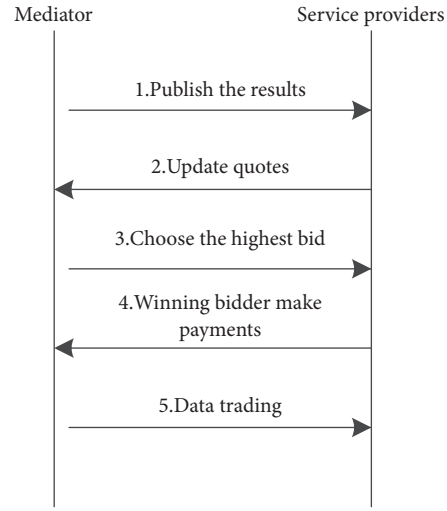


Figure 3: The interactive process of auction mechanism.

4.3.2. Auction Mechanism. Assuming the rationality that the service providers participate in the auction, they choose the calculation strategy for maximized benefit. The interaction between the mediator and service providers is shown in Figure 3.

The mediator first announces the auction starting price from the last step of the secure multiparty computation and, then, the service providers participating in the auction update their quotations according to the starting price adjustment strategy. To close the auction, the mediator selects the service provider with the highest offer as the winning bidder from multiple quotes, followed by data trading to complete this process.

5. Security Analysis

As addressed earlier, we consider scenario with the existence of a strong attacker in the system, who cannot only monitor the communication channel in the system but also collude with the mediator and gain access to the information stored in the mediator. Additionally, attacker $\mathcal{A}$ can break into a small number of sensing users and obtain these users' private keys. Since this paper primarily considers protecting data privacy for the sensing users, attackers tampering with the messages are beyond the scope this particular research, although this

can be prevented by enforcing strong authentication. In this section, we analyze in detail how the proposed schemes are resistant against a variety of attacks by the attacker.

(i) Our scheme can guarantee that the ciphertext submitted by the sensing users will not disclose their personal or private data. As mentioned in the security model, the attacker $\mathcal{A}$ may monitor the data transmitted to the mediator during the user upload phase. Our privacy preserving data aggregation scheme ensures that even if the attacker $\mathcal{A}$ listens to all ciphertext for all sensing users in each time interval, he still cannot derive any plaintext information from these ciphertext, and therefore data privacy is guaranteed. The encryption scheme used in this research is a one-way trap function, based on the Diffie-Hellman hypothesis that the advantage of a polynomial time attacker $\mathcal{A}$ is negligible [30]. At this point, our scheme is to specify what is IND-CPA security.

(ii) In our system, we consider that the attacker $\mathcal{A}$ may break into individual user mobile crowdsensing devices and obtain their private keys. In this case, a user's uploaded ciphertext data will undoubtedly be completely exposed. However, due to the large number of users in the system, the attacker $\mathcal{A}$ may not want to use this costly approach to break into multiple individual user devices. Alternatively, attacker $\mathcal{A}$ may expect to be able to use the private key that he has obtained to analyze the private keys of other users. However, this attack would not be successful, because each user's private key is randomly generated. As a result, data privacy for noncompromised users can still be guaranteed.

(iii) The user's private data will not be exposed at the mediator. At any time interval, the mediator cannot decrypt any ciphertext to obtain the corresponding plaintext information x_i even after collecting the ciphertext from all the sensing users. Therefore, even if the attacker $\mathcal{A}$ colludes with the mediator, $\mathcal{A}$ can only have access to the users' ciphertext. On the other hand, the mediator does not have the corresponding decryption key of any ciphertext and, as a result, $\mathcal{A}$ cannot either obtain the corresponding decryption key for decryption. Moreover, the aggregated ciphertext can only be decrypted with the mediator's random key sk_0, and $\mathcal{A}$ neither knows any users' keys nor knows the mediator's random key sk_0. Therefore, $\mathcal{A}$ cannot break the aggregated ciphertext. Even if the attacker $\mathcal{A}$ obtains the mediator's random key sk_0, the result of the decryption $X + R$ is the aggregated result of both ciphertext and added noise. Based on the above analysis, the user's private data and data aggregation results will not be exposed to the strong attacker $\mathcal{A}$.

(iv) The original data from the sensing user attacked by $\mathcal{A}$ will not be compromised. In the proposed system, we consider that the attacker $\mathcal{A}$ has the ability to obtain the sensing user's private key, and subsequently decrypts the user's uploaded ciphertext. As a countermeasure, we introduce the geometric distribution differential privacy for adding noise to the original sensing data. In this case, even if the ciphertext uploaded by the compromised user is completely leaked, the decrypted data is still different from the original data due to the added noise for protecting original data.

6. Performance Analysis and Evaluation

6.1. Complexity Analysis. This paper mainly focuses on the complexity analysis of the algorithm based on the computation costs at the sensing users, mediator, and service providers. In each time interval, the sensing user's computational cost is mainly due to the addition of differential noise and the use of random keys to encrypt the sensing data. Let T_m be the time of modulo multiplication operation and T_e be the time of modulo exponentiation operation. In our scheme, the sensing user generates a ciphertext using formula $c_i \longleftarrow g^{\tilde{x}_i} \cdot H(t)^{sk_i}$. Therefore, the sensing user needs to calculate the modulo exponential operation twice and the modulo multiplication operation once, i.e., $2T_e + T_m$. Also in each time interval, each sensing user needs to use the distributed geometric distribution to add noise. The time to add noise is however negligible compared to the modulo operation. The computational cost of the mediator is mainly reflected in the collection of the decryption of aggregated data, with decryption formula $V \longleftarrow H(t)^{sk_0} \cdot \prod_{i=1}^{n} C_i$. Therefore, the mediator needs to calculate one modulo multiplication operation and one modulo exponential operation, i.e., $T_e + T_m$. The computational cost of the service provider incurs due to the participation in the secure multiparty computation with the time complexity $O(1)$. In our scheme, the total communication overhead is $O(n)$, and its individual user's communication overhead is $O(1)$.

6.2. Performance Evaluation. In this section, we evaluate the performance of the privacy protection scheme proposed in this paper, mainly focusing on the two aspects of calculation time cost and statistical error. In terms of calculating the time cost, we consider the effect of different variables on the calculation time in the three phases of encryption, aggregation, and decryption. In terms of statistical error, we compare the geometric distribution differential privacy with the addition of the random noise. We conducted a set of simulation experiments using C++ on a Linux computer with the following hardware and configurations: Intel Xeon E5-2650 v3 @ 2.30 GHz CPU, 8 GB RAM, Ubuntu 16.04LTS Operating System, and GCC 5.4.0 Compiler.

The experiment tests the calculation running time in data encryption phase, data aggregation phase, and data decryption phase, respectively. Experiment runs 1000 times to receive the average as the experimental results.

(1) Data encryption phase. The experiment tests the running time using symmetric encryption with user key for different sizes of sensing data, which selects 7 randomly generated different sizes of sensing data from 32 bits to 2048

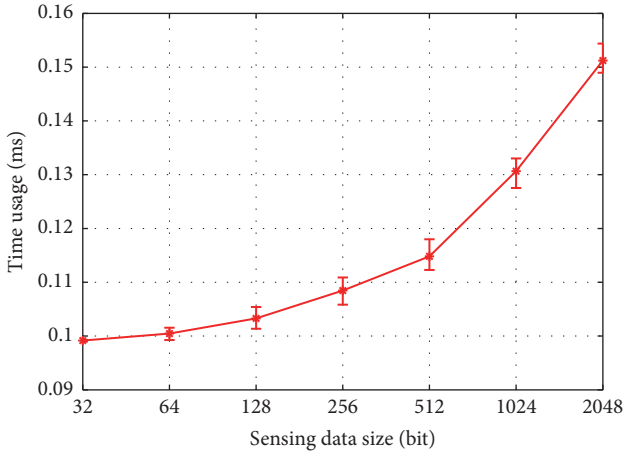


Figure 4: Running time of data encryption.

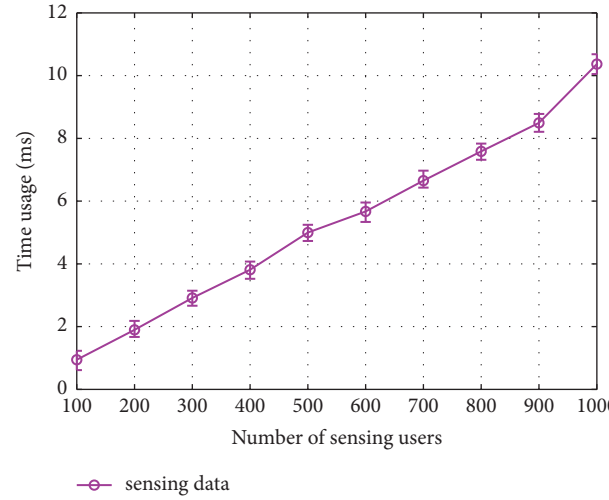


Figure 5: Running time of data aggregation.

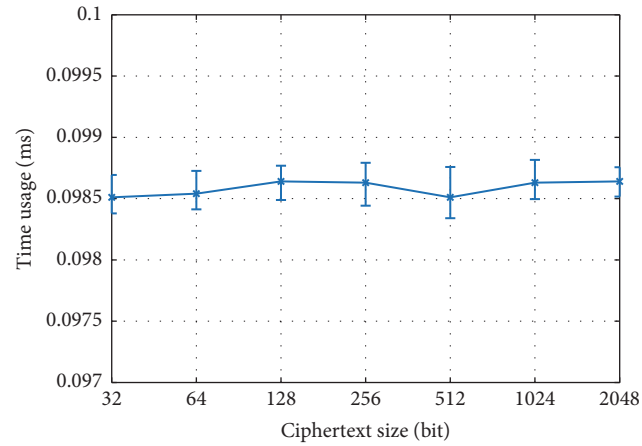


Figure 6: Running time of data decryption.

bits: 32 bits, 64 bits, 128 bits, 256 bits, 512 bits, 1024 bits, and 2048 bits, as shown in Figure 4.

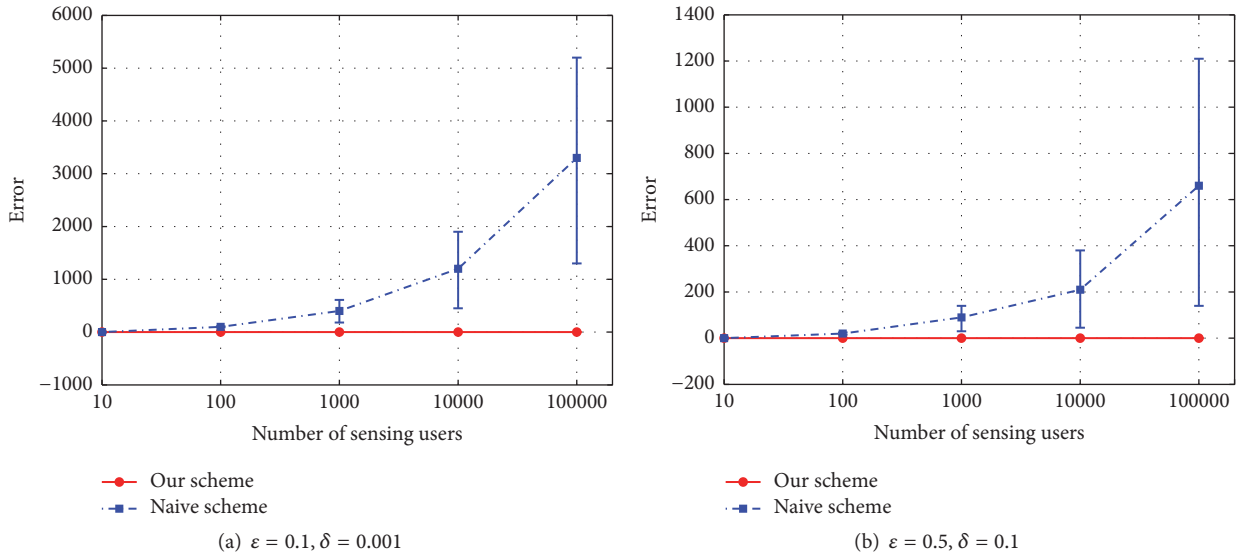
The data encryption phase mainly considers the relationship between the size of the plaintext and the time consumed for encryption. The plaintext uses a binary string encoding, which is expressed as the length of the string in bits. The plaintext content selection uses randomly generated values that are uniformly distributed. As the size of the plaintext increases, the time required for encryption increases significantly, as shown in Figure 4. As the data length increases, the encryption running time also increases. When the data length reaches 512 bits, the data encryption time is clocked at only 0.112 milliseconds. It is also worth noting that both the ciphertext and the parameter selection within the encryption process affect the output of the encryption. We use multiple sets of generators to randomly generate ciphertexts by repeating experiments to observe their effects in time. Small localized fluctuations, not affecting the overall trend, can be found in Figure 4. Therefore, it can be considered that, in a certain range of errors, the time required for the encryption process is exponential to the size of the plaintext as expected.

(2) Data aggregation phase. The experiment tests the running time of different numbers of sensing data aggregations. Experiments were performed in turn from 100 to 1000 different numbers of the sensing users, i.e., 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000, as shown in Figure 5. Single data block with a size of 1024 bits is used for 5 different sets of sensing data for polymerization time tests.

We expect that a large influx of sensing users under the current system architecture will not cause a sudden change in system stability, where the number of participating users has a linear growth with the privacy preserving data aggregation time cost. We observe the time cost of the system under a single privacy preserving data aggregation with different users. As shown in Figure 5, the amount of time required for privacy preserving data aggregation increases almost linearly with the number of sensing users. When this number reaches 1000, the time of aggregating data is clocked at only about 10 milliseconds. Considering that the content submitted by the sensing users and the aggregation of the internal operations will affect the efficiency of aggregation, our scheme is designed to use multiple sets of the same number of values. By repeating the experiment, it is discovered that the time cost required for the aggregation aligns to the overall trend of the user's participation. Therefore, in a reasonable range of error, the privacy preserving data aggregation is considered stable without causing any sudden change in system overhead as the number of sensing users increases rapidly.

(3) Data decryption phase. This experiment tests the running time of data decryption of the encrypted sensing data of different content. Experiments randomly select 5 groups of encrypted sensing data selected from 32 bits to 2048 bits: 32 bits, 64 bits, 128 bits, 256 bits, 512 bits, 1024 bits, and 2048 bits, as shown in Figure 6.

It is shown that the impact of decryption time variations can be ignored. Specifically, the decryption is only concerned with the results of the previous aggregation. Most of the time spent in the decryption phase is on decrypting aggregated data using decryption key. As shown in Figure 6, there is no



(a) $\varepsilon = 0.1, \delta = 0.001$ (b) $\varepsilon = 0.5, \delta = 0.1$

Figure 7: Error comparison.

significant change in the time cost of decryption as the data size increases. In the vicinity of 0.0985 milliseconds, the trend of the curve is consistent with the computation complexity of the scheme. In other words, the sensing user's overall change will not cause any significant time delay. At the same time, we consider decrypting the results of the data aggregation, and decryption of the internal parameters will jointly affect the decryption efficiency. It is designed to use the overall performance of multiple groups of users and the aggregation of multiple user data to observe the overall trend of the fluctuations. It can be seen from Figure 6 that its fluctuation is confined within a small and permittable range and therefore the data decryption process can be considered stable.

(4) Error comparison of geometric distribution with noise reduction and random noise reduction.

In Figure 7, the x-axis represents the number of sensing users, and the y-axis represents the mean value of the error (absolute). The naive scheme is that each sensing user adds independent geometric noise to his sensing data and uploads the perturbed data to the mediator. It is clear that this approach will cause a significant discrepancy between the aggregated data and the original values. Our scheme uses the distributed geometric plus noise to simulate the discrete Laplacian de-noising, which solves this problem effectively.

7. Conclusion

With the continuous service expansion and extension of mobile crowdsensing systems, privacy protection schemes and incentive mechanisms are highly demanded for their adoptions to the dynamic heterogeneous sensing systems. This paper proposed a reliable hybrid incentive mechanism based on both reputation and service return to inspire more high-quality sensing users to participate in mobile crowdsensing tasks. We constructed a privacy preserving data aggregation scheme based on the assumption that the mediator and/or sensing users as incompletely credible. Homomorphic encryption is used to allow the mediator to decrypt the collection of the sensing data from multiple ciphertext that are encrypted using different sensing user keys. The proposed privacy preserving data aggregation scheme also utilizes the differential privacy mechanism by adding noise to the sensing data, ensuring the security of the remaining data of the sensing users even with the exposure of partial data. We discussed the security problem of data transaction between service provider and mediator and put forwarded a novel secure multiparty auction mechanism based on both the auction game theory and secure multiparty computation to solve the problem of prisoner's dilemma at the service providers. Finally, we proved the security and efficiency of the proposed schemes through security analyses and simulation experiments. Future study will focus on how to improve calculation efficiency and the accuracy-privacy tradeoff for mobile crowdsensing.

Data Availability

Our data is obtained through simulation experiments, and we can provide it if needed.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported in part by the Natural Science Foundation of China (61772008, 61402109, 61502489, 61502248, and 61502102) and Guizhou Provincial Key Laboratory of Public Big Data Research Fund (2017BDKFJJ028).

References

[1] Z. Xu, L. Mei, K. R. Choo et al., "Mobile crowd sensing of human-like intelligence using social sensors: a survey," *Neurocomputing*, vol. 279, pp. 3–10, 2018.

[2] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 29–35, 2014.

[3] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute-based multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, 2017.

[4] S. Kim, C. Robson, T. Zimmerman, J. Pierce, and E. M. Haber, "Creek watch: pairing usefulness and usability for successful citizen science," in *Proceedings of the 29th Annual CHI Conference on Human Factors in Computing Systems (CHI '11)*, pp. 2125–2134, ACM, Vancouver, Canada, May 2011.

[5] J. Ni, A. Zhang, X. Lin, and X. S. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 146–152, 2017.

[6] Z. Xu, H. Zhang, C. Hu et al., "Building knowledge base of urban emergency events based on crowdsourcing of social media," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 15, pp. 4038–4052, 2016.

[7] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: efficient policy-hiding attribute-based access control," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2130–2145, 2018.

[8] J. Weppner and P. Lukowicz, "Bluetooth based collaborative crowd density estimation with mobile phones," in *Proceedings of the 11th IEEE International Conference on Pervasive Computing and Communications (PerCom '13)*, pp. 193–200, San Diego, Calif, USA, March 2013.

[9] D. Wu, S. Si, S. Wu, and R. Wang, "Dynamic trust relationships aware data privacy protection in mobile crowd-sensing," *IEEE Internet of Things Journal*, 2017.

[10] Y. Wen, J. Shi, Q. Zhang et al., "Quality-driven auction-based incentive mechanism for mobile crowd sensing," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4203–4214, 2015.

[11] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, 2018.

[12] T. Luo, S. S. Kanhere, J. Huang, S. K. Das, and F. Wu, "Sustainable incentives for mobile crowdsensing: Auctions, lotteries, and trust and reputation systems," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 68–74, 2017.

[13] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "TRAC: truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *Proceedings of the 33rd IEEE Conference on Computer Communications (IEEE INFOCOM '14)*, pp. 1231–1239, May 2014.

[14] D. Wu, J. Yan, H. Wang, D. Wu, and R. Wang, "Social Attribute Aware Incentive Mechanism for Device-to-Device Video Distribution," *IEEE Transactions on Multimedia*, vol. 19, no. 8, pp. 1908–1920, 2017.

[15] T. Luo, S. S. Kanhere, and H. Tan, "SEW-ing a Simple Endorsement Web to incentivize trustworthy participatory sensing," in *Proceedings of the Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON '14)*, pp. 636–644, Singapore, Singapore, June 2014.

[16] Y. Ueyama, M. Tamai, Y. Arakawa, and K. Yasumoto, "Gamification-based incentive mechanism for participatory sensing," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS '14)*, pp. 98–103, Budapest, Hungary, March 2014.

[17] D. Wu, Q. Liu, H. Wang, D. Wu, and R. Wang, "Socially aware energy-efficient mobile edge collaboration for video distribution," *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2197–2209, 2017.

[18] T. Yu, Z. Zhou, D. Zhang, X. Wang, Y. Liu, and S. Lu, "INDAPSON: an incentive data plan sharing system based on self-organizing network," in *Proceedings of the IEEE INFOCOM, IEEE Conference on Computer Communications*, pp. 1545–1553, Toronto, ON, Canada, April 2014.

[19] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining 'gamification'," in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (MindTrek '11)*, pp. 9–15, ACM, Tampere, Finland, September 2011.

[20] Y. Wang, X. Jia, Q. Jin, and J. Ma, "QuaCentive: a quality-aware incentive mechanism in mobile crowdsourced sensing (MCS)," *The Journal of Supercomputing*, vol. 72, no. 8, pp. 2924–2941, 2016.

[21] R. Ma, J. Xiong, M. Lin, Z. Yao, H. Lin, and A. Ye, "Privacy protection-oriented mobile crowdsensing analysis based on game theory," in *Proceedings of the IEEE Trustcom/BigDataSE/ICESS*, pp. 990–995, Sydney, Australia, August 2017.

[22] X. Liu, R. Deng, K.-K. R. Choo, Y. Yang, and H. Pang, "Privacy-preserving outsourced calculation toolkit in the cloud," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[23] H. Wang, D. He, Y. Sun, N. Kumar, and K.-K. R. Choo, "PAT: a precise reward scheme achieving anonymity and traceability for crowdcomputing in public clouds," *Future Generation Computer Systems*, vol. 79, pp. 262–270, 2018.

[24] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Information Sciences*, vol. 379, pp. 42–61, 2017.

[25] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proceedings of the SIGMOD International Conference on Management of Data (SIGMOD '10)*, pp. 735–746, ACM, Indianapolis, IN, USA, June 2010.

[26] E. Rieffel, J. Biehl, W. van Melle et al., "Secured histories: computing group statistics on encrypted data while preserving individual privacy," 2010.

[27] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. S. Shen, "Providing task allocation and secure deduplication for mobile crowdsensing via fog computing," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[28] Y. Yang, X. Liu, R. H. Deng, and Y. Li, "Lightweight sharable and traceable secure mobile health system," *IEEE Transactions on Dependable and Secure Computing*, 2017.

[29] J. Xiong, Y. Zhang, X. Li et al., "Rse-pow: a role symmetric encryption pow scheme with authorized deduplication for multimedia data," *Mobile Networks and Applications*, vol. 23, no. 3, pp. 650–663, 2018.

[30] L. Chen, R. Lu, and Z. Cao, "PDAFT: a privacy-preserving data aggregation scheme with fault tolerance for smart grid communications," *Peer-to-Peer Networking and Applications*, vol. 8, no. 6, pp. 1122–1132, 2014.

[31] M. A. Alsheikh, Y. Jiao, D. Niyato, P. Wang, D. Leong, and Z. Han, "The accuracy-privacy trade-off of mobile crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 132–139, 2017.

[32] E. Shi, T. H. H. Chan, and E. Rieffel, "Privacy-preserving aggregation of time-series data," in *Proceedings of the Annual Network And Distributed System Security Symposium*, 2011.

[33] S. Shen, G. Yue, Q. Cao, and F. Yu, "A survey of game theory in wireless sensor networks security," *Journal of Networks*, vol. 6, no. 3, pp. 521–532, 2011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5702068, 10 pages
https://doi.org/10.1155/2018/5702068

Research Article

An Anonymous Multireceiver with Online/Offline Identity-Based Encryption

Qihua Wang,[1,2] **Fagen Li**,[1] **and Huaqun Wang**[3]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2]*School of Medical Information Engineering, Jining Medical University, Rizhao 272067, China*
[3]*School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*

Correspondence should be addressed to Fagen Li; fagenli@uestc.edu.cn

Received 2 March 2018; Revised 21 May 2018; Accepted 13 June 2018; Published 12 August 2018

Academic Editor: Mohammad Shojafar

Anonymous multireceiver encryption scheme can not only protect the privacy of the receiver but also ensure the security of message. However, the computational cost of this scheme is very large. It is not suitable for the sender which has limited resources, such as mobile devices and sensor nodes. In this work, an anonymous multireceiver online/offline identity-based encryption is proposed based on offline/online and identity-based encryption (IBE). In identity-based encryption scheme, the sender can encrypt the message using the unique information of the user (such as identity number or e-mail address) as its public key. The receiver obtains the private key from a central authority. For mobile device with limited resource, the online/offline encryption scheme can reduce the computational cost. Compared to the previous anonymous multireceiver schemes, the proposed scheme can efficiently encrypt message with offline/online method and ensure the anonymity of receivers. The analysis results also show that our scheme is efficient in terms of computational cost by comparing to the previous works.

1. Introduction

Multireceiver communication [1] is a crucial way to send and receive message. It can effectively solve the problem of key management and data sending. Multireceiver encryption also is converted to broadcast encryption [2] in certain extent. In multireceiver encryption strategy, the sender/encryptor can select any receiver. In broadcast encryption scheme, the sender/encryptor sends message to a group of users; only the legal uses can decrypt the ciphertext. This scheme is widely used in pay-TV applications, the distribution of copyright materials, etc.

In [3], the authors use the idea of identity-based encryption (IBE for short) for reference. The identity information of the receiver is converted to a public key. The receiver's private key which is distributed by a Key Generator Center (KGC) is connected with the identity information. The receiver can use the private key to decrypt the ciphertext. In [4], Lu and Hu addressed a pairing based multireceiver encryption scheme which can broadcast sensitive information in a complex environment, but it did not protect the privacy of the users. That is to say, this scheme cannot reach the anonymity of the users. A secure and efficient anonymous multireceiver IBE scheme was proposed in [5]. Based on [5], an anonymous multireceiver IBE scheme was improved by Wang et al. [6]. The proposed method cannot truly attain the anonymity of the receiver's information, and the receiver's privacy was not protected. In [5, 6], a legal receiver can easily verify whether a specific user is one of the legal receiver or not using only two bilinear pairing computational costs. Li et al. [7] analyzed the security vulnerabilities that exist in [6], but they did not give specific solutions. In order to deal with the privacy of the legal receivers, a really anonymous multireceiver IBE scheme was proposed in [8]. In the proposed scheme, all users can receive the broadcast ciphertext of the sender/encryptor, but only the receiver which was selected by the sender/encryptor can decrypt the ciphertext information. No one except the sender knows who the receiver is. The key issue of this scheme is how to design encryption scheme by using Lagrange interpolation function.

Chien [9] proposed an improved scheme which can achieve the receiver's anonymity and enhance the security of the message. However, in encryption phase, this scheme requires a number of bilinear pairing operations which is proportional to the number of receivers. He et al. [10] addressed an efficient certificateless anonymous multireceiver encryption scheme according to elliptic curve cryptography for devices with limited resources. The anonymous multirecipient IBE scheme can be used in pay per-view TV channel and sensitive program order. The receiver does not want any other receivers to know his or her identity information.

In IBE, the computational cost of multiplication and exponentiation operations in groups is larger. It takes much more time and battery power to execute exponential operations for the receiver with limited energy such as mobile phones or mobile devices. In IBE, data encryption needs bilinear pairing operation which can increase the runtime of encryption because the computational cost of bilinear pairing operation is very large. It is difficult to complete the encryption task in a short time for lightweight devices such as wireless sensor nodes or smart cards. Moreover, the anonymous multireceiver IBE takes more time compared to standard IBE.

One challenge in the anonymous multireceiver IBE is that the added functionality may increase the computation cost compared to standard public key cryptography. Online/offline technology can effectively reduce encryption time. The first online/offline IBE scheme was proposed by Guo et al. [11]. The scheme divided the encryption process into two stages: online stage and offline stage. In offline stage, the complex operation is preprocessed. In online encryption stage, the sender performs simple operations and generates the ciphertext. The online phase would be very fast. Moreover, it requires little computational cost in this phase. The online/offline encryption strategy is more suitable for lightweight equipment such as wireless sensor nodes or smart cards [12, 13]. Online/offline identity-based encryption scheme has attracted extensive attention, and series of research results have emerged [14–16]. Recently online/offline technology is also used in attribute-based encryption [17, 18]. However, previous literatures did not apply the online/offline scheme to the anonymous multireceiver IBE.

In this article, we concentrate on multireceiver IBE scheme that takes into consideration online/offline encryption. The offline information cannot be reused in previous work. In our proposed scheme, a few operations can be done in offline phase. The offline ciphertext which is computed in offline phase can be reused for the same receiver sets. This method can reduce the computation cost for the senders when they encrypt the message to the same receive sets.

Our motivating application for the work in this way is mobile device with limited resources. The preparation computation can be done while the mobile device is plugged into a power supply, and then when it is on the move without plugging, it performs the encryption operations with little computational cost.

The structure of this work is organized as follows. Section 2 reviews the cryptographic backgrounds and Section 3 describes an anonymous multireceiver online/offline identity-based encryption. The security proof and performance analysis are given in Section 4. Finally, Section 5 is the conclusions of this work.

2. Preliminary

Some fundamental backgrounds related to this work are given in this section.

2.1. Lagrange Interpolation Theorem. Fitting the curve through these points $(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)$ can be expressed as follows [6]:

$$f(\mathrm{x}) = \sum_{i=1}^{t} F_i(\mathrm{x}) = \sum_{i=0}^{t} a_i x^i \tag{1}$$

where for each i

$$\begin{aligned} F_i(\mathrm{x}) &= \mathrm{y}_i \prod_{1 \le j \ne i \le t} \frac{x - x_j}{x_i - x_j} \\ &= \begin{cases} y_i & x = x_i \\ 0 & x \in \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_t\} \end{cases} \end{aligned} \tag{2}$$

x_i is mapped by identity information id_i of the receiver.

2.2. Bilinear Maps. Let G_1 and G_2 be two multiplicative cyclic groups with the same prime order p. Let P be a generator of G_1. Let $e : G_1 \times G_1 \longrightarrow G_2$ be a bilinear map which has the following properties [19]:

(1) Bilinearity: $\forall Q, R \in G_1$ and $a, b \in \mathbb{Z}_p$, $e(Q, R) = e(aP, bP) = e(P, P)^{ab}$.

(2) Nondegeneracy: $\exists Q, R \in G_1$, such that $e(R, Q) \neq 1$. 1 denotes the identity element of G_2.

(3) Computability: $\forall Q, R \in G_1$; there is an efficient polynomial algorithm to calculate $e(Q, R)$.

According to the bilinearity, the bilinear mapping e has the following specific property:

$$e(aP, bP) = e(P, P)^{ab} = e(bP, aP) \tag{3}$$

2.3. Hard Problems. The following security assumptions are used in many encryption schemes. We will use them to deal with some problems in our scheme. In our paper, P denotes the generator of G_1.

(1) Computational Diffie-Hellman problem: given (P, aP, bP) for any $a, b \in \mathbb{Z}_q^*$, compute P^{ab}.

(2) Bilinear Diffie-Hellman (BDH) problem: given (P, aP, Pb, cP) for some $a, b, c \in \mathbb{Z}_q^*$ compute $e(g, g)^{abc}$.

(3) Cobilinear Diffie-Hellman (Co-BDH) problem [6]: given (P, aP, bP, Q) for any $a, b \in \mathbb{Z}_q^*$ and $Q \in G_1$, compute $e(P, Q)^{ab}$.

(4) Codecision bilinear Diffie-Hellman (Co-DBDH) problem [6]: given (P, aP, bP, Q, Z) for any $a, b \in \mathbb{Z}_q^*$, $Q \in G_1$ and $Z \in G_2$, decide whether $Z = e(P,Q)^{ab}$.

(5) Codecision bilinear Diffie-Hellman (Co-DBDH) assumption [5]: an algorithm B with an output $\beta \in \{0,1\}$ has advantage ε in solving the Co-DBDH problem if

$$\left| Pr\left[\mathrm{B}\left(P, aP, bP, Q, e\left(P,Q\right)^{ab}\right) = 1\right] - Pr\left[\mathrm{B}\left(P, aP, bP, Q, Z\right) = 1\right]\right| \geq \varepsilon \tag{4}$$

(6) Given two groups G_1 and G_2 of the same prime order q, $(P, \alpha P, \alpha^2 P, \ldots, \alpha^q P)$, $\alpha \in Z_p^*$, a generator P of G_1, and a bilinear map $e : G_1 \times G_1 = G_2, q$-bilinear Diffie-Hellman inversion (q-BDHI) problem is to compute $e(P,P)^{1/\alpha}$.

(7) Given two groups G_1 and G_2 of the same prime order q, $(P, \alpha P, \gamma)$, $\alpha \in Z_p^*$, $\gamma \in Z_p^*$, a generator P of G_1, and a bilinear map $e : G_1 \times G_1 = G_2$ the modified bilinear inverse Diffie-Hellman (mBIDH) problem is to compute $e(P,P)^{1/(\alpha+\gamma)}$.

2.4. Security Definition. According to the works [3, 5, 6], a general model and security formalization problem is given. Security formalization problem is indistinguishability encryptions of chosen ciphertext attacks, under selective multi-ID (IND-CCA-sMID for short) [5, 6]. The notion of IND-CCA-sMID is given as follows.

Definition 1 ((IND-CCA-sMID) [5, 6]). Let A be a polynomial-time algorithm attacker. Symbol $\prod$ denotes a general multireceiver IBE scheme. Attacker A interacts with the challenger in the following steps.

Setup. The challenger executes the setup algorithm. Attacker A attains the resulting public parameters from challenger. The attacker does not know any information about private key. The challenger keeps the master key secret.

Phase 1. A outputs multiple targets identities $(id_1, \ldots, id_t)$ where t denotes a positive integer.

Phase 2. A publishes private key extraction queries. When a private key extraction query with identity id_i is received, the challenger obtains private key $d_j = Extract(params, s, id_j)$ by running the private key extraction algorithm. The only constraint is that $id_j \neq id_i$ for $i = 1, \ldots, t$.

Phase 3. A publishes decryption queries for target identity information. When a decryption query denoted by (C^*, id_i) for some $t \in \{1, 2, \ldots, t\}$ is received, the challenger creates a private key which is denoted by d_i associated with identity information id_i. The challenger returns the information $D = Deccrypt(params, C^*, id_i, d_i)$ to A.

Challenge. A outputs a target plaintext message pair (M_0, M_1); the challenger randomly selects $\beta \in \{0,1\}$ and creates a target ciphertext information $C = Encrypt(params, id_1, \ldots, id_t, M_\beta)$. Ciphertext C is given to A by the challenger.

Phase 4. A publishes the private key extraction queries and decryption queries for target identities, and query methods are the same as in phase 2 and phase 3, respectively. Restrictive condition is that $C^* \neq C$.

Guess. To the end, A outputs the result of conjecture $\beta' \in \{0,1\}$. We can say that A wins the game if $\beta' = \beta$. $A's$ conjecture advantage is defined as follows:

$$\mathrm{Adv}_{\prod}^{IND-CCA-sMID}(A) = \left| Pr\left(\beta = \beta'\right) - \frac{1}{2}\right|. \tag{5}$$

Our scheme $\prod$ is said to be (τ, ε)-IND-CCA-sMID secure if the conjecture advantage $\mathrm{Adv}_{\prod}^{IND-CCA-sMID}(A)$ of any attacker A with polynomial running time τ is less than ε.

A breaks IND-CCA-sMID of $\prod$ with $(\tau, q_1, q_2, \varepsilon)$ if and only if the conjecture advantage of the attack A is not less than ε with the running time τ. q_1 and q_2 denote the number of private of key extraction queries and decryption queries, respectively. Scheme $\prod$ is said to be $(\tau, q_1, q_2, \varepsilon)$- IND-CCA-sMID secure if there is no polynomial-time algorithm attacker A with $(\tau, q_1, q_2, \varepsilon)$ that can break IND-CCA-sMID of scheme $\prod$.

3. The Proposed Encryption Scheme

In this section, we introduce a novel anonymous multireceiver IBE on the basis of offline/online encryption. Our scheme ensures both the confidentiality of the information and the anonymity of the receiver. The process of our encryption scheme is given in Figure 1. As shown in Figure 1, the system framework comprises three types of participants: *Sender, Receiver*, and *KGC*.

Sender. The sender encrypts the information and sends the ciphertext message to the designed receivers.

Receiver. The receiver can decrypt the ciphertext message according to the private key.

KGC. It is responsible for the generation of receivers' private keys.

In this section, an anonymous multireceiver online/offline IBE is proposed according to literature [6, 20]. Our encryption scheme usually consists of six algorithms as follows: *Setup, Key extract, Offline encryption, Online encryption*, and *Decryption*. In the following, we will describe the processes of our encryption scheme in detail.

Setup Phase. The algorithm works in setup phase as follows:

(1) Pick a random value $s \in \mathbb{Z}_q^*$, *and* $P_1 \in G_1$.

(2) Compute $P_{pub} = sP$.

(3) Select six one-way hash functions.

$H : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$, $H_1 : \{0,1\}^* \longrightarrow G_1^*$, $H_2 : \mathrm{G}_2 \longrightarrow \{0,1\}^w$, $H_4 : \{0,1\}^w \longrightarrow \{0,1\}^w$, $H_5 : \mathrm{G}_2 \parallel \{0,1\}^* \parallel \mathrm{G}_2 \times \mathrm{G}_2 \times \cdots \times \mathrm{G}_2, \longrightarrow \{0,1\}^z$, $z < w$, $H_6 : G_2 \longrightarrow \{0,1\}^l$, $l < w$, $l + z = w$. The symbols w, z, and l are some positive integers. They denote the length of binary data.

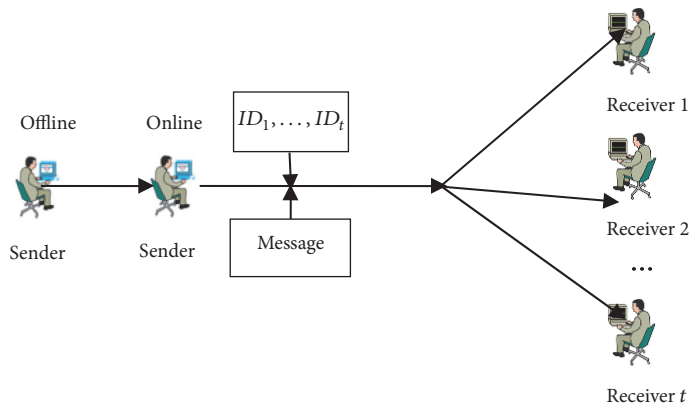


Figure 1: The process of our encryption scheme.

(4) Issue the public parameters $\{q, G_1, G_2, e, n, P, P_{pub}, H, H_1, H_2, H_4, H_5, H_6\}$ and make the private key $msk = s$ secret.

Private Key Extract Phase. Input public parameters and the identity information id_i of the receiver, and the PKC executes the algorithm as follows:

(1) Compute $Q_i = H_1(id_i)$.

(2) Compute the secret key d_i for the identity id_i of the receiver as

$$d_i = (d_{i1}, d_{i2}) = \left(s(Q_i + P_1), \frac{1}{H(id_i) + s}P\right) \quad (6)$$

Offline Encryption Phase. In this phase, the sender computes the following steps:

(1) Randomly choose $x \in \mathbb{Z}_q^*$ and compute $K = e(P, P)^x$;

(2) For $i = 1$ *to* t, randomly choose $y_{1i} \in \mathbb{Z}_q$, and computer $C_{i1} = (xP_{pub}) + (xy_{1i}P)$.

(3) Pick a random $r \in \mathbb{Z}_q^*$; compute $U = rP$.

(4) For $i = 1$ *to* t, randomly choose $\alpha_i \in \mathbb{Z}_q$; compute $y_i = \alpha_i^{-1} r \bmod q$.

(5) For $i = 1$ *to* t, compute

$$R_i = \sum_{j=1}^{t} a_{ji} y_j Q_j = \sum_{j=1}^{t} b_{ji} Q_j \quad (7)$$

(6) For $i = 1$ *to* t, compute

$$K_i = \alpha_i P_{pub} \quad (8)$$

Online Encryption Phase

(1) According to the identity information, compute each potential receiver's x_i and Q_i.

For $i = 1$ to t, compute $x_i = H(id_i)$ and $Q_i = H_1(id_i)$.

(2) $f_i(x)$ can be calculated, respectively, as follows:

For $i = 1$ to t, compute

$$f_i(x) = \prod_{1 \le j \le t, j \ne i} \frac{x - x_j}{x_i - x_j} = a_{i1} + a_{i2}x + \cdots + a_{it}x^{t-1} \quad (9)$$

Inputting message M and selecting t identities of the receivers, the sender performs the following steps.

Compute $V = H_6(K) \parallel H_5(K \parallel M \parallel K_1 \parallel \cdots \parallel K_t) \oplus H_2(e(P_{pub}, P_1)^r), W = E_{H_4(H_5(K \| M \| K_1 \| \cdots \| K_t))}(M)$, where $E(\cdot)$ denotes the symmetric encryption function.

For $i = 1$ to t, compute

$$C_{1i} = x(H(id_i) - y_{1i}) \quad (10)$$

The result ciphertext is $\widetilde{C} = (U = rP, K_1, \ldots, K_t, V, W, C_{11}, C_{12}, \ldots, C_{1t}, R_1, \ldots, R_t)$.

Decryption. Given ciphertext information $\widetilde{C}$, the legal receiver uses the private key to perform the tasks as follows:

(1) Compute $x_i = H(id_i)$.

(2) Compute $\lambda_i = R_1 + x_i R_2 + \cdots x_i^{t-1} R_t$.

(3) Compute $H_6(K) \parallel H_5(K \parallel M \parallel K_1 \parallel \cdots \parallel K_t) = V \oplus H_2(e(U, d_{i1})/e(K_i, \lambda_i))$.

(4) Compute $K = e(C_1 + (C_{1i}P), d_{i2})$ and separate $H_6(K) \parallel H_5(K \parallel M \parallel K_1 \parallel \cdots \parallel K_t)$ to obtain $H_5(K \parallel M \parallel K_1 \parallel \cdots \parallel K_t)$.

(5) Compute $M' = D_{H_4(H_5(K \| M \| K_1 \| \cdots \| K_t))}(W)$, where $D(\cdot)$ denotes the symmetric decryption function. Test whether $H_5(K \parallel M \parallel K_1 \parallel \cdots \parallel K_t) = H_5(K \parallel M' \parallel K_1 \parallel \cdots \parallel K_t)$ or not. Decrypted message M' is message M if equality holds.

4. Security and Performance Analysis

In Section 4, we first give the correctness analysis of our scheme, and then we compare security and computational cost with the previous literatures [5, 6, 8, 9].

4.1. Security Analysis. In Section 4.1, the correctness and security of our encryption scheme are analyzed.

4.1.1. Correctness. For each authorized receiver, it can decrypt the ciphertext by the following way. First, it can compute K' and then recover the message M' in the following way:

$$\begin{aligned} K' &= e\left(C_{i1} + \left(C_{1i}P\right), d_{i2}\right) = e\left(\left(xP_{pub}\right) + \left(xy_{1i}P\right)\right. \\ &\quad \left. + \left(x\left(H\left(id_i\right) - y_{1i}\right)\right)P, \frac{1}{H\left(id_i\right) + s}P\right) \\ &= e\left(\left(xsP + xH\left(id_i\right)P\right), \frac{1}{H\left(id_i\right) + s}P\right) \\ &= e\left(\left(sx + xH\left(id_i\right)\right)P, \frac{1}{H\left(id_i\right) + s}P\right) = e\left(P, P\right)^x \\ &= K \end{aligned} \tag{11}$$

For each authorized receiver i ($i \in [1, t]$) with identity id_i, the receiver i can compute $x_i = H(id_i)$ and structure function $\lambda(x)$. It can obtain λ_i using Lagrange interpolating polynomial theorem.

$$\begin{aligned} \lambda_i &= R_1 + x_i R_2 + \cdots + x_i^{i-1} R_i + \cdots + x_i^{t-1} R_t \\ &= \left(a_{11} + a_{12}x_i + \cdots + a_{1t}x^{t-1}\right) y_1 Q_1 + \cdots \\ &\quad + \left(a_{i1} + a_{i2}x_i + \cdots + a_{it}x_i^{t-1}\right) y_i Q_i + \cdots \\ &\quad + \left(a_{t1} + a_{t2}x_i + \cdots + a_{tt}x^{t-1}\right) y_t Q_t = y_i Q_i \end{aligned} \tag{12}$$

The receiver can obtain

$$\begin{aligned} \frac{e\left(U, d_{i1}\right)}{e\left(K_i, \lambda_i\right)} &= \frac{e\left(rP, s\left(Q_i + P_1\right)\right)}{e\left(\alpha_i P_{pub}, y_i Q_i\right)} \\ &= \frac{e\left(rP, sQ_i\right) e\left(rP, sP_1\right)}{e\left(P_{pub}, Q_i\right)^{\alpha_i y_i}} \\ &= \frac{e\left(sP, Q_i\right)^r e\left(sP, P_1\right)^r}{e\left(P_{pub}, Q_i\right)^r} = e\left(sP, P_1\right)^r \\ &= e\left(P_{pub}, P_1\right)^r. \end{aligned} \tag{13}$$

Thus, the authorized receiver can perform the following steps:

(1) $H_6(K) \parallel H_5(K \parallel M \parallel K_1 \parallel \cdots \parallel K_t) = V \oplus H_2(e(P_{pub}, P_1)^r)$.

(2) Separate $H_6(K) \parallel H_5(K \parallel M \parallel K_1 \parallel \cdots \parallel K_t)$ to obtain $H_5(K \parallel M \parallel K_1 \parallel \cdots \parallel K_t)$.

(3) Finally, the authorized receiver decrypts the ciphertext $M' = D_{H_4(H_5(K \| M \| K_1 \| \cdots \| K_t))}(W)$.

4.1.2. The Confidentiality of Message. In order to decrypt the ciphertext information, the decryptor should know the symmetric secret key $H_4(H_5(K \parallel M \parallel K_1 \parallel \cdots \parallel K_t))$.

From $H_6(K) \parallel H_5(K \parallel M \parallel K_1 \parallel \cdots \parallel K_t) = V \oplus H_2(e(P_{pub}, P_1)^r)$, we can know that the only way to obtain the symmetric secret key is to calculate $H_2(e(P_{pub}, P_1)^r)$. If decryptor is not an authorized receiver, he/she would deal with the Co-BDH problem to compute $e(P, P_1)^{sr}$. In order to make our paper rigorous and complete, Theorem 2 is given in detail according to the proving process of papers [5, 6].

Theorem 2. *Our proposed scheme is* $(\tau, q_H, q_{H_1}, q_{H_2}, q_{H_4}, q_{H_5}, q_{H_6}, q_1, q_2, \varepsilon)$- *IND-CCA-sMID secure under the* (τ', ε')- *codecision bilinear Diffie-Hellman assumption, where* $\varepsilon' \geq \varepsilon$ *and* $\tau' \simeq \tau + (qH_1 + qH_5 + q_1)O(\tau_1) + q_2O(\tau_1 + \tau_2) + qHO(1) + qH_2O(1) + qH_4O(1) + qH_6O(1)$. $q_H, q_{H_1}, q_{H_2}, q_{H_4}, q_{H_5}$, *and* q_{H_6} *denote the number of queries of hash functions,* H_1, H_2, H_4, H_5, *and* H_6.

Proof. Assume that A is a $(\tau, q_H, q_{H_1}, q_{H_2}, q_{H_4}, q_{H_5}, q_{H_6}, q_1, q_2, \varepsilon)$ attacker that can break our scheme $\prod$. The challenger B can resolve the Co-DBDH problem with advantage ε' in runtime time τ' by using A. According to the Co-DBDH assumption, the confidentiality of the proposed scheme can be guaranteed.

Assume that B is given the tuple $(q, G_1, G_2, e, n, P, aP, bP, Q, Z)$ as an instance of the Co-DBDH problem. B simulates the challenging environment in IND-CCA-sMID game for A as follows.

Phase 1. Assume that A output is the target identity information $(id_1, id_2, \ldots, id_t)$ where t is a positive integer.

Setup. B sets $P_1 = Q$ and $P_{pub} = bP$. B gives the public parameters N to the attacker A where $N = (q, G_1, G_2, e, n, P, P_1, P_{pub}, H, H_1, H_2, H_4, H_5, H_6)$ and n is a positive integer and denotes the number of all users. Let $T, T_1, T_2, T_4, T_5, T_6$ store the results of querying hash functions $H, H_1, H_2, H_4, H_5, H_6$, respectively.

H-Query. Input identity information id_j to H. B checks T. If there exists (id_j, x_j) in T return x_j. Otherwise, do the following steps:

(1) Pick a randomly integer $x_j \in \mathbb{Z}_q^*$.

(2) Put (id_j, x_j) to T.

(3) Return x_j.

H_1-Query. Input identity information id_j to H_1. B checks the T_1. If there exists (id_j, l_j, Q_j) in T_1 return Q_j. Otherwise, do the following steps:

(1) Pick a randomly integer $l_j \in \mathbb{Z}_q^*$.

(2) If $i \in \{1, 2, \ldots, t\}$ compute $Q_j = l_jP$ else compute $Q_j = l_jP - P_1$.

(3) Put (id_j, l_j, Q_j) to T_1.

(4) Return Q_j.

H_2-Query. Input $Z_j \in G_2$ to H_2. B checks the T_2. If there exists (Z_j, δ_j) in T_2, return δ_j. Otherwise, do the following steps:

(1) Pick a randomly string $\delta_j \in \{0, 1\}^w$.

(2) Put (Z_j, δ_j) to T_2 and return δ_j.

H_6-*Query*. Input $Z_j' \in G_2$ to H_6. B checks T_6. If there exists (Z_j', δ_j') in T_6 return δ_j'. Otherwise, do the following steps:

(1) Pick a randomly string $\delta_j' \in \{0,1\}^l$.

(2) Put (Z_j', δ_j') to T_6 and return δ_j'.

H_5-*Query*. Input $(K_j', \aleph_j, M_j)$ to H_5. B checks T_5. If there exists $(K_j', \aleph_j, M_j, \rho_j, \Gamma_j)$ in T_5 where $K_j', \Gamma_j \in G_2$, $\rho_j \in \mathbb{Z}_q^*$, $\aleph_j \in G_2 \times G_2 \times \cdots \times G_2$ return ρ_j. Otherwise, do the following steps:

(1) Pick a randomly sting $\rho_j \in \mathbb{Z}_q^*$ and compute $\Gamma_j = \rho_j P$.

(2) Put $(Q_j', \aleph_j, M_j, \rho_j, \Gamma_j)$ to T_5 and return ρ_j.

H_4-*Query*. Input $\sigma_j \in \{0,1\}^w$ to H_4. B checks T_4. If there exists (σ_j, η_j) in T_4 return η_j. Otherwise, do the following steps:

(1) Pick a randomly string $\eta_j \in \{0,1\}^w$.

(2) Put (σ_j, η_j) to T_4 and return η_j.

Phase 2. A issues private key extraction queries for id_j where $j \in \{1, 2, \ldots, \mathrm{t}\}$. B does the following steps:

(1) If there exists (id_j, l_j, Q_j) in T_1, then compute $d_{j1} = l_{j1} P_{pub}$; otherwise, pick a randomly integer $l_j \in \mathbb{Z}_q^*$ and compute $d_{j1} = l_j P_{pub}$, $Q_j = l_j P - P_1$.

(2) Put (id_j, l_j, Q_j) to T_1 and return d_{j1} to A.

Phase 3. A issues decryption query (C^*, id_i) for identity information id_i where $i \in [1,t]$ and $C^* = (U, K_1, \ldots, K_t, V, W, C_{11}, C_{12}, \ldots, C_{1t})$. B does the following steps:

(1) Search T_5 to obtain $(K_j', \aleph_j, M_j, \rho_j, \Gamma_j)$ when $\Gamma_j = K_j'$. If not found, return "reject" to A.

(2) Compute $x_i = H(id_i)$.

(3) Compute $\lambda_i = R_1 + x_i R_2 + \cdots x_i^{t-1} R_t$

(4) Compute

$$\sigma' = V \oplus H_2\left(\frac{e\left(P_{pub}, \rho_j P_1\right) e\left(U, l_i P_{pub}\right)}{e\left(K_i, \lambda_i\right)}\right) \tag{14}$$

where

$$\begin{aligned} e\left(P_{pub}, \rho_j P_1\right) e\left(U, l_i P_{pub}\right) &= e\left(sP, \rho_j P_1\right) e\left(U, l_i sP\right) \\ &= e\left(\rho_j P, sP_1\right) e\left(U, l_i sP\right) = e\left(U, sP_1\right) e\left(U, l_i sP\right) \\ &= e\left(U, s\left(P_1 + Q_i\right)\right) = e\left(U, d_{i1}\right) \end{aligned} \tag{15}$$

(5) Set $K = K_j'$ and separate $\eta' = H_5(K \parallel M_j \parallel K_1 \parallel \cdots \parallel K_t)$ from σ'.

(6) Test whether $M_j = D_{H_4(\sigma')}(W)$ or not. If not, return "reject" to A; else return M_j to A.

Challenge. A outputs a plaintext information pair (M_0, M_1). When receiving (M_0, M_1), B does the following steps:

(1) Randomly pick $\beta \in \{0,1\}$.

(2) For $i = 1, 2, \ldots, t$ search T_1 to obtain l_i according to id_i.

(3) Set $U = aP = rP$ and $\mathrm{I} = Z$.

(4) For $i = 1, 2, \ldots, t$ compute

$$f_i(x) = \prod_{1 \le j \ne i \le t} \frac{x - x_j}{x_j - x_i} = a_{i1} + a_{i2} x + \cdots + a_{it} x^{t-1} \tag{16}$$

where $a_{ij} \in \mathbb{Z}_q$, $j \in \{1, 2, \ldots, t\}$.

(5) Pick a randomly integer $\alpha_i \in \mathbb{Z}_q^*$. For $i = 1, 2, \ldots, t$ compute

$$\begin{aligned} R_i &= \sum_{j=1}^{t} a_{ji} \alpha_j^{-1} l_j U = \sum_{j=1}^{t} a_{j,i} \alpha_j^{-1} r l_j P = \sum_{j=1}^{t} a_{ji} y_j Q_j \\ &= \sum_{j=1}^{t} b_{ji} Q_j \end{aligned} \tag{17}$$

$$K_i = \alpha_i P_{pub}$$

$$C_{1i} = x\left(H_1\left(id_i\right) - y_{1i}\right)$$

(6) Compute $K = e(C_{i1} + (C_{1i} P), d_{i2})$.

(7) Set $\xi = H_5(K \parallel M_\beta \parallel K_1 \parallel \cdots \parallel K_t)$, where $\xi \in \{0,1\}^z$ and create a target ciphertext information C, where $C = (U, K_1, K_2, \ldots, K_t, H_6(K) \parallel \xi \oplus H_2(\mathrm{I}), E_{H_4(\xi)}(M_\beta))$.

(8) Return C to attacker A.

Phase 4. A publishes private key extraction and decryption queries, and they are the same as phases 2 and 3. The constraint condition of decryption queries is that $C^* \ne C$.

Guess. To the end, A outputs the guessing result $\beta' \in \{0,1\}$. If $\beta' = \beta$ then B outputs 1; else it outputs 0. If $\mathrm{I} = e(P,Q)^{ab}$ then

$$\begin{aligned} H(K) \parallel \xi \oplus H_2(\mathrm{I}) &= H_6(K) \parallel \xi \oplus H_2\left(e\left(bP, Q\right)^a\right) \\ &= H_6(K) \parallel \xi \oplus H_2\left(e\left(P_{pub}, P_1\right)^r\right) \end{aligned} \tag{18}$$

That is to say, C is a valid ciphertext message. Otherwise, K is a randomized element of G_2 and C is invalid. According to the above constructions, B simulates the random oracles hash function $\{H, H_1, H_2, H_4, H_5, H_6\}$, the private key extraction, and the decryption queries in *phases 2, 3*, and *4* successfully. So, we have

$$\left| Pr\left[\mathrm{B}\left(P, aP, bP, Q, e\left(P,Q\right)^{ab}\right) = 1\right] = \Pr\left[\beta' = \beta\right]\right| \tag{19}$$

where $|\Pr[\beta' = \beta] - 1/2| \ge \varepsilon$, and $Pr[\mathrm{B}(P, aP, bP, Q, Z) = 1] = 1/2$, Z is a random element in G_2. Hence, we obtain

Table 1: The notations of the symbols.

Symbols	Meanings of the symbols
n	The number of all users
t	The number of authorized receivers, $t \in [1, n]$
C_P	The computational cost of one bilinear pairing operation
C_{Enc}	The cost of symmetric encryption or decryption one message
C_{Sm}	The computational cost of one scalar multiplication in G_1
C_{Ex-z}	The computational cost of exponentiation computation in Z_q^*
C_{Mu-z}	The computational cost of multiplication in Z_q^*
C_{Mu}	The computational cost of multiplication in G_1
C_{Add-z}	The computational cost of addition in Z_q^*
C_{Add}	The computational cost of addition in G_1
C_{Ex}	The computational cost of exponentiation computation in G_2
C_H	The computational cost of one general hash operation
C_{H-p}	The computational cost of one hash-to-point operation
L_{Ecp}	The bit length of an element in G_1.
L_H	The length of hash value
L_{Enc}	The bit length of a plaintext messages

$$\begin{aligned} &\left|Pr\left[\mathrm{B}\left(P, aP, bP, Q, e\left(P,Q\right)^{ab}\right)=1\right]\right. \\ &\left.-\,Pr\left[\mathrm{B}\left(P, aP, bP, Q, Z\right)=1\right]\right| \geq \left|\left(\frac{1}{2}\pm\varepsilon\right)-\frac{1}{2}\right| \\ &=\varepsilon \end{aligned} \tag{20}$$

Thus, $\varepsilon' \geq \varepsilon$ and $\tau' \simeq \tau + (qH_1 + qH_5 + q_1)O(\tau_1) + q_2O(\tau_1 + \tau_2) + qHO(1) + qH_2O(1) + qH_4O(1) + qH_6O(1)$. $q_H, q_{H_1}, q_{H_2}, q_{H_4}$, q_{H_5}, and q_{H_6} denote the number of queries to hash function, H_1, H_2, H_4, H_5, and H_6. □

4.1.3. The Anonymity of Receivers. Fan et al.'s encryption strategy [5] cannot satisfy the anonymity of multireceiver. Every legal receiver can easily verify whether anyone is a legal receiver or not. A legal receiver with identity id_i can compute $f(H(id_k))$, where id_k denotes the identity information of the receiver. If equation $e(Q_i, f(H(x_k))) = e(Q_k, f(H(x_i)))$ holds, the receiver with identity id_k is an authorized receiver. In order to achieve multireceiver anonymously, Wang et al. [6] improved the multireceiver anonymous encryption scheme. For a ciphertext, Q_i, P_{pub}, r are fixed 9 in their scheme. The authorized receiver with identity information id_i can obtain value r from decryption process, although the numerical value of symbol r is randomly generated in encryption stage. If the equation $e(\lambda(H(id_l)), \upsilon(H(id_l))) = e(H_1(id_l), P_{pub})^r$ holds, the receiver with identity id_l also is an authorized receiver. Random number r can be recovered by symmetric key and message M. Unfortunately, their encryption scheme cannot protect the privacy of the receiver. That is to say, it did not satisfy the anonymity of the receiver.

In our proposed scheme, the above problems are solved. Only the authorized receiver can decrypt ciphertext information. Each receiver does not know whether others are authorized receivers or not. Thus, the privacy of the user can be protected.

Theorem 3. *Our scheme satisfies the anonymity of receiver if the Co-DBDH problem is hard.*

In this work, we do not give the proofs of Theorem 3. We can refer to literature [9] and literature [8] for details.

Theorem 4. *In the random oracle model, our scheme is IND-CCA2 secure under the q-BDHI and mBIDH assumptions.*

This proof is similar to the proof of literature [21, 22]. Please refer to literature [21, 22] for details.

4.2. Performance Analysis. In this section, the computational consumption of our scheme is given. In order to analyze the computational performance, some notations of the symbols are summarized in Table 1.

The implementation environment is on a mobile phone (Samsung Galaxy S5 with a Quad-core 2.45G processor, 2G bytes memory, and the Google Android 4.4.2 operating system) [10]. The implementation runtime results of main operations are listed in Table 2 [10, 23]. The efficiency comparison is summarized in Tables 3 and 4. The computational cost in our scheme is compared to literature [5, 6, 8, 9]. In addition, the mentioned five schemes contain encryption and decryption computational cost. From Table 5, we can see that our scheme is nearly identical to the ciphertext length of other schemes in [5, 6, 8, 9]. As shown from Table 6, our offline/online encryption scheme is the same as literature [8, 9], and encryption schemes of them are anonymous. However, literature [8] and literature [9] do not use the offline/online encryption scheme.

From Tables 3 and 4, we can see that our scheme needs one bilinear pairing operation and three bilinear pairing operations in encryption phase and decryption phase. The number of bilinear pairing operation increases linearly with the number of recipients in encryption and decryption phase

Table 2: Computational cost of main operations.

Operations	Computational cost (ms)
C_P	32.713
C_{H-p}	33.582
C_{Sm}	13.401
C_{Add}	0.056
C_{Ex-z}	0.002
C_{Ex}	2.249
C_H	0.006
C_{Enc}	0.001
C_{Add-z}	0.001
C_{Mu-z}	0.001
C_{Mu}	0.008

in literature [9]. Bilinear pairing operation requires a lot of calculation consumption. It is not suitable for mobile devices with limited energy. From Table 6, we know that only our scheme uses offline/online encryption.

In order to give an intuitive knowledge, Figures 2 and 3 also describe the computational cost in encryption and decryption schemes, respectively. Symbol t denotes the number of authorized receivers in Figures 2 and 3. According to Tables 2 and 3, we can easily compute the runtime of encryption and decryption scheme at different literatures [5, 6, 8, 9]. The computational cost on encryption and decryption is summarized in Figures 2 and 3, respectively. From Figure 2, we can see that the computational cost in [9] is the least, and our proposed scheme consumes little computation time in encryption. Literature [9] does not use offline/online encryption, and computational cost in our proposed scheme contains runtime of offline encryption and online encryption. When receivers decrypt the ciphertext, our scheme consumes the minimum computation time. As legal receiver t increases, the computational cost increases gradually in Figures 2 and 3.

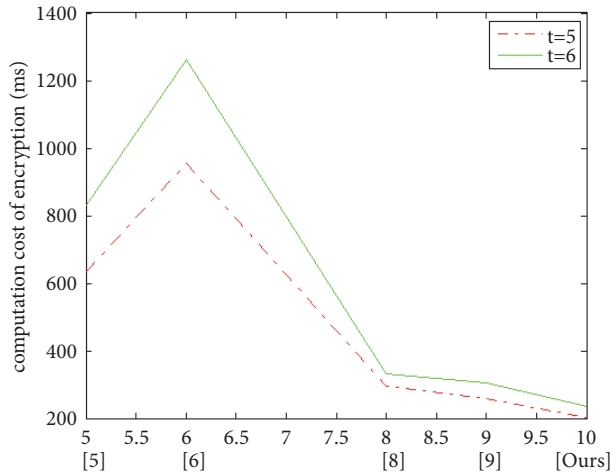


Figure 2: Computational cost of different scheme in encryption.

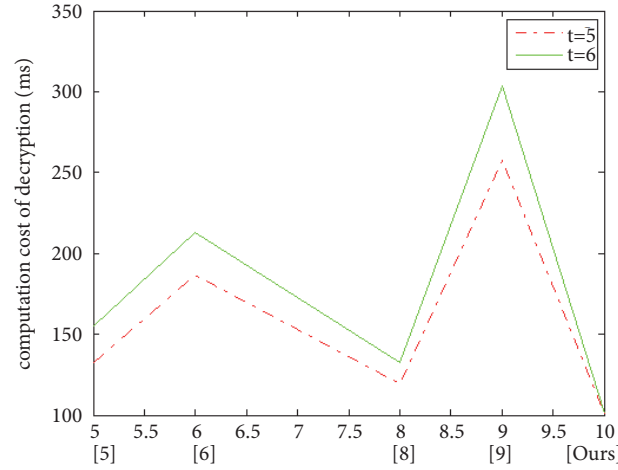


Figure 3: Computational cost of different scheme in decryption.

5. Conclusion

Finally, conclusion and future work are summarized. An anonymous multireceiver online/offline identity-based encryption was proposed in our work. We developed an efficient offline/online encryption scheme which can ensure the anonymity of the receiver. Our scheme divided encryption into two phases: offline and online. A sender can do a lot of preparatory calculations on offline phases, and a receiver can encrypt the message with little computational cost on online phases. The computational cost of the receivers was improved in the proposed scheme. The analysis results demonstrated that our scheme is secure and efficient, and it is suitable for mobile devices. The preparation computation can be done while mobile device is plugged into a power supply. When it is on the move without plugging, it performs the encryption operations with little computational cost.

An interesting future work is that we will pay more attention to anonymous attribute-based encryption using offline/online scheme for mobile devices.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of Interest.

Acknowledgments

The work is supported by the Supporting Fund for Teachers' Research of Jining Medical University (no. JY2017KJ053), Qing Lan Project of Jiangsu Province, 1311 Talent Plan Foundation of Nanjing University of Posts and Telecommunications, and Doctoral Research Fund of Jining Medical University.

Table 3: The comparisons of computation cost in different encryption schemes.

Scheme	The cost of encryption
Literature [5]	$(t+3)C_H + tC_{H-p} + \left(t^2+1+\left(t^2+2t-2\right)(t-1)^2\right)C_{Mu-z} + \left(t^2+t+2\right)C_{Sm} + C_{Ex} + C_{Ex-z} + \left(\left(t^2+t-1\right)(t-1)^2\right)C_{Add-z} + t(t-1)C_{Add} + C_P$
Literature [6]	$(t+3)C_H + tC_{H-p} + \left(t^2+t+\left(t^2+2t-2\right)(t-1)^2\right)C_{Mu-z} + \left(2t^2+t+1\right)C_{Sm} + C_{Ex} + tC_{Ex-z} + \left(\left(t^2+t-1\right)(t-1)^2\right)C_{Add-z} + 2t(t-1)C_{Add} + C_P$
Literature [8]	$(t+4)C_H + tC_{H-p} + \left(t^2+t+\left(t^2+2t-2\right)(t-1)^2\right)C_{Mu-z} + (t+1)C_{Sm} + C_{Ex} + tC_{Ex-z} + \left(\left(t^2+t-1\right)(t-1)^2\right)C_{Add-z} + t(t-1)C_{Mu} + C_P$
Literature [9]	$(t+2)C_H + (t+2)C_{Sm} + tC_P$
Proposed scheme	$tC_H + tC_{H-p} + t(t-1)C_{Add-z} + t(t-1)C_{Mu-z} + t(t-1)C_{Ex-z} + C_p$

Table 4: The comparisons of computation cost in different decryption schemes.

Scheme	The cost of decryption
Literature [5]	$3C_H + tC_{Sm} + (t-1)C_{Ex-z} + (t-1)C_{Add} + 2C_P$
Literature [6]	$3C_H + (2t-1)C_{Sm} + 2(t-1)C_{Ex-z} + 2(t-1)C_{Add} + 2C_P$
Literature [8]	$6C_H + (t-1)C_{Sm} + (t-2)C_{Ex-z} + (t-1)C_{Add} + 2C_P$
Literature [9]	$(t+2)C_H + (t+2)C_{Sm} + tC_P$
Proposed scheme	$6C_H + tC_{Add} + (t-2)C_{Ex-z} + 3C_P + C_{Ex}$

Table 5: The comparisons of ciphertext length in different schemes.

Scheme	The length of ciphertext
Literature [5]	$(t+2)L_{Ecp} + L_H + L_{Enc}$
Literature [6]	$(2t+2)L_{Ecp} + L_H + L_{Enc}$
Literature [8]	$(2t+1)L_{Ecp} + L_H + L_{Enc}$
Literature [9]	$(t+2)L_{Ecp} + L_H + L_{Enc}$
Proposed scheme	$(3t+1)L_{Ecp} + L_H + L_{Enc}$

Table 6: Comparison of different schemes.

Scheme	Offline/online encryption	Anonymity
Literature [5]	no	no
Literature [6]	no	no
Literature [8]	no	yes
Literature [9]	no	yes
Proposed scheme	yes	yes

References

[1] M. Bellare, A. Boldyreva, and J. Staddon, "Randomness re-use in multi-recipient encryption schemes," in *Public key cryptography (PKC) 2003*, vol. 2567 of *Lecture Notes in Comput. Sci.*, pp. 85–99, Springer, Berlin, 2002.

[2] A. Fiat and M. Naor, *Broadcast Encryption. In Crypto '94, LNCS 773*, Springer-Verlag, 1994.

[3] J. Baek, R. Safavi-Naini, and W. Susilo, "Efficient multi-receiver identity-based encryption and its application to broadcast encryption," in *Public key cryptography—PKC 2005*, vol. 3386 of *Lecture Notes in Computer Science*, pp. 380–397, Springer, Berlin, Germany, 2005.

[4] L. Lu and L. Hu, "Pairing-based multi-recipient public key encryption," in *Proceedings of the International Conference on Security & Management*, pp. 159–165, 2006.

[5] C.-I. Fan, L.-Y. Huang, and P.-H. Ho, "Compact Anonymous Hierarchical Identity-Based Encryption with Constant Size Private Keys," *The Computer Journal*, vol. 59, no. 4, pp. 452–461, 2018.

[6] H. Wang, Y. Zhang, H. Xiong, and B. Qin, "Cryptanalysis and improvements of an anonymous multi-receiver identity-based encryption scheme," *IET Information Security*, vol. 6, no. 1, pp. 20–27, 2012.

[7] H. Li and L. Pang, "Cryptanalysis of Wang et al.'s improved anonymous multi-receiver identity-based encryption scheme," *IET Information Security*, vol. 8, no. 1, pp. 8–11, 2014.

[8] J. Zhang and J. Mao, "Comment on Anonymous Multi-receiver Identity-Based Encryption Scheme," in *Proceedings of the Fourth International Conference on Intelligent networking and Collaborative Systems*, pp. 473–476, 2012.

[9] H.-Y. Chien, "Improved anonymous multi-receiver identity-based encryption," *The Computer Journal*, vol. 55, no. 4, pp. 439–446, 2012.

[10] D. He, H. Wang, L. Wang, J. Shen, and X. Yang, "Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices," *Soft Computing*, vol. 21, no. 22, pp. 6801–6810, 2017.

[11] C. F. Guo, Y. Mu, and Z. D. Chen, "Identity-Based Online/Offline Encryption," in *Financial Cryptography and Data Security*, pp. 247–261, Cozumel, Mexico, 2008.

[12] Z. Pooranian, K. Chen, C. Yu, and M. Conti, "RARE: Defeating side channels based on data-deduplication in cloud storage," in *Proceedings of the IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 444–449, Honolulu, HI, April 2018.

[13] S. Javanmardi, M. Shojafar, S. Shariatmadari, and S. S. Ahrabi, "FR trust: a fuzzy reputation-based model for trust management

in semantic P2P grids," *International Journal of Grid and Utility Computing*, vol. 6, no. 1, pp. 57–66, 2015.

[14] C. Chu, J. K. Liu, J. Zhou, F. Bao, and R. H. Deng, "Practical ID-based encryption for wireless sensor network," in *Proceedings of the the 5th ACM Symposium*, pp. 337–340, Beijing, China, April 2010.

[15] J. K. Liu, J. Baek, J. Zhou, Y. Yang, and J. W. Wong, "Efficient online/offline identity-based signature for wireless sensor network," *International Journal of Information Security*, vol. 9, no. 4, pp. 287–296, 2010.

[16] J. Lai, Y. Mu, F. Guo, and W. Susilo, "Improved Identity-Based Online/Offline Encryption," in *Information Security and Privacy*, vol. 9144 of *Lecture Notes in Computer Science*, pp. 160–173, Springer International Publishing, 2015.

[17] Z. Wang, F. Chen, and A. Xia, "Attribute-based online/offline encryption in smart grid," in *Proceedings of the 24th International Conference on Computer Communications and Networks, ICCCN 2015*, pp. 1–5, 2015.

[18] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Public-key cryptography (PKC)*, vol. 8383 of *Lecture Notes in Comput. Sci.*, pp. 293–310, Springer, Heidelberg, 2014.

[19] S. Ruj, A. Nayak, and I. Stojmenovic, "Distributed fine-grained access control in wireless sensor networks," in *Proceedings of the 25th IEEE International Parallel and Distributed Processing Symposium, IPDPS 2011*, pp. 352–362, Anchorage, Alaska, USA, May 2011.

[20] C. Chu K, J. Liu K, and J. Zhou, "Practical ID-based Encryption for Wireless Sensor Network," in *Proceedings of Acm Symposium on Information Computer Communications Security*, pp. 337–340, 2010.

[21] F. Li, Y. Han, and C. Jin, "Certificateless online/offline signcryption for the Internet of Things," *Wireless Networks*, vol. 23, no. 1, pp. 145–158, 2017.

[22] C. Gentry, "Practical identity-based encryption without random oracles," in *Advances in cryptology—EUROCRYPT*, vol. 4004 of *Lecture Notes in Comput. Sci.*, pp. 445–464, Springer, Berlin, 2006.

[23] Li. G. F. and W. F. Wu, *Pairing-Based Cryptography*, Science Press, 2014.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 6925019, 11 pages
https://doi.org/10.1155/2018/6925019

Research Article

Strong Identity-Based Proxy Signature Schemes, Revisited

Weiwei Liu,[1] **Yi Mu**,[2] **Guomin Yang**,[2] **and Yangguang Tian**[3]

[1]*School of Mathematics and Statistics, North China University of Water Resources and Electric Power, Zhengzhou 450046, Henan, China*
[2]*Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, NSW 2522, Australia*
[3]*School of Information Systems, Singapore Management University, Singapore*

Correspondence should be addressed to Weiwei Liu; liuweiwei@ncwu.edu.cn

Received 7 March 2018; Revised 30 May 2018; Accepted 14 June 2018; Published 6 August 2018

Academic Editor: Kim-Kwang Raymond Choo

Proxy signature is a useful cryptographic primitive that has been widely used in many applications. It has attracted a lot of attention since it was introduced. There have been lots of works in constructing efficient and secure proxy signature schemes. In this paper, we identify a new attack that has been neglected by many existing proven secure proxy signature schemes. We demonstrate this attack by launching it against an identity-based proxy signature scheme which is proven secure. We then propose one method that can effectively prevent this attack. The weakness in some other proxy signature schemes can also be fixed by applying the same method.

1. Introduction

Proxy signature is a special type of digital signature which allows one user (original signer) to delegate his/her signing right to another signer (proxy signer). The latter can then issue signatures on behalf of the former. The corresponding proxy signature can be verified by the public that it is indeed generated by the proxy signer with proper delegation from the original signer [1, 2]. Proxy signature has been found useful in many applications, such as distributed computing [3], electronic commerce [4], mobile agents [5], and grid computing [6]. It is worth noticing that proxy signature can also serve as a useful tool in Internet of things (IoT), since most of the RFID tags in IoT only have limited storage and computing ability. For those operations involving a large amount of computation, those tags can authorize the tag readers with strong computing ability to perform those operations with the help of a proxy signature scheme [7, 8].

The concept of proxy signature was introduced by Mambo, Usuda, and Okamoto in 1996 [9]. They presented three different types of proxy signature, namely, full delegation, partial delegation, and delegation by warrant in their seminal work. Shortly after Mambo et al.'s work, Kim et al. [10] proposed a new type of proxy signature combing partial delegation and warrant. They demonstrated that schemes combining partial delegation and warrant can provide a higher level of security than schemes based on partial delegation or warrant separately. Since then, proxy signature has been extensively researched in different settings, such as blind proxy signature [11], anonymous proxy signature [12], and identity-based proxy signature [13].

These delegation-by-warrant proxy signature schemes can be further classified into two categories according to whether the proxy signature is generated by the proxy signer using his own private key or not. In the first type, the proxy signer generates a new proxy signing key using the delegation information and his own private key. The proxy signatures are generated under the new proxy signing key. The proxy signature schemes in [5, 14–17] fall into the first type. In the second type, the proxy signer issues a proxy signature using his own private key. The proxy signatures are essentially combinations of the original signer's signature on the warrant and the proxy signer's signature on the message. Such proxy signature schemes could be found in [13, 18–21].

On the security modelling of proxy signature, Boldyreva et al. [22] proposed a comprehensive security model for

the delegation-by-warrant proxy signature, where an original signer can also perform self-delegation. Malkin et al. [23] extended the security model to allow fully hierarchical proxy signatures. They also proved that proxy signatures are essentially equivalent to key-insulated signatures. The security model proposed in [22, 23] is in the registered key model, which means the adversary has to submit every public and private key pair in the security game except the challenge one. Later, Schuldt et al. [24] proposed an enhanced security model for proxy signature by allowing the adversary to query arbitrary proxy signing keys. Roughly speaking, a secure proxy signature scheme should satisfy the following requirements.

(i) **Verifiability**: given a proxy signature, a verifier can be convinced that the proxy signature is indeed a valid signature generated by the proxy signer with proper delegation from an original signer on the signed message.

(ii) **Identifiability**: given a proxy signature, a verifier is able to determine the identities of the corresponding original signer and proxy signer.

(iii) **Unforgeability**: no one, except the designated proxy signer, can create a valid proxy signature.

(iv) **Untenability**: a proxy signer cannot deny at a later time on a proxy signature that he has created before.

(v) **Prevention of misuse**: it is required in the first type of proxy signature schemes that the proxy signing key cannot be used for purposes other than creating proxy signatures. Once misused, the identity of the misbehaving proxy signer can be determined explicitly.

1.1. Our Contribution. We revisit proxy signature and show an attack that has been neglected by the second type of proxy signature schemes [13, 18–21] that have been proven secure. In these schemes, a proxy signature is essentially the combination of the original signer's standard signature on a warrant and the proxy signer's standard signature on a message. In the security analysis, it is assumed that an adversary has access to the original signer and proxy signer's standard signature oracles. We show that, under such a circumstance, some proxy signature schemes [13, 18–21] that have been previously proved secure are in fact not secure.

We demonstrate a new attack by launching it against an identity-based proxy signature scheme [13] that has been proven secure. We show that a malicious adversary can create a proxy signature on a message, if he has access to the standard signature of the original signer and proxy signer, which is as defined in the security models in [13, 18]. Thus, these proxy signature schemes [13, 18–21], which we believe is not a complete list, are in fact not secure. We propose an efficient solution by revising the identity-based proxy signature scheme [13] to thwart this attack. It is worth noticing that the same method can also be applied to [18–21] to resist this attack.

We have noticed there have been several works [5, 22] aiming to transform normal proxy signature schemes into strong ones. The authors in [22] suggested to add two different prepositive tags "00" and "11" to distinguish the signatures generated by the original signer and proxy signer. However, this simple solution cannot prevent the attack proposed in this paper according to the original security model in [13]. The adversaries are able to query any message of their choices. To stop the proxy signer from misusing the proxy signing key, the authors in [5] classified existing proxy signature schemes into strong and weak ones and proposed one method to transform weak proxy signature schemes into strong ones. However, as have been mentioned above, their method is only applicable when a proxy signature is generated from a proxy signing key which is created by the proxy signer using the delegation information and his own private key. Therefore, the method proposed in [5] is not suitable for the scenarios discussed in this paper.

Paper Organization. The rest of the paper is organized as follows. We introduce some preliminaries in Section 2. Then we present a new attack in some proxy signature schemes in Section 3 by attacking an identity-based proxy signature scheme. The security model for proxy signature that captures the attack is presented in Section 4. We then revise the identity-based proxy signature scheme in Section 5. The security proof and efficiency analysis are presented in Section 6 and the paper is concluded in Section 7.

2. Preliminaries

In this section, we introduce some preliminaries used throughout this paper.

2.1. Bilinear Map. Let $\mathbb{G}_1$, $\mathbb{G}_2$ be two cyclic groups of prime order q and P a generator of $\mathbb{G}_1$. The $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ is said to be an admissible bilinear map if the following conditions hold:

(i) Bilinearity: $e(aP_1, bP_2) = e(abP_1, P_2) = e(P_1, abP_2)$ for all $P_1, P_2 \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q$.

(ii) Nondegeneracy: there exists $P_1, P_2 \in \mathbb{G}_1$ such that $e(P_1, P_2) \neq 1_{\mathbb{G}_2}$.

(iii) Computability: there is an efficient algorithm to compute $e(P_1, P_2)$ for all $P_1, P_2 \in \mathbb{G}_1$.

2.2. Complexity Assumption

Definition 1 (computational Diffie-Hellman (CDH) problem). Given $P, aP, bP \in \mathbb{G}_1$ for some random $a, b \in \mathbb{Z}_q$, compute $abP \in \mathbb{G}_1$. Define the success probability of a polynomial algorithm $\mathscr{A}$ in solving the CDH problem as

$$Succ^{CDH}_{\mathscr{A},\mathbb{G}_1}(\kappa) = \Pr\left[\mathscr{A}(P, aP, bP) = abP : a, b \in_R \mathbb{Z}_q\right] \quad (1)$$

where $\kappa = \log(q)$ is the security parameter. The CDH assumption states that, for any polynomial algorithm adversary $\mathscr{A}$, $Succ^{CDH}_{\mathscr{A},\mathbb{G}_1}(\kappa)$ is negligible in κ.

3. A New Attack in Some Proxy Signature Schemes

In this section, we present an attack that has been neglected by many existing proxy signature schemes [13, 18–21]. To better explain how an attacker works, we demonstrate this attack via a concrete example. Before we start to introduce the attack, we first review an identity-based proxy signature scheme proposed in [13].

3.1. An Identity-Based Proxy Signature Scheme

(1) **Setup**: let $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ be a bilinear pairing map, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are of prime order q. Let P be a generator of $\mathbb{G}_1$. Choose a random number $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$. Select three collision-resistant hash functions H_0, H_1, H_2 such that $H_0, H_1, H_2 : \{0,1\}^* \longrightarrow \mathbb{G}_1$. The system parameters $params = \{e, \mathbb{G}_1, \mathbb{G}_2, q, P_{pub}, H_0, H_1, H_2\}$, the master secret key $Msk = s$.

(2) **KeyExtract**: on input a user's identity ID, output the secret key for this identity $sk_{ID} = sH_0(ID)$.

(3) **StandardSign**: on input a message m, the standard signature on m under identity ID is $\sigma = (\sigma_1, \sigma_2)$ such that $\sigma_1 = sk_{ID} + rH_1(M)$ and $\sigma_2 = rP$, where $r \in \mathbb{Z}_q$.

(4) **StandardVer**: on input a standard signature $\sigma = (\sigma_1, \sigma_2)$ of message m under identity ID, output "1" if $e(\sigma_1, P) = e(H_0(ID), P_{pub})e(H_1(m), \sigma_2)$; otherwise, output "0".

(5) **DelegationGen**: let w be a warrant that includes the delegation information such as the identities of the original signer and the designated proxy signer, the delegation period, the types of messages that a proxy signer can sign, and so on. Then the original signer with identity ID_A generates the delegation information $\sigma_w = (\sigma_{W_1}, \sigma_{W_2})$ such that $\sigma_{W_1} = sk_{ID_A} + r_A H_1(m_w)$ and $\sigma_{W_2} = r_A P$, where $r_A \in \mathbb{Z}_q$. The original signer sends the delegation signing key σ_w to the proxy signer.

(6) **ProSign**: upon receiving the delegation information $\sigma_w = (\sigma_{W_1}, \sigma_{W_2})$ and w from the original signer, the proxy signer with identity ID_B generates a proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$ on a message m such that $\sigma_{M_1} = \sigma_{W_1} + sk_{ID_B} + r_B H_2(m)$, $\sigma_{M_2} = \sigma_{W_2}$, $\sigma_{M_3} = r_B P$.

(7) **ProVer**: on input the identities ID_A, ID_B of the original signer and proxy signer, a warrant $w \in \{0,1\}^*$ and a message $m \in \{0,1\}^*$ and the proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$, output "1" if

$$e\left(\sigma_{M_1}, P\right) = e\left(H_0\left(ID_A\right), P_{pub}\right) e\left(H_1\left(w\right), \sigma_{M_2}\right) \cdot e\left(H_0\left(ID_B\right), P_{pub}\right) e\left(H_2\left(m\right), \sigma_{M_3}\right). \quad (2)$$

Otherwise, output "0".

3.2. An Attack against the ID-Based Proxy Signature Scheme.

Wu et al.'s identity-based proxy signature scheme [13] is proven secure. However, we show below that if the original signer and proxy signer also use their private keys to generate standard signatures, which is just as defined in their security models, then their scheme could be broken by a malicious outsider attacker. Assume the identities of the original signer and proxy signer are ID_A, ID_B, respectively, in the security model in [13], three types of adversaries are defined, namely,

(i) $\mathcal{A}_I$, which is an outsider adversary that has knowledge of (ID_A, ID_B),

(ii) $\mathcal{A}_{II}$, which is a malicious proxy signer that has knowledge of (ID_A, ID_B, sk_{ID_B}),

(iii) $\mathcal{A}_{III}$, which is a malicious original signer that has knowledge of (ID_A, sk_{ID_A}, ID_B).

The original signer and proxy signer could use the same key pairs to generate normal signatures using the standard signature scheme introduced in [13]. Suppose $\mathcal{A}_I$ aims to generate a proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$ on a message m with a warrant w; it is worth noticing that $\mathcal{A}_I$ might obtain such a genius warrant w when verifying a valid proxy signature. Then $\mathcal{A}_I$ acts as follows:

(i) $\mathcal{A}_I$ requires a standard signature $(\sigma_{A_1}, \sigma_{A_2})$ on warrant w of the original signer with identity ID_A, where w is a warrant containing the delegation information. The original signer chooses a random $r_A \in \mathbb{Z}_q$ and generates the standard signature $(\sigma_{A_1}, \sigma_{A_2})$ such that $\sigma_{A_1} = sk_{ID_A} + r_A H_1(w)$ and $\sigma_{A_2} = r_A P$.

(ii) Upon receiving the standard signature $(\sigma_{A_1}, \sigma_{A_2})$ on w from the original signer. $\mathcal{A}_I$ aborts if $e(\sigma_{A_1}, P) \neq e(H_0(ID_A), P_{pub})e(H_1(w), \sigma_{A_2})$.

(iii) $\mathcal{A}_I$ requires a standard signature $(\sigma_{B_1}, \sigma_{B_2})$ on message $w \parallel m$ of the proxy signer with identity ID_B, where m is a message. The proxy signer chooses a random $r_B \in \mathbb{Z}_q$ and generates the standard signature $(\sigma_{B_1}, \sigma_{B_2})$ such that $\sigma_{B_1} = sk_{ID_B} + r_B H_2(w, m)$ and $\sigma_{B_2} = r_B P$.

(iv) Upon receiving the standard signature $(\sigma_{B_1}, \sigma_{B_2})$ on m from the proxy signer. $\mathcal{A}_I$ aborts if $e(\sigma_{B_1}, P) \neq e(H_0(ID_B), P_{pub})e(H_2(w, m), \sigma_{B_2})$.

(v) If both $(\sigma_{A_1}, \sigma_{A_2})$ and $(\sigma_{B_1}, \sigma_{B_2})$ are valid. $\mathcal{A}_I$ outputs a proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$ on message m with warrant w such that $\sigma_{M_1} = \sigma_{A_1} + \sigma_{B_1} = sk_{ID_A} + r_A H_1(w) + sk_{ID_B} + r_B H_2(w, m)$, $\sigma_{M_2} = \sigma_{A_2} = r_A P$ and $\sigma_{M_3} = \sigma_{B_2} = r_B P$.

It can be verified that $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$ is a valid proxy signature. Thus, the proposed identity-based proxy signature is insecure, since given a proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$, it might come from a malicious adversary. The proposed attack is a practical attack since a malicious adversary could launch such an attack without notice of both the original signer and the proxy signer. Besides the scheme mentioned in this paper, we have found that the proxy signature schemes in [18–21] are also subjected to this attack.

4. Security Model for Proxy Signature

4.1. Malicious Attackers. We revise the security model for identity-based proxy signature defined in [13] to capture the new attack in this section. In the security model for proxy signature, the capability of an adversary is modelled by its ability to query different oracles. Before we formally define each adversarial game, we first introduce four types of oracle queries that will appear in the models:

(i) **Key extract query**: $\mathcal{A}$ can query an identity $ID \in \mathcal{ID}$, where $\mathcal{ID}$ represents the identity space, to the key extract oracle $\mathcal{O}_{KE}(\cdot)$. The corresponding key sk_{ID} is then generated and returned to $\mathcal{A}$.

(ii) **Original signer's standard signing query**: $\mathcal{A}$ can query the original signer's signing oracle $\mathcal{O}_{OS'S}(\cdot)$ with any warrant $w \in \mathcal{W}$ under the original signer's identity $ID \in \mathcal{ID}$, where $\mathcal{W}$ represents the warrant space. The private key sk_{ID} on identity ID is generated using the key extraction algorithm. The corresponding original signer's signature σ_o on warrant w is generated and returned to $\mathcal{A}$.

(iii) **Proxy signing query**: $\mathcal{A}$ can query the proxy signing oracle $\mathcal{O}_{PS}(\cdot)$ with any message $m \in \mathcal{M}$ with warrant $w \in \mathcal{W}$ of his choice under the original signer's identity ID_A and the proxy signer's identity ID_B such that $ID_A, ID_B \in \mathcal{ID}$, where $\mathcal{M}$ represents the message space. The private keys sk_{ID_A} and sk_{ID_B} on identities ID_A, ID_B are generated using the key extraction algorithm. A valid proxy signature on m is then generated and returned to $\mathcal{A}$.

(iv) **Proxy signer's signing query**: $\mathcal{A}$ can query the standard signature with any message $m \in \mathcal{M}$ of his choice to the proxy signer's standard signing oracle $\mathcal{O}_{PS'S}(\cdot)$. A valid standard signature of the proxy signer σ_p on m under the proxy signer's identity is then generated and returned to $\mathcal{A}$.

According to the information held by an attacker, three different types of adversaries are defined:

(1) $\mathcal{A}_I$: an outsider attacker who only has the identities of the original signer and the proxy signer that aims to forge a valid proxy signature.

(2) $\mathcal{A}_{II}$: a malicious proxy signer who possesses the private key sk_{ID_B} of the proxy signer and the identity of the original signer, and tries to forge a valid proxy signature σ without knowledge of the private key sk_{ID_A} of the original signer.

(3) $\mathcal{A}_{III}$: a malicious original signer that possesses the private key sk_{ID_A} of the original signer and the identity ID_B of the proxy signer, and tries to forge a valid proxy signature σ without knowing the private key sk_{ID_B} of the proxy signer.

4.2. Adversarial Game with a Malicious Outsider Adversary $\mathcal{A}_I$. We first define the adversarial game between a malicious outsider adversary $\mathcal{A}_I$ and a simulator $\mathcal{S}$ as follows:

(i) **Setup**: the simulator $\mathcal{S}$ runs **Setup** algorithm to generate the *params* and *MSK* and sends *params* to $\mathcal{A}_I$ as well as keeping *MSK* secret.

(ii) **Original signer's standard signing queries**: $\mathcal{A}_I$ can choose any warrant $w \in \mathcal{W}$ with the original signer's identity ID_A and queries the original signer's standard signing oracle $\mathcal{O}_{OS'S}$. $\mathcal{S}$ generates the private key sk_{ID_A} using the key extract algorithm $sk_{ID_A} \longleftarrow$ **KeyExtract**$(MSK, ID_A, params)$; then $\mathcal{S}$ generates the delegation information $\sigma_o \longleftarrow$ **StandardSign**$(sk_{ID_A}, w, params)$ and sends σ_o to $\mathcal{A}_I$.

(iii) **Proxy Signer's Standard Signature Queries**: $\mathcal{A}_I$ queries the proxy signer's standard signing oracle $\mathcal{O}_{PS'S}$ with a message $m \in \mathcal{M}$ of his choice under the proxy signer's identity $ID_B \in \mathcal{ID}$. $\mathcal{S}$ generates the private key sk_{ID_B} using the key extract algorithm $sk_{ID_B} \longleftarrow$ **KeyExtract**$(MSK, ID_B, params)$; then $\mathcal{S}$ generates the standard signature $s\sigma \longleftarrow$ **StandardSign**$(sk_{ID_B}, m, params)$ and sends $s\sigma$ to $\mathcal{A}_I$.

(iv) **Forgery Phase**: finally, $\mathcal{A}_I$ outputs a proxy signature σ^* on message M^* for a warrant W^* with the original signer's identity ID_A and the proxy signer's identity ID_B.

We say $\mathcal{A}_{II}$ wins the game if

(i) **ProVer**$(\sigma^*, ID_A, ID_B, W^*, M^*) = 1$;

(ii) (W^*, ID_A) has been queried to the original signer's standard signing oracle $\mathcal{O}_{OS'S}$;

(iii) (W^*, M^*, ID_B) has been queried to the proxy signer's standard signing oracle $\mathcal{O}_{PS'S}$.

Define the advantage of a malicious adversary $\mathcal{A}_I$ in winning the game as

$$Adv_{\mathcal{A}_I}(\kappa) = \Pr\left[\mathcal{A}_I \text{ Wins the game}\right]. \tag{3}$$

Definition 2. We say an identity-based proxy signature scheme is secure against an outsider adversary $\mathcal{A}_I$ if for any probabilistic polynomial time $\mathcal{A}_I$, $Adv_{\mathcal{A}_I}(\kappa)$ is negligible in κ.

4.3. Adversarial Game with a Malicious Proxy Signer $\mathcal{A}_{II}$. We first define the adversarial game between a malicious proxy signer $\mathcal{A}_{II}$ and a simulator $\mathcal{S}$ as follows:

(i) **Setup**: the simulator $\mathcal{S}$ runs **Setup** algorithm to generate the *params* and *MSK* and sends *params* to $\mathcal{A}_{II}$ as well as keeping *MSK* secret.

(ii) **Key extract queries**: $\mathcal{A}_{II}$ selects an identity ID such that $ID \in \mathcal{ID}$, the simulator $\mathcal{S}$ runs $sk_{ID} \longleftarrow$ **KeyExtract**$(MSK, ID, params)$ and returns sk_{ID} to $\mathcal{A}_{II}$.

(iii) **Original signer's standard signing queries**: $\mathcal{A}_{II}$ can choose any warrant $w \in \mathcal{W}$ with an identity $ID \in \mathcal{ID}$ and queries original signer's standard signing oracle $\mathcal{O}_{OS'S}$. $\mathcal{S}$ generates the private key sk_{ID} using the key extract algorithm $sk_{ID} \longleftarrow$ **KeyExtract**$(MSK, ID, params)$; then $\mathcal{S}$ generates the

original signer's standard signature $\sigma_o \longleftarrow$ **StandardSign**$(sk_{ID}, w, params)$ and sends σ_o to $\mathscr{A}_{II}$.

(iv) **Proxy signing queries**: $\mathscr{A}_{II}$ chooses a warrant $w \in \mathscr{W}$ and a message $m \in \mathscr{M}$ and queries the proxy signing oracle $\mathscr{O}_{PS}$ with the original signer's identity ID_1 and the proxy signer's identity ID_2. $\mathscr{S}$ generates

$$\begin{aligned} &sk_{ID_1}, sk_{ID_2} \\ &\quad \longleftarrow \textbf{KeyExtract}\left(MSK, ID_1, ID_2, params\right) \\ &\sigma_w \longleftarrow \textbf{DelegationGen}\left(sk_{ID_1}, w, params\right), \\ &\sigma \longleftarrow \textbf{ProSign}\left(\sigma_w, sk_{ID_2}, m, params\right) \end{aligned} \tag{4}$$

and returns σ to $\mathscr{A}_{II}$.

(v) **Forgery Phase**: finally, $\mathscr{A}$ outputs a proxy signature σ^* on message M^* for a warrant W^* with the original signer's identity ID_A and the proxy signer's identity ID_B.

We say $\mathscr{A}_{II}$ wins the game if

(i) **ProVer**$(\sigma^*, ID_A, ID_B, W^*, M^*) = 1$;

(ii) ID_A has not been queried to the key extraction oracle $\mathscr{O}_{KE}(\cdot)$;

(iii) (W^*, ID_A) has not been queried to the delegation oracle $\mathscr{O}_{DG}$;

(iv) (W^*, M^*, ID_A, ID_B) has not been queried to the proxy signing oracle $\mathscr{O}_{PS}$.

Define the advantage of a malicious adversary $\mathscr{A}_{II}$ in winning the game as

$$Adv_{\mathscr{A}_{II}}(\kappa) = \Pr\left[\mathscr{A}_{II} \text{ Wins the game}\right]. \tag{5}$$

Definition 3. We say an identity-based proxy signature scheme is secure against the $\mathscr{A}_{II}$ under chosen identity and warrant attacks if for any probabilistic polynomial time $\mathscr{A}_{II}$, $Adv_{\mathscr{A}_{II}}(\kappa)$ is negligible in κ.

4.4. Adversarial Game with Malicious Original Signer. The adversarial game between a malicious original signer $\mathscr{A}_{III}$ and a simulator $\mathscr{S}$ is defined as follows:

(i) **Setup**, **Key Extract Queries** and **Proxy Signing Queries** are the same as those in the adversarial game against a malicious proxy signer.

(ii) **Proxy Signer's Standard Signature Queries**: $\mathscr{A}_{III}$ queries the proxy signer's standard signing oracle $\mathscr{O}_{ps's}$ with a message $m \in \mathscr{M}$ of his choice under an identity $ID \in \mathscr{ID}$. $\mathscr{S}$ generates the private key sk_{ID} using the key extract algorithm $sk_{ID} \longleftarrow$ **KeyExtract**$(MSK, ID, params)$; then $\mathscr{S}$ generates the standard signature $\sigma_p \longleftarrow$ **StandardSign**$(sk_{ID}, m, params)$ and sends σ_p to $\mathscr{A}_{III}$.

(iii) **Forgery Phase**: finally, $\mathscr{A}_{III}$ outputs a proxy signature σ^* on message M^* for a warrant W^* with the original signer's identity ID_A and the proxy signer's identity ID_B.

We say $\mathscr{A}_{III}$ wins the game if

(i) **ProVer**$(\sigma^*, ID_A, ID_B, W^*, M^*) = 1$;

(ii) ID_B has not been queried to the key extraction oracle $\mathscr{O}_{KE}$;

(iii) (W^*, M^*, ID_B) has not been queried to the proxy signer's standard signing oracle $\mathscr{O}_{PS'S}$;

(iv) (W^*, M^*, ID_A, ID_B) has not been queried to the proxy signing oracle $\mathscr{O}_{PS}$.

Define the advantage of a malicious adversary $\mathscr{A}_{III}$ in winning the game as

$$Adv_{\mathscr{A}_{III}}(\kappa) = \Pr\left[\mathscr{A}_{III} \text{ Wins the game}\right]. \tag{6}$$

Definition 4. We say an identity-based proxy signature scheme is secure against the $\mathscr{A}_{III}$ under chosen identity and message attacks if for any probabilistic polynomial time $\mathscr{A}_{III}$, $Adv_{\mathscr{A}_{III}}(\kappa)$ is negligible in κ.

5. The Revised Identity-Based Proxy Signature Scheme

We present the revised ID-based proxy signature scheme that efficiently thwarts the proposed attack in this section.

(1) **Setup**: let $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ be a bilinear pairing map, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are of prime order q. Let P be a generator of $\mathbb{G}_1$. Choose a random number $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$. Select three collision-resistant hash functions H_0, H_1, H_2 such that $H_0, H_1, H_2 : \{0,1\}^* \longrightarrow \mathbb{G}_1$. The system parameters $params = \{e, \mathbb{G}_1, \mathbb{G}_2, q, P_{pub}, H_0, H_1, H_2\}$, the master secret key $Msk = s$.

(2) **KeyExtract**: on input a user's identity ID, output the secret key for this identity $sk_{ID} = sH_0(ID)$.

(3) **StandardSign**: on input a message m, the standard signature on m under identity ID is $\sigma = (\sigma_1, \sigma_2)$ such that $\sigma_1 = sk_{ID} + rH_1(m)$ and $\sigma_2 = rP$, where $r \in \mathbb{Z}_q^*$.

(4) **StandardVer**: on input a standard signature $\sigma = (\sigma_1, \sigma_2)$ of message m under identity ID, output "1" if $e(\sigma_1, P) = e(H_0(ID), P_{pub})e(H_1(m), \sigma_2)$; otherwise, output "0".

(5) **DelegationGen**: let w be a warrant that includes the delegation information such as the identities of the original signer and the designated proxy signer, the delegation period, the types of messages that a proxy signer can sign, and so on. Then the original signer with identity ID_A generates the delegation information $\sigma_w = (\sigma_{W_1}, \sigma_{W_2})$ such that $\sigma_{W_1} = sk_{ID_A} + r_A H_1(w)$ and $\sigma_{W_2} = r_A P$, where $r_A \in \mathbb{Z}_q$. The original signer sends the delegation information σ_w to the proxy signer.

(6) **ProSign**: upon receiving the delegation information $\sigma_w = (\sigma_{W_1}, \sigma_{W_2})$ and w from the original signer, the proxy signer with identity ID_B generates a proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3}$ on a message m such that $\sigma_{M_1} = \sigma_{W_1} + sk_{ID_B} + r_B H_2(w, m) + r_B H_1(w)$, $\sigma_{M_2} = \sigma_{W_2} + r_B P$, $\sigma_{M_3} = r_B P$.

(7) **ProVer**: on input the identities ID_A, ID_B of the original signer and proxy signer, a warrant w and a message m and the proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$, outputs "1" if $e(\sigma_{M_1}, P) = e(H_0(ID_A), P_{pub})e(H_1(w)), \sigma_{M_2}) \cdot e(H_0(ID_B), P_{pub})e(H_2(w, m), \sigma_{M_3})$. Otherwise, output "0".

6. Security Analysis

In this section, we analyse the security of the revised ID-based proxy signature scheme against $\mathscr{A}_I$, $\mathscr{A}_{II}$, and $\mathscr{A}_{III}$ adversaries.

Theorem 5. *The revised ID-based proxy signature scheme is secure against an outsider adversary $\mathscr{A}_I$ if the CDH assumption holds.*

Proof. The proof is by contradiction under the random oracle model. Suppose there exists an outsider adversary $\mathscr{A}_I$ that has a nonnegligible advantage ϵ in attacking the proposed scheme; then we can build another algorithm $\mathscr{B}$ that uses $\mathscr{A}_I$ to solve the CDH problem. Let $\mathbb{G}_1$ be a bilinear pairing group of prime order q; $\mathscr{B}$ is given $P, aP, bP \in \mathbb{G}_1$ which is a random instance of the CDH problem. Its goal is to compute abP. Algorithm $\mathscr{B}$ will simulate the challenger and interact with the forger $\mathscr{A}_I$ as described below.

(1) **Setup**: $\mathscr{B}$ selects a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ where $\mathbb{G}_1$ and $\mathbb{G}_2$ are of prime order q. $\mathscr{B}$ chooses a generator P of $\mathbb{G}_1$. Let (P, aP, bP) be the inputs of the CDH problem. $\mathscr{B}$ sets the master public key $P_{pub} = sP$, where $s \in \mathbb{Z}_q^*$. $\mathscr{B}$ selects three collision-resistant hash functions $H_0, H_1, H_2 : \{0,1\}^* \longrightarrow \mathbb{G}_1$. $\mathscr{B}$ sends $(e, \mathbb{G}_1, \mathbb{G}_2, q, P, P_{pub}, H_0, H_1, H_2)$ to $\mathscr{A}_{II}$.

(2) **Hash queries**: in the security proof, the hash functions H_0, H_1, H_2 are modelled as random oracles. We regard the identity, warrant, and message queries as H_0, H_1, and H_2 queries, respectively. Assume $\mathscr{B}$ keeps hash tables T_0, T_1, and T_2 for these queries.

(a) H_0 **Query**: for each query on identity ID_i, if ID_i has existed in T_0, the same value $H_0(ID_i)$ is returned to $\mathscr{A}_{II}$. Otherwise, $\mathscr{B}$ chooses a random $c_i \in \mathbb{Z}_q$ and sets $H_0(ID_i) = c_i P$. $\mathscr{B}$ sends $c_i P$ to $\mathscr{A}_I$ as well as stores $(ID_i, c_i, H_0(ID_i))$ to T_0.

(b) H_1 **Query**: assume $\mathscr{A}_I$ makes q_{H_1} warrant queries; $\mathscr{B}$ selects a random number $\beta \in (1, q_{H_1})$, for each query on warrant w_i such that $1 \le i \ne \beta \le q_{H_1}$; if w_i has existed in T_1, the same value $H_1(w_i)$ is returned to $\mathscr{A}_I$. Otherwise,

(i) if $w_i \ne w_\beta$, $\mathscr{B}$ chooses a random $k_i \in \mathbb{Z}_q$ and sets $H_1(w_\beta) = k_i P$. $\mathscr{B}$ sends $H_1(w_\beta)$ to $\mathscr{A}_I$ as well as storing $(w_\beta, k_i, H_1(w_\beta))$ to T_1.

(ii) If $w_i = w_\beta$, $\mathscr{B}$ sets $H_1(w_\beta) = aP$. $\mathscr{B}$ sends $H_1(w_\beta)$ to $\mathscr{A}_I$.

(c) H_2 **Query**: for each query on message m_i accompanying with a warrant w_i, if $H_2(w_i, m_i)$ has existed in T_2, the same value $H_2(w_i, m_i)$ is returned to $\mathscr{A}_I$. Otherwise, $\mathscr{B}$ chooses a random $u_i \in \mathbb{Z}_q$ and sets $H_2(w_i, m_i) = u_i P$. $\mathscr{B}$ sends $H_2(w_i, m_i)$ to $\mathscr{A}_I$ as well as storing $((w_i, m_i), u_i, H_2(w_i, \mathrm{m}_i))$ to T_2.

(3) **Original signer's standard signing queries**: $\mathscr{A}_I$ can query the original signer's standard signature on a warrant w_i. Assume $\mathscr{A}_I$ makes $q_{os's}$ queries with the original signer's identity ID_A, for each query on w_i, assume $H_0(ID_A)$ and $H_1(w_i)$ have existed in T_0 and T_1; if they are not the cases, $\mathscr{B}$ performs the above algorithms to assign values for $H_0(ID_A)$ and $H_1(w_i)$. Assume $H_0(ID_A) = c_A P$, $\mathscr{B}$ simulates as follows:

(i) If $w_i \ne w_\beta$, assume $H_1(w_i) = k_i P$; then $\mathscr{B}$ chooses randomly $r_{A_i} \in \mathbb{Z}_q$ and sets $\sigma_{w_i} = (\sigma_{w_{i1}}, \sigma_{w_{i2}})$ such that $\sigma_{w_{i1}} = c_A sP + r_{A_i} k_i P = sH_0(ID_A) + r_{A_i} H_1(w_i)$ and $\sigma_{w_{i2}} = r_{A_i} P$.

(ii) If $w_i = w_\beta$, then $\mathscr{B}$ chooses randomly $r_{A_\beta} \in \mathbb{Z}_q$ and sets $\sigma_\beta = (\sigma_{w_{\beta 1}}, \sigma_{2_{\beta 2}})$ such that $\sigma_{w_{\beta 1}} = c_A sP + r_{A_\beta} bP = sH_0(ID_A) + r_{A_\beta} H_1(w_\beta)$ and $\sigma_{w_{\beta 2}} = r_{A_\beta} P$.

(4) **Proxy signer's standard signing queries**: assume $\mathscr{A}_I$ makes $q_{ps's}$ standard signature queries under the proxy signer's identity ID_B. For each query on $M_i = w_i \parallel m_i$, assume $H_0(ID_B)$ and $H_2(M_i)$ have existed in T_0 and T_2; if they are not the cases, $\mathscr{B}$ performs the above algorithms to assign values for $H_0(ID_A)$ and $H_2(M_i)$. Assume $H_0(ID_B) = c_B P$; $\mathscr{B}$ chooses a number $\delta \in (1, q_{ps's})$ and simulates as follows:

(i) If $M_i \ne M_\delta$, assume $H_2(M_2) = u_i P$; then $\mathscr{B}$ chooses randomly $r_{B_i} \in \mathbb{Z}_q$ and sets $\sigma_{p_i} = (\sigma_{p_{i1}}, \sigma_{p_{i2}})$ such that $\sigma_{p_{i1}} = c_B sP + r_{B_i} k_i P = sH_0(ID_B) + r_{B_i} H_2(M_i)$ and $\sigma_{p_{i1}} = r_{B_i} P$.

(ii) If $M_i = M_\delta$, assume $H_2(M_\delta) = u_\delta P$; then $\mathscr{B}$ sets $dsk_\delta = (\sigma_{B1_\delta}, \sigma_{B2_\delta})$ such that $\sigma_{B1_\delta} = c_B sP + bu_\delta P = sH_0(ID_B) + bH_2(M_\delta)$ and $\sigma_{B2_i} = bP$.

(5) **Forgery**: assume $\mathscr{A}_I$ outputs a valid proxy signature $\sigma^* = (\sigma^*_{M_1}, \sigma^*_{M_2}, \sigma^*_{M_3})$ on message M^* under a warrant W^* with the proxy signer's identity ID_A and the proxy signer's identity ID_B. Besides,

(i) (ID_A, W^*) has been queried in the original signer's standard signing queries;

(ii) (ID_B, W^*, M^*) has been queried in the proxy signer's standard signing queries.

If $W^* \ne w_\beta$ or $M^* \ne M_\delta$, $\mathscr{B}$ will abort. Otherwise, given the forged proxy signature $\sigma^* = (\sigma^*_{M_1}, \sigma^*_{M_2}, \sigma^*_{M_3})$. $\mathscr{B}$ can solve the CDH problem

$$abP = \sigma_{M_1^*} - \sigma_{A1_\beta} - \sigma_{B1_\delta} \tag{7}$$

$\mathscr{B}$ will not abort when $W^* = w_\beta$ and $M^* = M_\delta$. Thus, if there exists an outsider adversary $\mathscr{A}_I$ that has a nonnegligible probability ϵ in breading the proposed identity-based proxy signature scheme, then there exists another probabilistic polynomial time algorithm $\mathscr{B}$ that has a probability

$$Succ^{CDH}_{\mathscr{B},\mathbb{G}_1} = \frac{\epsilon}{q_{os's} \cdot q_{ps's}} \tag{8}$$

which is nonnegligible. Thus, we reach a contradiction. □

Theorem 6. *The revised ID-based proxy signature scheme is secure against the $\mathscr{A}_{II}$ chosen identity and chosen warrant attacks if the CDH assumption holds.*

Proof. Let us recall the definition of $\mathscr{A}_{II}$; $\mathscr{A}_{II}$ is a malicious proxy signer possessing the private key of the proxy signer. With this in mind, the simulation is as follows:

(1) **Setup**: $\mathscr{B}$ selects a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are of prime order q. $\mathscr{B}$ chooses a generator P of $\mathbb{G}_1$. Let (P, aP, bP) be the inputs of the CDH problem. $\mathscr{B}$ sets the master public key $P_{pub} = aP$. $\mathscr{B}$ selects three collision-resistant hash functions $H_0, H_1, H_2 : \{0,1\}^* \longrightarrow \mathbb{G}_1$. $\mathscr{B}$ sends $(e, \mathbb{G}_1, \mathbb{G}_2, q, P, P_{pub}, H_0, H_1, H_2)$ to $\mathscr{A}_{II}$.

(2) **Hash queries**: regard the identity, warrant, and message queries as H_0, H_1, and H_2 queries, respectively. $\mathscr{B}$ keeps hash tables T_0, T_1, and T_2 for these queries.

(a) H_0 **Query**: assume $\mathscr{A}_{II}$ makes q_{H_0} identity queries, choose $\alpha \in (1, q_{H_0})$, for each query on identity ID_i such that $1 \le i \ne \alpha \le q_{H_0}$, if ID_i has existed in T_0, the same value $H_0(ID_i)$ is returned to $\mathscr{A}_{II}$. Otherwise,

(i) If $i \ne \alpha$, $\mathscr{B}$ chooses a random $c_i \in \mathbb{Z}_q$ and sets $H_0(ID_i) = c_iP$. $\mathscr{B}$ sends c_iP to $\mathscr{A}_{II}$ as well as storing $(ID_i, c_i, H_0(ID_i))$ to T_0.

(ii) If $i = \alpha$, $\mathscr{B}$ sets $H_0(ID_\alpha) = bP + c_\alpha P$, where $c_\alpha \in \mathbb{Z}_q$ and returns $H_0(ID_I)$ to $\mathscr{A}_{II}$. $\mathscr{B}$ adds $(ID_\alpha, c_\alpha, H_0(ID_\alpha))$ to T_0.

(b) H_1 **Query**: assume $\mathscr{A}_{II}$ makes q_{H_1} warrant queries; $\mathscr{B}$ selects a random number $\beta \in (1, q_{H_1})$, for each query on warrant w_i such that $1 \le i \ne \beta \le q_{H_1}$, if w_i has existed in T_1, the same value $H_1(w_i)$ is returned to $\mathscr{A}_{II}$. Otherwise,

(i) if $w_i \ne w_\beta|_{ID_\alpha \longrightarrow o}$, which means ID_α is included in w_i and the user with identity ID_α plays the role of original signer in the system. $\mathscr{B}$ chooses a random $k_i \in \mathbb{Z}_q$ and sets $H_1(w_i) = k_iP - bP$. $\mathscr{B}$ sends $H_1(w_i)$ to $\mathscr{A}_{II}$ as well as storing $(w_i, b_i, H_1(w_i))$ to T_1;

(ii) if $w_i \ne w_\beta|_{ID_\alpha \longrightarrow p}$, which means ID_α is included in w_i and the user with identity ID_α plays the role of proxy signer in the system. $\mathscr{B}$ chooses a random $k_i \in \mathbb{Z}_q$ and sets $H_1(w_i) = k_iP$. $\mathscr{B}$ sends $H_1(w_i)$ to $\mathscr{A}_{II}$ as well as stores $(w_i, k_i, H_1(w_i))$ to T_1;

(iii) if $w_i = w_\beta$, $\mathscr{B}$ chooses a random $k_i \in \mathbb{Z}_q$ and sets $H_1(w_\beta) = k_iP$. $\mathscr{B}$ sends $H_1(w_\beta)$ to $\mathscr{A}_{II}$ as well as storing $(w_\beta, k_i, H_1(w_\beta))$ to T_1.

(c) H_2 **Query**: assume $\mathscr{A}_{II}$ makes q_{H_2} message queries, $\mathscr{B}$ selects a random number $\delta \in (1, q_{H_1})$, for each query on message m_i accompanying with a warrant w_i such that $1 \le i \ne \delta \le q_{H_2}$, if $H_2(w_i, m_i)$ has existed in T_2, the same value $H_2(w_i, m_i)$ is returned to $\mathscr{A}_{II}$. Otherwise,

(i) if $w_i \ne w_\beta, m_i \ne m_\delta$, $\mathscr{B}$ chooses a random $u_i \in \mathbb{Z}_q$ and sets $H_2(w_i, m_i) = u_iP + aP$. $\mathscr{B}$ sends $H_2(w_i, m_i)$ to $\mathscr{A}_{II}$ as well as storing $((w_i, m_i), c_i, H_2(w_i, m_i))$ to T_2;

(ii) if $w_i = w_\beta, m_i \ne m_\delta$, the same as the case when $w_i \ne w_\beta, m_i \ne m_\delta$;

(iii) if $w_i \ne w_\beta, m_i = m_\delta$, the same as the case when $w_i \ne w_\beta, m_i \ne m_\delta$;

(iv) if $w_i = w_\beta, m_i = m_\delta$, $\mathscr{B}$ chooses a random $u_i \in \mathbb{Z}_q$ and sets $H_2(w_\beta, m_\delta) = u_iP$. $\mathscr{B}$ sends $H_2(w_\beta, m_\beta)$ to $\mathscr{A}_{II}$ as well as storing $((w_\beta, m_\delta), u_i, H_2(w_\beta, m_\delta))$ to T_2.

(3) **Key extraction queries**: $\mathscr{A}_{II}$ can make key extraction queries on any identity $ID \in \mathscr{ID}$ such that $ID \ne ID_\alpha$. If $\mathscr{A}_{II}$ makes key extraction query on identity ID_α, $\mathscr{B}$ just terminates the simulation and reports a failure. Assume $\mathscr{A}_{II}$ makes q_k key extractions queries, for each query on identity ID_i for $1 \le i \le q_k$.

(i) If ID_i has existed in table T_0, assume $H_0(ID_i) = c_iP$; then $\mathscr{B}$ returns $sk_{ID_i} = c_iaP = aH_0(ID_i)$ to $\mathscr{A}_{II}$.

(ii) Otherwise, $\mathscr{B}$ chooses a random $c_i \in \mathbb{Z}_q$ and sets $H_0(ID_i) = c_iP$. $\mathscr{B}$ returns $sk_{ID_i} = c_iaP$ to $\mathscr{A}_{II}$ and adds $(ID_i, c_i, H_0(ID_i))$ to T_0.

(4) **Original signer's standard signing queries**: $\mathscr{A}_{II}$ can query original signer's standard signature on a warrant $w_i \in \mathscr{W}$ under an identity $ID_i \in \mathscr{ID}$. Assume $\mathscr{A}_{II}$ makes $q_{os's}$ original signer's standard signing queries. For each query, assume ID_i and w_i have been submitted to the H_0 and H_1 queries, respectively. If they are not the cases, $\mathscr{B}$ performs the above algorithms to set values for $H_0(ID_i)$ and $H_1(w_i)$; then $\mathscr{B}$ simulates σ_{w_i} as follows:

(i) If $ID_i \ne ID_\alpha$ and $w_i \ne w_\beta|_{ID_\alpha \longrightarrow o}$, assume $H_0(ID_i) = c_iP$ and $H_1(w_i) = k_iP - bP$, respectively; then $\mathscr{B}$ chooses a random $r_i \in \mathbb{Z}_q$ and returns the original signer's standard signature $\sigma_{w_i} = (\sigma w_{i1}, \sigma w_{i2})$ such that $\sigma w_{i1} = c_iP_{pub} + r_i(k_iP - bP) = sk_{ID_i} + r_iH_1(w_i)$ and $\sigma w_{i2} = r_iP$ and to $\mathscr{A}_{II}$.

(ii) If $ID_i \ne ID_\alpha$ and $w_i \ne w_\beta|_{ID_\alpha \longrightarrow p}$, assume $H_0(ID_i) = c_iP$ and $H_1(w_i) = k_iP$, respectively; then $\mathscr{B}$ chooses a random $r_i \in \mathbb{Z}_q$ and returns original signer's standard signature $\sigma_{w_i} = (\sigma w_{i1}, \sigma w_{i2})$ such that $\sigma w_{i1} = c_iP_{pub} + r_ik_iP = sk_{ID_i} + r_iH_1(w_i)$ and $\sigma w_{i2} = r_iP$ to $\mathscr{A}_{II}$.

(iii) If $ID_i = ID_\alpha$ and $w_i \neq w_\beta|_{ID_\alpha \longrightarrow o}$, assume $H_0(ID_i) = bP + c_iP$ and $H_1(w_i) = k_iP - bP$, respectively; then $\mathscr{B}$ simulates the original signer's standard signature $\sigma_{w_i} = (\sigma w_{i1}, \sigma w_{i2})$ by setting $\sigma w_{i2} = r_iP = l_iP + aP$, where $l_i \in_R \mathbb{Z}_q^*$ and $\sigma w_{i1} = (c_i + k_i)P_{pub} + k_il_iP - l_ibP$. It can be verified that $(\sigma w_{i1}, \sigma w_{i2})$ is a correct simulation since

$$\begin{aligned}\sigma w_{i1} &= (c_i + k_i) P_{pub} + k_il_iP - l_ibP \\ &= abP + c_iaP + k_iaP + k_il_iP - l_ibP - abP \\ &= a(c_iP + bP) + (a + l_i)(k_iP - bP) \\ &= aH_0(ID_\alpha) + r_iH_1(w_i)\end{aligned} \tag{9}$$

(iv) If $ID_i = ID_\alpha$ and $w_i \neq w_\beta|_{ID_\alpha \longrightarrow p}$, since we do not consider self-delegation in our scheme, then $\mathscr{B}$ just terminates the simulation and reports failure.

(v) If $ID_i = ID_\alpha$ and $w_i = w_\beta$, $\mathscr{B}$ terminates the simulation and reports failure.

(5) **Proxy signing queries**: $\mathscr{A}_{II}$ can query a proxy signature on a message $m_i \in \mathscr{M}$ under a warrant $w_i \in \mathscr{W}$ with the proxy signer's identity ID_{1_i} and the original signer's identity ID_{2_i} such that $ID_{1_i}, ID_{2_i} \in \mathscr{ID}$. Assume ID_{1_i}, ID_{2_i} have been submitted to the H_0 query and w_i and $w_i \parallel m_i$ have been submitted to the H_1 and H_2 queries, respectively. If they are not the cases, the above algorithms will be performed to assign new values $H_0(ID_{1_i})$, $H_0(ID_{2_i})$, $H_1(w_i)$, and $H_2(w_i, m_i)$. Assume $\mathscr{A}_{II}$ makes q_{ps} proxy signing queries. For each queries on a message m_i with warrant w_i such that $1 \leq i \leq q_{ps}$, $\mathscr{B}$ simulates the corresponding proxy signature as follows:

(a) If $ID_{1_i} \neq ID_\alpha$, $ID_{2_i} \neq ID_\alpha$ assume $H_0(ID_{1_i}) = c_{1_i}P$, $H_0(ID_{2_i}) = c_{2_i}P$; then $\mathscr{B}$ chooses two random numbers $r_{1_i}, r_{2_i} \in \mathbb{Z}_q^*$ and returns the proxy signature $\sigma_i = (\sigma_{M_{i_1}}, \sigma_{M_{i_2}}, \sigma_{M_{i_3}})$ such that $\sigma_{M_{i1}} = c_{1_i}aP + r_{1_i}H_1(w_i) + c_{2_i}aP + r_{2_i}H_2(w_i, m_i) + r_{2_i}H_2(m_i)$, $\sigma_{M_{i_2}} = (r_{1_i} + r_{2_i})P$ and $\sigma_{M_{i3}} = r_{2_i}P$ to $\mathscr{A}_{II}$. It is a correct simulation since

$$\begin{aligned}&e\left(\sigma_{M_{i1}}, P\right) = e\left(c_{1_i}aP + r_{1_i}H_1(w_i) + c_{2_i}aP\right. \\ &\left. + r_{2_i}H_2(w_i, m_i) + r_{2_i}H_1(w_i), P\right) = e\left(c_{1_i}P, aP\right) \\ &\cdot e\left(H_1(w_i), \left(r_{1_i} + r_{2_i}\right)P\right) e\left(c_{2_i}P, aP\right) \\ &\cdot e\left(H_2(m_i, w_i), r_{2_i}P\right) = e\left(H_0\left(ID_{1_i}\right), P_{pub}\right) \\ &\cdot e\left(H_1(w_i), \sigma_{M_{i_2}}\right) e\left(H_0\left(ID_{2_i}\right), P_{pub}\right) \\ &\cdot e\left(H_2(m_i, w_i), \sigma_{M_{i_3}}\right)\end{aligned} \tag{10}$$

(b) If $ID_{1_i} \neq ID_\alpha$, $ID_{2_i} = ID_\alpha$, assume $H_0(ID_{1_i}) = c_{1_i}P$, $H_0(ID_{2_i}) = c_\alpha P + bP$; then

(i) If $w_i \neq w_\beta|_{ID_\alpha \longrightarrow o}, m_i \neq m_\delta$ or $w_i \neq w_\beta|_{ID_\alpha \longrightarrow o}, m_i = m_\delta$, $\mathscr{B}$ terminates the simulation and reports failure.

(ii) If $w_i \neq w_\beta|_{ID_\alpha \longrightarrow p}$ and $m_i \neq m_\delta$, assume $H_1(w_i) = k_iP$ and $H_2(w_i, m_i) = u_iP + aP$; $\mathscr{B}$ simulates the proxy signature $\sigma_i = (\sigma_{M_{i_1}}, \sigma_{M_{i_2}}, \sigma_{M_{i_3}})$ by setting $\sigma_{M_{i_3}} = r_{2_i}P = v_iP - bP$, $\sigma_{M_{i_2}} = r_{1_i}P + v_iP - bP$ and $\sigma_{M_{i_1}} = (c_{1_i} + c_\alpha + v_i)P_{pub} + r_{1_i}H_2(w_i) + k_i(v_iP - bP) + u_i(v_iP - bP)$, where $v_i, r_{1_i} \in \mathbb{Z}_q$. It can be verified that it is a correct simulation since

$$\begin{aligned}&e\left(\sigma_{M_{i1}}, P\right) = e\left(\left(c_{1_i} + c_\alpha + v_i\right)P_{pub} + r_{1_i}H_2(w_i)\right. \\ &\left. + k_i(v_iP - bP) + u_i(v_iP - bP), P\right) = e\left(c_{1_i}P, aP\right) \\ &\cdot e\left(H_1(w_i), \left(r_{1_i} + r_{2_i}\right)P\right) e\left(abP + c_\alpha aP + v_iaP\right. \\ &\left. + u_iv_iP - u_ibP - abP, P\right) = e\left(c_{1_i}P, aP\right) \\ &\cdot e\left(H_1(w_i), \left(r_{1_i} + r_{2_i}\right)P\right) e\left(a\left(bP + c_\alpha P\right), P\right) \\ &\cdot e\left(\left(v_i - b\right)\left(u_iP + aP\right), P\right) = e\left(H_0\left(ID_{1_i}\right), P_{pub}\right) \\ &\cdot e\left(H_1(w_i), \sigma_{M_{i_2}}\right) e\left(H_0\left(ID_{2_i}\right), P_{pub}\right) \\ &\cdot e\left(H_2(w_i, m_i), \sigma_{M_{i_3}}\right)\end{aligned} \tag{11}$$

(iii) If $w_i \neq w_\beta|_{ID_\alpha \longrightarrow p}, m_i = m_\delta$ or $w_i = w_\beta, m_i \neq m_\delta$, $\mathscr{B}$ performs the same as that in case (ii).

(iv) If $w_i = w_\beta$ and $m_i = m_\delta$, $\mathscr{B}$ terminates the simulation and reports failure.

(c) If $ID_{1_i} = ID_\alpha$, $ID_{2_i} \neq ID_\alpha$, assume $H_0(ID_{1_i}) = c_\alpha P + bP$, $H_0(ID_{2_i}) = c_{2_i}P$, then

(i) if $w_i \neq w_\beta|_{ID_\alpha \longrightarrow o}$ and $m_i \neq m_\delta$, assume $H_1(w_i) = k_iP - bP$ and $H_2(w_i, m_i) = u_iP + aP$. $\mathscr{B}$ chooses $l_i, r_{2_i} \in \mathbb{Z}_q^*$ and simulates the proxy signature $\sigma_i = (\sigma_{M_{i_1}}, \sigma_{M_{i_2}}, \sigma_{M_{i_3}})$ by setting $\sigma_{M_{i_3}} = r_{2_i}P$, $\sigma_{M_{i_2}} = v_iP - bP + r_{2_i}P$ and $\sigma_{M_{i_1}} = (c_\alpha + k_i + c_{2_i})P_{pub} + l_i(k_iP - bP) + r_{2_i}(k_iP - bP) + r_{2_i}(u_iP + aP)$. It is a correct simulation since

$$\begin{aligned}&e\left(\sigma_{M_{i1}}, P\right) = e\left(\left(c_\alpha + k_i + c_{2_i}\right)P_{pub} + l_i\left(k_iP - bP\right)\right. \\ &\left. + r_{2_i}\left(k_iP - bP\right) + r_{2_i}\left(u_iP + aP\right), P\right) = e\left(abP\right. \\ &\left. + ac_\alpha P + l_ik_iP - l_ibP + ak_iP - abp\right. \\ &\left. + r_{2_i}\left(k_iP - bP\right), P\right) e\left(c_{2_i}P_{pub}, P\right) \\ &\cdot e\left(r_{2_i}\left(u_iP + aP\right), P\right) = e\left(a\left(c_\alpha P + bP\right), P\right)\end{aligned}$$

$$\cdot e\left(\left(l_i + a + r_{2_i}\right)\left(k_i P - bP\right), P\right) e\left(c_{2_i} P, aP\right) e\left(u_i P + aP, r_{2_i} P\right) = e\left(H_0\left(ID_{1_i}\right), P_{pub}\right) \cdot e\left(H_1\left(w_i\right), \sigma_{M_{i_2}}\right) e\left(H_0\left(ID_{2_i}\right), P_{pub}\right) \cdot e\left(H_2\left(w_i, m_i\right), \sigma_{M_{i_3}}\right) \tag{12}$$

(ii) If $w_i \neq w_\beta|_{ID_\alpha \longrightarrow p}, m_i \neq m_\delta$ or $w_i \neq w_\beta|_{ID_\alpha \longrightarrow p}, m_i = m_\delta$, $\mathscr{B}$ terminates the simulation and reports failure.

(iii) If $w_i \neq w_\beta|_{ID_\alpha \longrightarrow 0}$ and $m_i = m_\delta$, assume $H_1(w_i) = k_i P - bP$ and $H_2(w_i, m_\beta) = u_i P + aP$; $\mathscr{B}$ performs the same as that in case (i).

(iv) If $w_i = w_\beta$ and $m_i \neq m_\delta$, assume $H_1(w_\beta) = k_i P$ and $H_2(w_\beta, m_i) = u_i P + aP$; $\mathscr{B}$ chooses $v_i, r_{1_i} \in \mathbb{Z}_q^*$ and simulates the proxy signature $\sigma_i = (\sigma_{M_{i_1}}, \sigma_{M_{i_2}}, \sigma_{M_{i_3}})$ by setting $\sigma_{M_{i_3}} = v_i P - bP$, $\sigma_{M_{i_2}} = v_i P - bP + r_{1_i} P$, and $\sigma_{M_{i_1}} = (c_\alpha + c_{2_i} + v_i) P_{pub} + r_{1_i} k_i P + k_i(v_i P - bP) + u_i(v_i P - bP)$. It is a correct simulation since

$$\begin{aligned} e\left(\sigma_{M_{i1}}, P\right) &= e\left(\left(c_\alpha + c_{2_i} + v_i\right) P_{pub} + r_{1_i} k_i P + k_i\left(v_i P - bP\right) + u_i\left(v_i P - bP\right), P\right) \\ &= e\left(\left(c_\alpha + b\right) aP + c_{2_i} aP + r_{1_i} k_i P + k_i\left(v_i P - bP\right) + \left(v_i - b\right) aP + u_i\left(v_i P - bP\right), P\right) = e\left(c_\alpha P + bP, aP\right) e\left(k_i P, r_{1_i} P + v_i P - bP\right) e\left(c_{2_i} P, aP\right) \\ &\cdot e\left(u_i P, v_i P - bP\right) = e\left(H_0\left(ID_{1_i}\right), P_{pub}\right) \cdot e\left(H_1\left(w_i\right), \sigma_{M_{i_2}}\right) e\left(H_0\left(ID_{2_i}\right), P_{pub}\right) \cdot e\left(H_2\left(w_i, m_i\right), \sigma_{M_{i_3}}\right) \end{aligned} \tag{13}$$

(v) If $w_i = w_\beta$ and $m_i = m_\delta$, $\mathscr{B}$ terminates the simulation and reports failure.

(d) If $ID_{1_i} = ID_\alpha$, $ID_{2_i} = ID_\alpha$, $\mathscr{B}$ terminates the simulation and reports failure.

(6) **Forgery**: assume $\mathscr{A}_{II}$ outputs a valid proxy signature $\sigma^* = (\sigma^*_{M_1}, \sigma^*_{M_2}, \sigma^*_{M_3})$ on message M^* under a warrant W^* with the proxy signer's identity ID_A and the proxy signer's identity ID_B. Besides,

(i) ID_A has not been queried in the key extraction queries,

(ii) (ID_A, W^*) has not been queried in the delegation queries,

(iii) (ID_A, ID_B, W^*, M^*) has not been queried in the proxy signing queries,

If $H_0(ID_A) \neq bP + c_\alpha P$ or $H_1(W^*) \neq k_\beta P$ or $H_2(W^*, M^*) \neq u_\delta P$, $\mathscr{B}$ will abort. Otherwise, given the forged proxy signature $\sigma^* = (\sigma^*_{M_1}, \sigma^*_{M_2}, \sigma^*_{M_3})$. $\mathscr{B}$ can solve the CDH problem

$$abP = \sigma^*_{M_1} - c_\alpha aP - k_\beta \sigma^*_{M_2} - c_{2_i} aP - u_\delta \sigma^*_{M_3} \tag{14}$$

when $H_0(ID_A) = bP + c_\alpha P$, $H_1(ID_B) = k_\beta P$, and $H_2(W^*, M^*) = u_\delta P$.

Next, we analyze the success probability of $\mathscr{B}$; $\mathscr{B}$ will not abort if the following conditions hold:

(i) $ID_A = ID_\alpha$.

(ii) $W^* = w_\beta$.

(iii) $M^* = m_\delta$.

Therefore, if $\mathscr{A}_{II}$ has a nonnegligible probability ϵ in breaking the proposed ID-based proxy signature scheme, then the success probability of $\mathscr{B}$ in solving CDH problem is

$$Succ^{CDH}_{\mathscr{B},\mathbb{G}_1} \geq \frac{\epsilon}{\left(q_{H_0} + q_k + q_{os's} + 2q_{ps}\right)\left(q_{H_1} + q_{os's} + q_{ps}\right)\left(q_{H_2} + q_{ps}\right)}. \tag{15}$$

which is nonnegligible. Thus, we reach a contradiction. □

Theorem 7. *The revised ID-based proxy signature scheme is secure against the $\mathscr{A}_{III}$ chosen message and identity attack if the CDH assumption holds.*

Proof. The security is similar to that in Theorem 6. Thus, we just describe it briefly.

(1) **Setup**, **Hash queries,** and **Key extract** queries are the same as those in the security proof against a malicious proxy signer.

(2) **Proxy signer's standard signing queries** and **Proxy signing queries** are similar to the **Original signer's stand signing queries** and **Proxy signing queries** in the security for Theorem 6.

Through simulation, it can be reduced that if there exists a malicious original signer that can break the proposed scheme with a nonnegligible probability ϵ, then we can build another probabilistic polynomial time algorithm $\mathscr{B}$ that can solve the CDH problem with a nonnegligible probability $Succ^{CDH}_{\mathscr{B},\mathbb{G}_1}$ such that

$$Succ^{CDH}_{\mathscr{B},\mathbb{G}_1} \geq \frac{\epsilon}{\left(q_{H_0} + q_k + q_{ps's} + 2q_{ps}\right)\left(q_{H_1} + q_{ps's} + q_{ps}\right)\left(q_{H_2} + q_{ps}\right)} \tag{16}$$

where $q_{ps's}$ refers to the number of proxy signer's standard signing queries. Thus, we reach a contradiction. □

Table 1: Comparison regarding the computational costs.

Schemes	ProSign	ProVer
Wu et al.'s scheme [13]	$2 \cdot A_{\mathbb{G}_1} + 2 \cdot M_{\mathbb{G}_1} + 1 \cdot T_H$	$5 \cdot P + 4 \cdot T_H$
Our scheme	$3 \cdot A_{\mathbb{G}_1} + 4 \cdot M_{\mathbb{G}_1} + 2 \cdot T_H$	$5 \cdot P + 4 \cdot T_H$

6.1. Efficiency Analysis. We analyze the efficiency of the revised proxy signature scheme and compare it with the original scheme. The detail computation costs are presented in Table 1. As have been noticed, some algorithms in the revised scheme remains unchanged; thus, we only concern those algorithms that are different in our and the original schemes. Let $M_{\mathbb{G}_1}$, $A_{\mathbb{G}_1}$ denote the multiplication add addition calculations in $\mathbb{G}_1$, T_H denote the calculation of hash function (either H_0, H_1, or H_2), and let P denote the calculation of paring. We can see that our revised proxy signature scheme involves only one addition, two multiplication, and one hash operation in the proxy signing algorithm. As for the expensive paring operations needed in the proxy verification parts, the numbers are exactly the same.

7. Conclusion

In this paper, we introduced a practical attack which has not been considered by some existing proxy signature schemes. In particular, we took an identity-based proxy signature scheme to describe how this attack works. We also presented an enhanced security model that can capture this attack. Our model has considered different types of potential adversaries against an identity-based proxy signature scheme and allowed the adversary to query the individual signatures of both the original signer and the proxy signer. The proposed new scheme inherits the good features of the original scheme and at the same time can effectively prevent the attack. The proposed method can also be applied in other proxy signature schemes [18–21] to ensure an improved security.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors gratefully thank Xinyi Huang and Yong Yu for discussions on this work.

References

[1] W. Ren, R. Liu, M. Lei, and K.-K. R. Choo, "SeGoAC: A tree-based model for self-defined, proxy-enabled and group-oriented access control in mobile cloud computing," *Computer Standards & Interfaces*, vol. 54, pp. 29–35, 2017.

[2] Y. Yang, H. Zhu, H. Lu, J. Weng, Y. Zhang, and K.-K. R. Choo, "Cloud based data sharing with fine-grained proxy re-encryption," *Pervasive and Mobile Computing*, vol. 28, pp. 122–134, 2016.

[3] C. Calvelli and V. Varadharajan, "An analysis of some delegation protocols for distributed systems," in *Proceedings of the [1992] The Computer Security Foundations Workshop V*, pp. 92–110, Franconia, NH, USA.

[4] B. Neuman, "Proxy-based authorization and accounting for distributed systems," in *Proceedings of the [1993]. The 13th International Conference on Distributed Computing Systems*, pp. 283–291, Pittsburgh, PA, USA.

[5] B. Lee, H. Kim, and K. Kim, "Strong proxy signature and its applications," in *Proceedings of the SCIS*, vol. 1, pp. 603–608.

[6] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "Security architecture for computational grids," in *Proceedings of the 1998 5th ACM Conference on Computer and Communications Security, CCS-5*, pp. 83–92, November 1998.

[7] X. Jia, D. He, Q. Liu, and K. R. Choo, "An efficient provably-secure certificateless signature scheme for Internet-of-Things deployment," *Ad Hoc Networks*, vol. 71, pp. 78–87, 2018.

[8] A. Castiglione, K. Raymond Choo, M. Nappi, and S. Ricciardi, "Context Aware Ubiquitous Biometrics in Edge of Military Things," *IEEE Cloud Computing*, vol. 4, no. 6, pp. 16–20, 2017.

[9] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pp. 48–56, ACM Press, March 1996.

[10] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," in *Information and Communications Security*, vol. 1334 of *Lecture Notes in Computer Science*, pp. 223–232, Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[11] Z. Fangguo, R. Safavi-Naini, and L. Chih-Yin, "New proxy signature, proxy blind signature and proxy ring signature schemes from bilinear pairing," *CiteSeer*, 2003.

[12] G. Fuchsbauer and D. Pointcheval, "Anonymous Proxy Signatures," in *Security and Cryptography for Networks*, vol. 5229 of *Lecture Notes in Computer Science*, pp. 201–217, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[13] B. Xiao, L. T. Yang, J. Ma, C. Muller-Schloer, and Y. Hua, "Identity-based proxy signature from pairings," in *Proceedings of the Autonomic and Trusted Computing, 4th International Conference, ATC 2007*, vol. 4610, Springer Berlin Heidelberg, Hong Kong, China, July 2007.

[14] K. Zhang, "Threshold proxy signature schemes," in *Proceedings of the Information Security, First International Workshop, ISW '97*, pp. 282–290, Tatsunokuchi, Japan, September 1997.

[15] B. Lee, H. Kim, and K. Kim, "Secure mobile agent using strong non-designated proxy signature," in *Information Security and Privacy: Proceedings of the 6th Australasian Conference (ACISP '01), Sydney, Australia, July 11–13, 2001*, vol. 2119 of *Lecture Notes in Computer Science*, pp. 474–486, Springer, Berlin, Germany, 2001.

[16] G. Wang, "Designated-Verifier Proxy Signature Schemes," in *Security and Privacy in the Age of Ubiquitous Computing*, vol. 181 of *IFIP Advances in Information and Communication Technology*, pp. 409–423, Springer US, Boston, MA, 2005.

[17] W. Liu, G. Yang, Y. Mu, and J. Wei, "k-time proxy signature: formal definition and efficient construction," in *Provable security*, vol. 8209 of *Lecture Notes in Computer Science*, pp. 154–164, Springer, Heidelberg, 2013.

[18] X. Huang, W. Susilo, Y. Mu, and W. Wu, "Proxy signature without random oracles," in *Mobile Ad-Hoc and Sensor Networks*, vol. 4325, pp. 473–484, Springer, Berlin, Germany, 2006.

[19] L. Jin, K. Kwangjo, Z. Fangguo, and C. Xiaofeng, "Aggregate proxy signature and verifiably encrypted proxy signature," in *Proceedings of the Provable Security, First International Conference, ProvSec 2007*, pp. 208–217, Wollongong, Australia, 2007.

[20] Y. Sun, C. X. Xu, Y. Yu, and Y. Mu, "Strongly unforgeable proxy signature scheme secure in the standard model," *The Journal of Systems and Software*, vol. 84, no. 9, pp. 1471–1479, 2011.

[21] W. Liu, Y. Mu, and G. Yang, "Attribute-Based Signing Right Delegation," in *Network and System Security*, vol. 8792 of *Lecture Notes in Computer Science*, pp. 323–334, Springer International Publishing, Cham, 2014.

[22] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature schemes for delegation of signing rights," *Journal of Cryptology*, vol. 25, no. 1, pp. 57–115, 2012.

[23] T. Malkin, S. Obana, and M. Yung, "The hierarchy of key evolving signatures and a characterization of proxy signatures," in *Advances in Cryptology—EUROCRYPT 2004*, vol. 3027 of *Lecture Notes in Computer Science*, pp. 306–322, Springer, Berlin, Germany, 2004.

[24] J. C. N. Schuldt, K. Matsuura, and K. G. Paterson, "Proxy signature secure against key exposure," in *Public Key Cryptography—PKC 2008: 11th International Workshop on Practice and Theory in Public-Key Cryptography, Barcelona, Spain, March 9-12, 2008. Proceedings*, vol. 4939 of *Lecture Notes in Computer Science*, pp. 141–161, Springer, Berlin, Germany, 2008.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 2769417, 19 pages
https://doi.org/10.1155/2018/2769417

Research Article

Hydra-Bite: Static Taint Immunity, Split, and Complot Based Information Capture Method for Android Device

Ziru Peng,[1,2] **Xiangyang Luo**,[1,2] **Fan Zhao**,[1,2] **Qingfeng Cheng**,[1,2] **and Fenlin Liu**[1,2]

[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China*
[2]*Zhengzhou Science and Technology Institute, Zhengzhou 450001, China*

Correspondence should be addressed to Xiangyang Luo; luoxy_ieu@sina.com

Received 8 March 2018; Revised 17 April 2018; Accepted 23 May 2018; Published 17 July 2018

Academic Editor: Kim-Kwang Raymond Choo

In order to attract attention to the malicious use of large-scale operation of applications, Hydra-Bite, an Android device privacy leak path implemented by splitting traditional malicious application and restructuring to a collaborative application group, is proposed in this paper. For Hydra-Bite, firstly, traditional privacy stealing Trojan is analyzed to obtain the permission set. And the permission set redundancy elimination splitting algorithm is subsequently adopted to extract the simplest key permission set and split the set by functions so as to form the collaborative application group. Then, a covert channel is adopted for the intergroup Apps to remove the information's taint tagged by security methods. Meanwhile, a communication medium selection algorithm and an information normalization coding method are proposed to improve the efficiency and the concealing property for taints removal. Finally, collaborative external transmission of information is realized on the basis of intragroup Apps' communication. The experimental results show that Hydra-Bite could resist the detecting and killing of about 60 security engines such as Kaspersky, McAfee, and Qihoo-360 in VirusTotal platform and capture the privacy information of the devices of different versions from Android 4.0 to Android 7.0. Hydra-Bite can resist the killing of the following two methods, the typical detection tool Androguard based on "permission-API" and the typical static taint tracking tool FlowDroid. Compared with traditional privacy stealing Trojan, Hydra-Bite has higher information capture rate and stronger antikilling performance.

1. Introduction

Android operating system is widely applied in ILDs (Intelligent Devices), covering home furnishing, communication, business and vehicle-mounted terminals, etc. As reported, Android operating system has a global occupancy of 86.2% in the ILD market. ILD can store massive key information of users, e.g., location, communication records, accounts, and movement tracks. Along with the large-scale operation of application programs, that is, the same operational entity operates multiple Apps, such operation mode may be utilized by information selling organization and the key information of the users may be stolen by interapplication collaboration. We explore and report such stealing mode. The first purpose is, at the research level, attracting the attention of relevant security researchers. The second purpose is, at the application level, promoting the research on the App security audit mechanism in the platform. Based on the above purposes, this paper wants to prevent potential large-scale user information collection behavior in ILDs.

Traditional information stealing Trojan is mainly implemented by a single App. Those Trojans can be divided into two categories. The first category is Root permission applying (Root for short) and the second category is non-Root permission applying (non-Root for short). Specifically, typical Root approaches include Rootcager [1], Hellfire, Jmedia, and Bgserv [2]. Once "Hellfire" succeeds in promoting to Root permission through in-packet nesting, cloud matching, etc., it has extremely strong antikilling ability. In this condition, common antivirus software cannot completely remove it. However, the Root category is fragile. For example, before installation, Root method greatly depends on user state and system environment; after installation, it may trigger antivirus software's alarm immediately before promoting to the permission, so Root category has poor operability. Typical non-Root category includes methods which are Zsone [3],

GPSSPY and Nickyspy [4], SMS Tracker, Spitmo, and Zitmo, and the information is stolen usually through the application for excessive permission. Typical non-Root Trojan is the Soundcomber [5] which steals the information through call recording, speech recognition, and other technologies. Compared with the method of capturing key information by applying Root permission, other methods that do not apply for Root permissions have low recognition degree but the methods that do not apply for Root permission are easily blocked or detected by the users through searching and uploading information behaviors, especially after dynamic permission mechanism is introduced into Android. These Trojans can steal a lot of information, thus causing harm to equipment privacy. For example, leakage of Wi-Fi information and social information may cause devices' location information to be tracked [6–8].

For the above information stealing Trojan, researchers have proposed multiple information protection strategies. These strategies can be divided into "permission-API" detection and "taints tracking." The "permission-API" detection strategy includes the typical selectable authorization tool Kirin [10] and its improved version Apex [11], excessive permission detection tool Stowaway [12] and PScout [13], etc. They judge whether App is malicious through calling sensitive API and detecting risk permission combination, but such mode has high false alarm rate and it is difficult to cope with the privacy stealing method based on "collaboration." Therefore, the "static taint tracking" of key information has been researched more and more since 2013. The representative tools are as follows: static analysis tool ScanDroid [14], DroidChecker [15], Chex [16], COVERT [17], etc. These tools carry out static analysis of App through detection of permission, sensitive data leak path and data-flow analysis, etc. Additionally, App data-flow analysis tool FlowDroid [9] establishes the propagation path of the key information from "sensitive source" (e.g., IMEI number, longitude, and latitude) to "receiving node, sink" (e.g., sending short message, uploading in Internet) through static taint tracking, thus tracking the taint information flow. At present, the method has been taken by several security engines as the analysis kernel of Android application, but it is difficult to implement "static taint tracking" on the system bottom frame. The literatures [18–20] are other covert communication detection methods.

In order to attract attention to the large-scale operation of Apps used maliciously, this paper researches the Android system's application layer and proposes Hydra-Bite. Specifically, firstly, Hydra-Bite uses the permission split and reconstruction module to split traditional privacy stealing Trojan, and collaborative App group is constructed. In the first step, the problem is the coarse permission particle size of the collaborative application group. To solve the above problem, the permission set redundancy elimination splitting algorithm is proposed to extract the key permission set and split it by functions. Secondly, the taint cleaning module and taint tagged by the static taint tracking method on the key information are cleaned through Android covert channel. To solve the problem of wide varieties of communication medium and bandwidth [23], information normalization coding method is proposed. In the second step, to solve the problem that communication medium is easily occupied by irregular user operations, the communication medium selection algorithm is proposed. The experimental results show that, compared with traditional privacy stealing Trojan, Hydra-Bite has higher information capture rate and stronger antisearching and antikilling rate.

This paper's major contributions are as follows.

(1) Static taint immunity, split, and complotted based information capture method: Hydra-Bite is an information capture method for Android devices. This method can get information and avoid killing by collaborative Apps. At the same time, Hydra-Bite cleans the mark tagged by security methods through covert channel. This paper sends a security alert about Hydra-Bite.

(2) Proper split method for permission sets: Hydra-Bite proposes permission set split after removal of redundancy algorithm. This algorithm can split the permission set of traditional malicious applications and reconstruct the permission sets after the split to be a cooperative application group.

(3) Information and communication media adaptation: Hydra-Bite proposes a normalized coding method. This method solves the problem of the difficulty of covert communication media and information adaptation before information enters the covert channel.

(4) Dynamic selection of communication media: Hydra-Bite proposes a dynamic selection algorithm for communication media. The algorithm solves the problem of how to dynamically select the largest communication bandwidth medium when some media are occupied.

(5) Taint cleaning: Hydra-Bite proves the shortage of existing static taint tracking methods. Hydra-Bite can bring coded information with taint through covert channel, so that the taint cannot be tracked.

The remaining content of this paper is arranged as follows: in Section 2, "Hellfire," "Soundcomber," and "FlowDroid" are taken as examples to introduce relevant work from the two aspects of "Trojan" and "Protection"; in Section 3, method's principles and steps are explained in detail; in Section 4, the proposed method is evaluated from the two aspects of "information capture" and "antikilling performance"; Section 5 is the conclusion.

2. Related Work

This section firstly illustrates the function basis of Hydra-Bite by introducing traditional data capture Trojans "Hundreds," "Gypsymoth," "Hellfire," and "Soundcomber"; then, through the introduction of the peculiarity of "Permission-API" and "static taint tracking," it explains the weakness of traditional privacy-preserving methods when facing the Hydra-Bite privacy leak path.

2.1. Traditional Key Information Capture Method. Traditional key information capture methods are based on obtaining Root permission and applying for excessive permissions. The following part will explain both in detail.

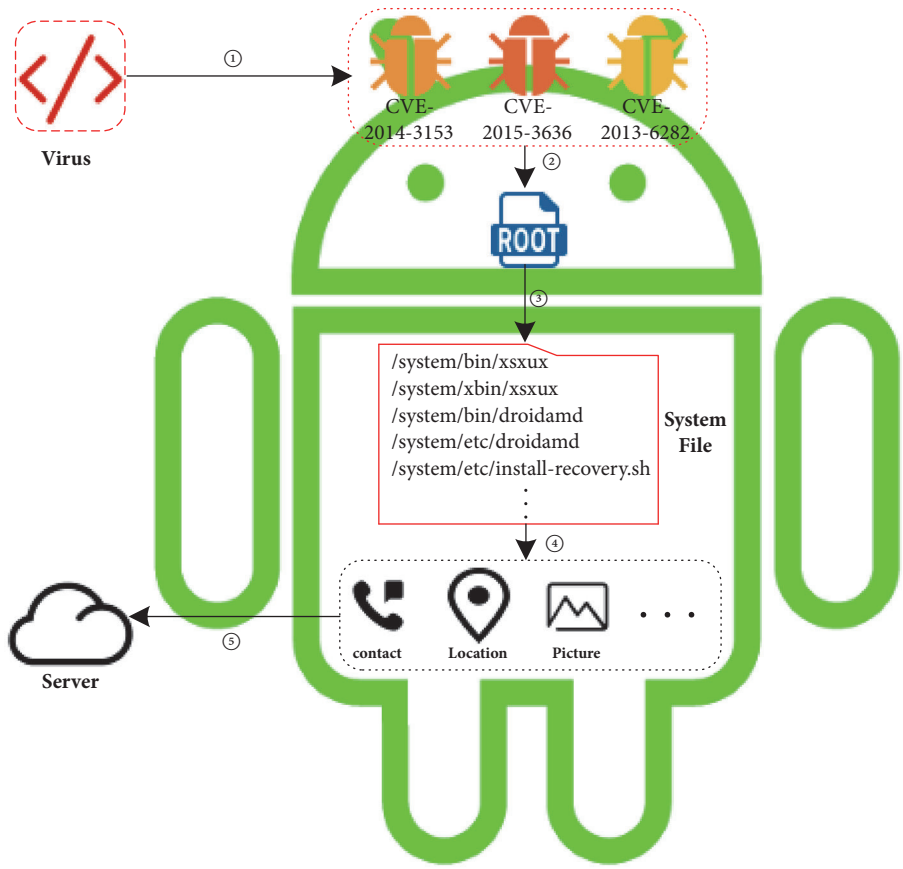


Figure 1: The process of "Hundreds" infects equipment.

(1) The Capturing Methods Based on Root Permission Acquisition. In June 2015, the virus named "Hundreds" is active. The principle of Hundreds to capture information is shown in Figure 1. Steps ①, ②, and ③: The App carrying virus enters the device and then releases the privilege promotion code and core module. Steps ④, ⑤, and ⑥: After the permission promotion code begins to work, the information of system version is uploaded to the server. Step ⑦: The server identifies the system and returns the Root scheme that is compatible with the current system version. Steps ⑧ and ⑨: The module uses the above Root scheme to invade the system folder for the convenience of fake itself as a system App, so that Hundreds can self-start and cannot be deleted. Steps ⑩ and ⑪: The Hundreds collects device information and uploads it to the server.

In December 2015, the virus named Gypsymoth is active. The principle of Gypsymoth to capture information is shown in Figure 2. Steps ① and ②: The virus has been promoted to Root permissions, through system's vulnerabilities. Step ③: Some system files are replaced by Gypsymoth after Root permission was promoted so that Gypsymoth can reside in the system by monitoring system running environment and sniffing file change. Steps ④ and ⑤: When the basic survival requirements are satisfied, Gypsymoth starts capturing information and uploads it to the server.

In June 2016, the virus named Hellfire is active. The principle of Hellfire to capture information is shown in Figure 3. Step ①: The virus carrying App enters the device. Step ②: The virus carrying App releases its subpackage which is used to obtain Root permission. Steps ③ and ④: The subpackage gets the SDK version of current system and sends it to server. Steps ⑤ and ⑥: Hellfire receives and executes Root scheme returned from server. After Root permission promotion, Hellfire can parasite to the underlying module of the system and reside in equipment to collect device information continuously.

Methods to capture key information have a significant effect when Root permission is promoted. However, these kinds of method have strong user and system environment dependence. These kinds of method do not work for systems that disable Root permission or users who pay attention to App's permissions.

(2) The Capturing Methods by Obtaining Excessive Permissions. Excessive permissions are the permissions beyond those which are necessary to meet the App's function. Method is based on excess permissions, usually applied for a large number of permissions and disguised as a normal application, by application repackaging.

In 2011, the Soundcomber [5] method was proposed by Schlegel et al. applied for excessive permission, such as sound

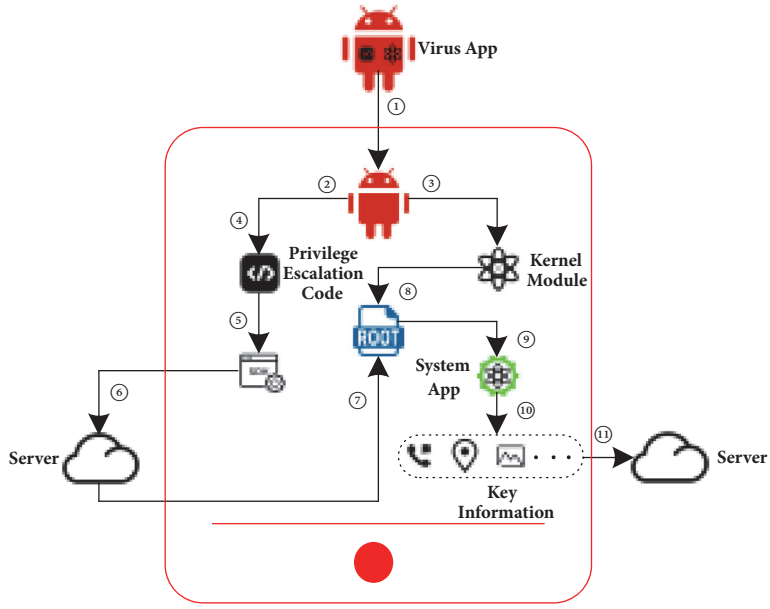


Figure 2: The process of "Gypsymoth" infects equipment.

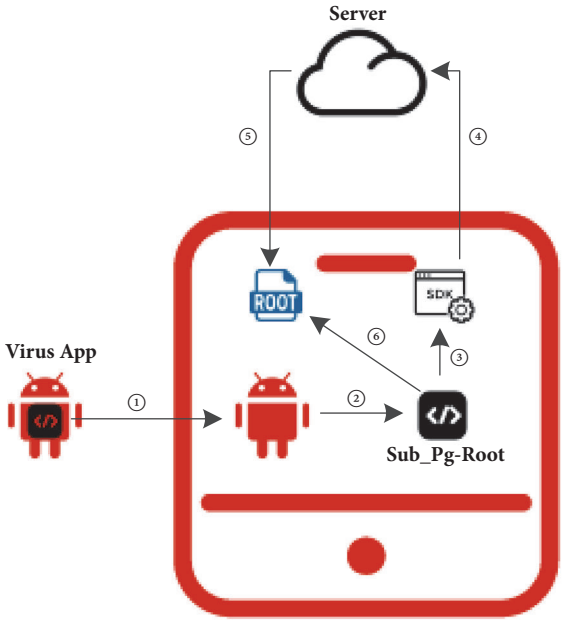


Figure 3: The process of "Hellfire" infects equipment.

recording, disk access, and Internet access. When monitoring the device calls, these permissions of Soundcomber are used to turn on the recording function. Then Soundcomber will store the audio files recorded before. Finally, when the condition is suitable, Soundcomber recognizes key words from audio files and uploads them to the network.

In 2012, the Tapprints [24] method proposed by Emiliano et al. applied permissions for accelerometer, gyroscope, and IMEI number. Tapprints identifies the device model with the IMEI number and then queries screen size by the device model. After that, Tapprints silently listens to users' clicking coordinates and clicking objects on screen. Tapprints combines machine learning method to infer users' input in devices, with the previous coordinates and objects.

In 2017, with the popularity of smart wearable equipment, Maiti et al. proposed a method of obtaining information user inputted through the smart wearers' sensors [25]. Their method applies for permissions to access sensors that belong to mobile communication equipment and wearable equipment. Finally, there is observation of hand movements by permissions applied before to improving the accuracy of information capturing.

The above device information stealing method based on excessive permission avoids the system environment dependence. However, this method still has strong user dependence. Meanwhile this kind of method can also be blocked by the existing security means [26, 27].

2.2. Methods for Preventing Key Information Capturing. To detect and block App's key information obtaining, there are mainly two methods: one based on detecting the mapping relation of "permission-API" and the other conducting static taint analysis on the App. The following part will introduce both in detail.

(1) The Protecting Methods Based on Detecting "Permission-API" Mapping. This detecting method is based on "permission-API" mapping which can estimate whether the Apps conduct key information capturing by analyzing if the App applies for high-risk permissions such as Root, or whether Apps call high-risk API combination such as recording and uploading. The representative within this type of protecting methods is PScout [13] which analyzes the Android system Source code and obtains the mapping relationship between API set and permission set, and it

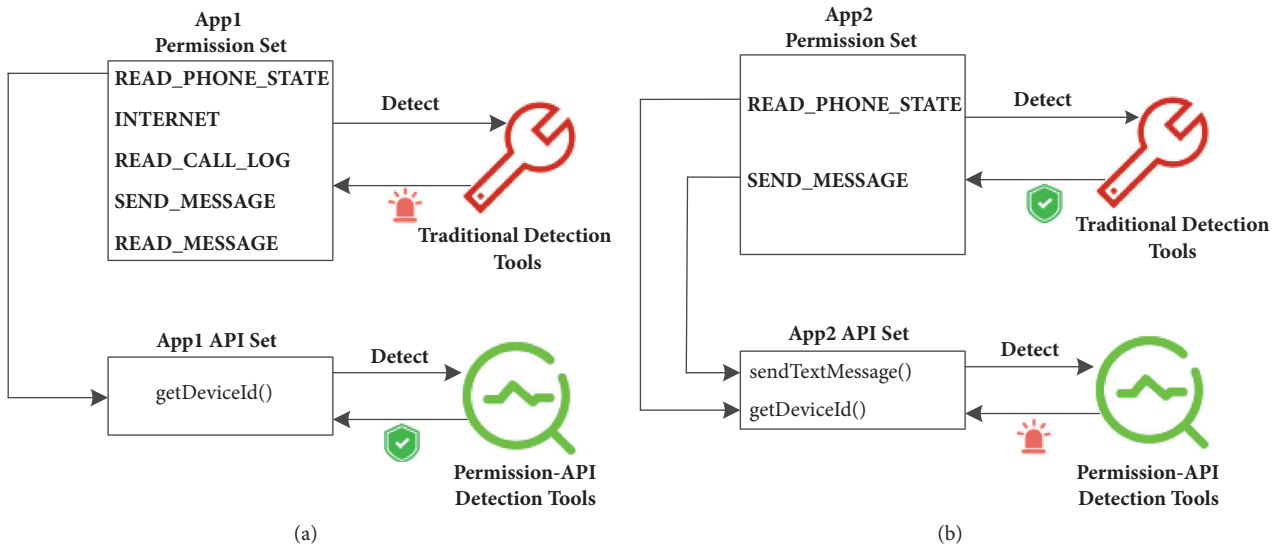


Figure 4: The different detection results between App1 and App2. (a) The detection results of App1. (b) The detection results of App2.

can conduct the API-Level analysis of the App through the particular mapping relationship.

As illustrated in Figure 4, App1 in Figure 4(a) is a benign application as it applies for unused sensitive permissions but it does not call any API to leak key information; App2 on Figure 4(b) only applies for "reading IMEI number" and "sending text messages" two permissions, which contains key information capturing potential. If we only analyze App1 and App2 from the permission aspect, App1 would trigger the alert and be wrongly diagnosed as it applies for lots of key information reading and uploading permissions and generates high-risk combination. However, if we analyze App1 and App2 from the more detailed API calling aspect through "permission-API" detecting method, App2 would trigger the alert as it calls for more dangerous API combination. Therefore, it improves the detecting accuracy through the API-Level scanning of the App.

(2) The Protecting Methods Based on Static Taint Tracking of the App. Static taint tracking method detects the complete path from where the key information firstly gets captured at the Source, to the sink point where it has leaked out of the App by analyzing the App on the Source level. This analysis method can monitor the data-flow path of Apps' key information reading and block the capture method through Apps applying for excess permissions. The representative work is FlowDroid [9], an Android App's high-accuracy static taint tracking method invented by Arzt et al. By imitating the complete Android life cycle, FlowDroid uses analysis method according to different demands, which is highly accurate and highly efficient comparing to other methods of this type.

Figure 5 is an illustration of taint analysis in real circumstances. This illustration contains two parts; the first one is forward taint analysis (①, ②, and ⑦), which tracks the flow of taint variables. The second one is backwards on-demand analysis, which detects the aliases before they get tagged by the taints. FlowDroid generates complete taints

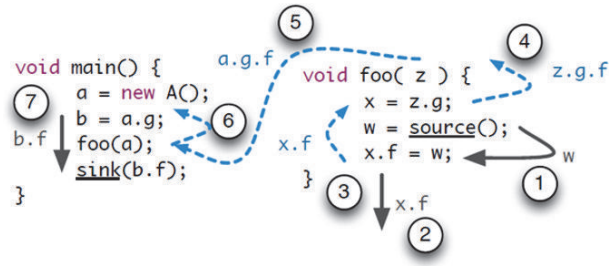


Figure 5: FlowDroid taint analysis [9].

path by using the above-mentioned forward taint analysis and backwards on-demand analysis. However, it faces APK files, and FlowDroid method can hardly track the taints hidden deep down in the bottom of the system.

3. Proposed Method

This section will outline the architecture of Hydra-Bite, according to Figure 6. Firstly, some terms and concepts which will appear in the later description are defined. Then, according to Figure 6, the process of Hydra-Bite will be outlined.

3.1. Terms and Definitions. Hydra-Bite will be introduced systematically in Section 3.1. The principle of Hydra-Bite is shown in Figure 4. And, for the convenience of the following description, definitions of terminology, abbreviated, are shown as follows:

(1) Traditional malicious applications: App_TM, normal applications: App_Norm.

(2) Key information (K_Info): The information can be read only when privacy-related permissions applied.

(3) Permission set (*PSet*): All permissions applied by the application form a set, marked as *PSet*, *PSet* can be divided

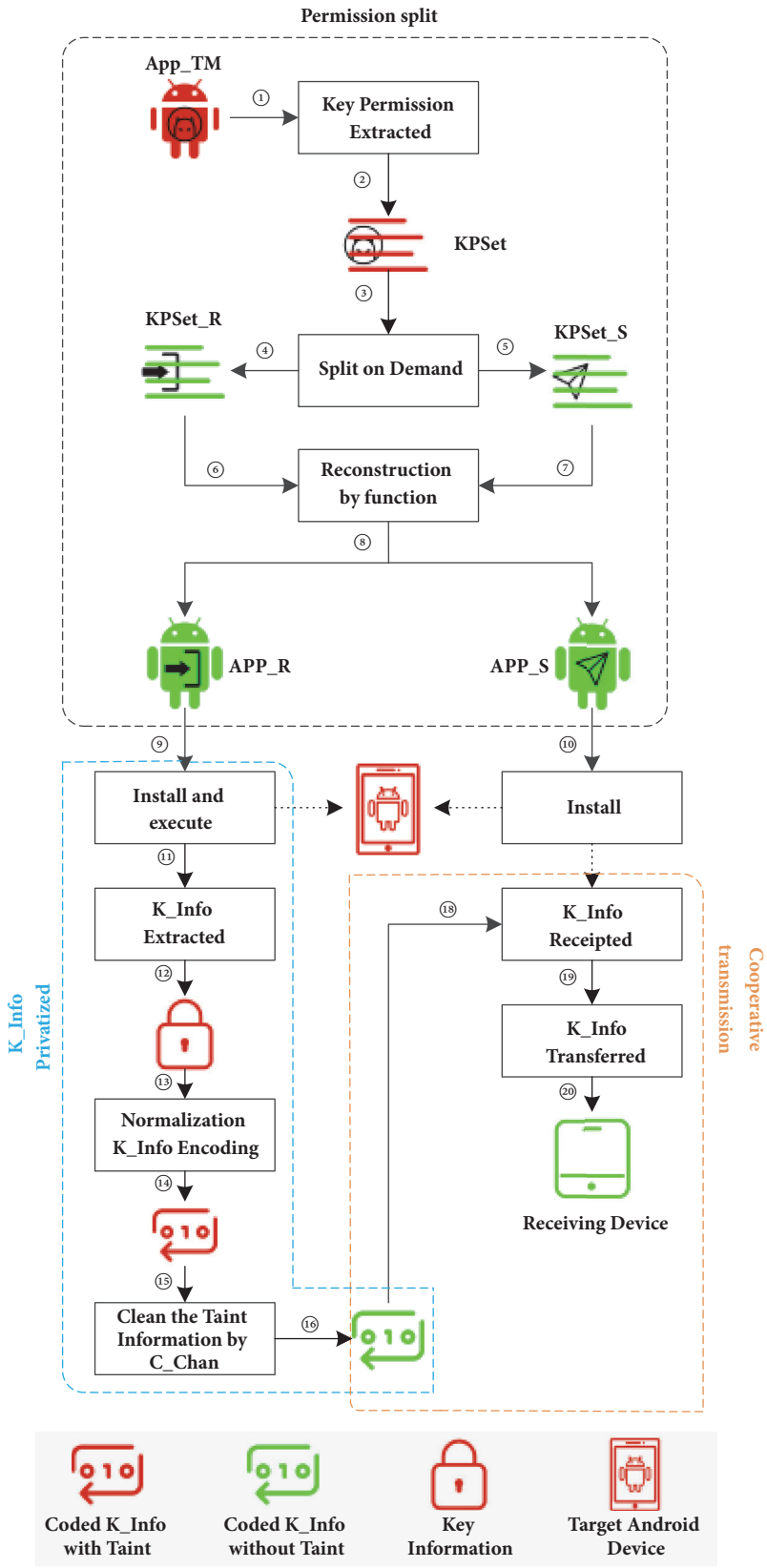


Figure 6: The general framework of Hydra-Bite.

into subsets *PSet_R* and *PSet_S* according to the process of information reading and sending. The relations between the above two subsets are as follows:

$$PSet\_R \cap PSet\_S = \emptyset.$$

$$PSet\_R \cup PSet\_S \subseteq PSet.$$

(4) Key permission set (*KPSet*): Permissions in *PSet* only related to the whole process of obtaining K_Info are selected to constitute a set, recorded as *KPSet*. The *KPSet* can be further divided into read permissions' set *KPSet_R* and send permissions' set *KPSet_S* according to API calls in process of obtaining K_Info. The relations of the above sets are as follows:

$$KPSet\_R \subseteq PSet\_R.$$

$$KPSet\_S \subseteq PSet\_S.$$

$$KPSet\_R \cup KPSet\_S \subseteq KPSet.$$

$$KPSet\_R \cap KPSet\_S = \emptyset.$$

$$KPSet \subseteq PSet.$$

(5) Key information with taint (K_Info_T): After K_Info is read, static taint tracing method will tag it, which is denoted as K_Info_T.

(6) Normalized coded key information with taint (K_Info_N_T): Normalized coded K_Info_T is written as K_Info_N_T.

(7) Normalized coded key information without taint (K_Info_N): After the taint cleaning process, K_Info_N_T's taint tag is washed off; the result is recorded as K_Info_N.

(8) Covert channel: It can be expressed as a triplet $\langle SRes, EM_h, EV_l \rangle$ [26] where *SRes* are shared resources in Android system; EM_h is a communication entity with higher security level which can modify *SRes*; EV_l is a communication entity with lower security level which can observe or perceive *SRes*. Communication from EM_h to EV_l is not allowed. Then the channel that completes the two-entity communication is called a covert channel.

(9) Cooperative applications group (Co_Apps): Co_Apps is created by our Hydra-Bite method, which can communicate with other applications which are in the group.

3.2. Process of Method. The method is composed of 3 parts: Co_Apps reformed before *KPSet* split modular, K_Info privatized modular, and cooperate transmission modular. The following steps will be described specifically based on Figure 6.

(1) Co_Apps Reformed before KPSet Split

(1) *KPSet* Extracted (① and ②). Our method parses App_TM's installation package to get the AndroidManifest file, from which *KPSet* is extracted according to API calls in process of obtaining K_Info.

(2) KPSet split (③, ④, and ⑤). Hydra-Bite splits *KPSet* into *KPSet_R* and *KPSet_S* according to the phases which are read and send in the process of App_TM obtaining K_Info;

(3) Co_Apps reformed (⑥⑦, and ⑧). The App_R and App_S are structured as Co_Apps, according to the *KPSet_R* and *KPSet_S* they only owned.

(2) K_Info Privatized

(1) K_Info read (⑨, ⑩, ⑪, and ⑫). After installing and executing on the target Android device, App_R will read the K_Info which is protected by the system security mechanism to the application layer. At this time, K_Info will be tagged by static taint method to become K_Info_T.

(2) K_Info_T normalized (⑬ and ⑭). After read by App_R, K_Info needs to be translated into a uniform format that supports covert channel, because of various forms. Therefore, Hydra-Bite normalizes K_Info_T to K_Info_N_T through coding it.

(3) K_Info_N_T cleaned (⑮ and ⑯). If K_Info_N_T is transferred at this time, it will be detected because of the tagged taint. To clean the taint, C_Chan is designed in our method, so that the Hydra-Bite can get the K_Info_N.

(3) Cooperative Transfer

(1) Message receive (⑱). App_R opens the information receiving component of App_S to transfer K_Info_N, after K_Info privatized operation is done. Now the K_Info_N is private information of App_R.

(2) Information transmission (⑲ and ⑳). After receiving the K_Info_N, App_R opens the information transmission component and sends the K_Info_N to the receiving equipment through the *KPSet_S* which it owns.

In the above processes, the key steps which will be explained separately are as follows: *KPSet* Split on demand(①–⑤) and the taint cleaning by covert channel(⑬–⑯).

4. Key Issues Description

This section will give a detailed description of the key issues raised in permission split. They are key permission set split after redundancy removal, key information normalized, communication media selection, and taint cleaning through covert channel.

4.1. Key Permission Set Split after Removal of Redundancy. Permission set redundancy means that, in the process of information capture, the required number of permissions for App calls sensitive APIs is less than permission items in AndroidManifest file. To solve the problem that the granularity of Co_App's permissions is coarse caused by permission redundancy, this section proposes a deduplication algorithm for permission set before splitting it. The algorithm uses double layer's key-value mapping to extract key permission set. Then algorithm sets different labels for the items in key permission set. Algorithm 1 is the pseudo code of the algorithm's main loop.

```
(1)  define sApiList <apiMap<api_Name, perm_Name >>:
       List<Map<String, String>>
(2)  define kPermList <permMap<perm_Name, flag>>:
       List<Map<String, String>>
(3)  while sApiList ≠ ∅ do
(4)    for i = 0 to sApiList.size()
       //Traversing list of key-value mappings for sensitive APIs
(5)      map sApiList.get(i).getKey() to perm via the Mapping file
         provided by Pscout
       //Mapping sensitive APIs to permissions corresponded
(6)      sApiList.get(i).getValue() ⟵ perm
(7)      if perm is the first time show then
       //Remove the duplicate permissions after the mapping
(8)        swich(perm)
       //Classify permissions before add them into the second key-value mapping
(9)        case perm is transfer class permission:
(10)         permMap.getKey() ⟵ perm
(11)         permMap.getValue() ⟵ trans
(12)         add Map2 to kPList
(13)         clear Map2
(14)       case perm is information read class permission:
(15)         permMap.getKey() ⟵ perm
(16)         permMap.getValue() ⟵ read
(17)         add Map2 to kPList
(18)         clear Map2
(19)       case perm is other permission:
(20)         drop perm
(21)     else
(22)       drop perm
(23)     end if
(24)   end for
(25) end while
```

Algorithm 1: Main loop of key information split.

The purpose of the first layer's key-value mapping is to extract the permissions actually used by the App and then remove the repeated entries in it. This process is implemented as follows. Firstly, the algorithm sets the APIs that App actually invokes as the keys. Then the algorithm maps the APIs to the required permissions for invoking them, through the mapping file provided by PScout [13]. The reason for the implementation of the algorithm is that one permission can call multiple API in some cases. For example, the permission READ_CALL_LOG can invoke APIs which can read call category and call time. This layer's key-value mapping is shown in (5)-(6) lines of pseudo code, in Algorithm 1.

The purpose of the second layer's key-value mapping is to split the permissions actually used by the App, according to category. This process is implemented as follows. Firstly, the algorithm refines the real permission set by extracting permission items related to key information capture. The product of this process is key permission set. Then, the algorithm classifies and gives labels to the key permission set's items, according to reading or sending information. Finally, the algorithm sets items of key permission set as the keys and sets the labels for each item as values. Finally, Hydra-Bite can split the key permission set according to the labels. This layer's key-value mapping is shown in (7)–(23) lines of pseudo code, in Algorithm 1.

4.2. Key Information Normalized. There are two problems to be solved for the pretreatment of key information. One problem is for key information. Because of the diverse storage types and rich contents of this information in Android device, we counted the storage formats of some pieces of key information stored in the device and shown in Table 1. "▲" indicates that the key information exists in this format or content. It can be seen that some key information stored in Table 1 is different in type and content. The other problem is for *SRes*. Different shared resources have different forms of existence and communication bandwidth. Take the volume of the alarm clock and the brightness of the screen as an example; the thresholds of volume and brightness are 0–8 and 0–255 integer numbers, respectively. It can be seen that, due to the disunity of format and content, the adaptation between shared resources and different key information is difficult.

Shared resources are determined by the system, and it is difficult to modify. Therefore, in this section, aiming at the above problems, a method to normalize key information is proposed. The method is divided into format unified module

Table 1: Storage formats of some pieces of key information and their contents.

	IMEI Number	Fine Location	Contacts	Call Type	Call Date	SMS content
			Information Type			
String	▲		▲			▲
Double		▲				
Int				▲		
Long					▲	
			Information Content			
Number	▲	▲	▲	▲	▲	▲
Characters		▲	▲		▲	▲
Letter			▲			▲
Chinese			▲			▲
Other Languages			▲			▲

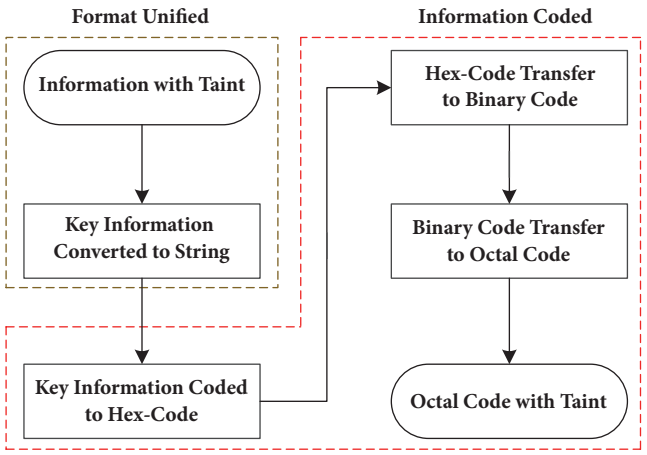


Figure 7: Key information normalization coding process.

and information coded module. And the flow chart is shown in Figure 7. The steps of the method are as follows.

Step 1. The key information to be processed is converted into string, which is convenient for subsequent coding.

Step 2. In order to unify the information format, all characters are converted into hex-code.

Step 3. Convert the hex-code into binary code and then turn binary code into octal code

For the convenience of shared resources' dynamic selection, each bit of coded information must be less than the bandwidth of the shared resource's threshold; besides the modification and observation of shared resources are a time-consuming process. Considering the time cost and bandwidth utilization rate, Hydra-Bite chose to turn the hex-code into octal code.

The number of bytes will increase after the information is encoded. This paper calculated the number of bytes after the information is encoded. When n is a number, the number of bytes before encoding and the number of bytes after encoding is (1). When n is a letter, the relationship is (2).

$$\text{Size}(n) = 3n - \left\lceil \frac{n}{3} \right\rceil, \quad n \text{ is a positive integer} \tag{1}$$

$$\text{Size}(n) = \begin{cases} 3n - \left\lfloor \frac{n}{3} \right\rfloor, & (n = 3i + 1,\ n, i \text{ are positive integers}) \\ 3n - \left\lceil \frac{n}{3} \right\rceil, & (n \neq 3i + 1,\ n, i \text{ are positive integers}). \end{cases} \tag{2}$$

4.3. Communication Media Selection. Another problem Hydra-Bite solved is the selection of shared resource. There are many kinds of shared resources, and the communication bandwidth between them is different. These shared resources are easily occupied by irregular user operations, which leads to low taint cleaning efficiency and the concealment of methods.

In this section, to solve the above problems, a communication media selection algorithm is proposed. By dynamically querying the occupancy status of shared resources, the algorithm iterates the largest *SRes* whose occupancy status is false. The result is the optimal shared resource.

The flow chart in Figure 8 and the pseudo code in Algorithm 2 describe the main loop of the algorithm. Firstly, a set of "medium-state" key-value mappings is used to identify the occupancy status of *SRes*. Hydra-Bite sets *SRes* as the "key" and sets the occupancy status as the "value." Algorithm queries *SRes* in real time and dynamically identifies its status, so as to realize the screening of available resources ((4)-(5) lines of pseudo code). Then Hydra-Bite queries the *SRes* in the "key-value" mapping table and recursively iterative "*fastKey*" with higher bandwidth to deploy *SRes* efficiently ((6)-(7) lines of pseudo code). The specific process of communication media dynamical selection is as follows.

Step 1. Define the state list *statList*<*SRes*> and its internal "key-value" map *SRes*< *SRes*, *OccStat*>, and set the names of shared resources as the "key" and occupancy status as the "value." Then judge whether the *statList* is empty, if *statList*

```
(1)  while statList ≠ ∅ do
(2)    declare fastKey : String
(3)    for i 0 to statList.size() by incr do //Traverse the list of key-value maps
(4)      while statList.get(i).getKey() is not occupied do
           //Determine whether the "key" in the current key-value map is occupied or not
(5)        Query bandwidth of statList.get(i).getKey()
             in speedMap<res, bandwidth>
           //Query unoccupied "key" bandwidth in the "SRes – Bandwidth" key table
(6)        if bandwidth > value.speedMap(fastKey)
             or value.speedMap(fastKey) = null then
(7)          fastKey ⟵ statList.get(i).getKey()
           //Iteration shared resources which has higher bandwidth
(8)        end if
(9)      end while
(10)   end for
(11)   return fastKey //Return the result of the iteration
(12) end while
```

Algorithm 2: Main loop of optimal shared resource selection.

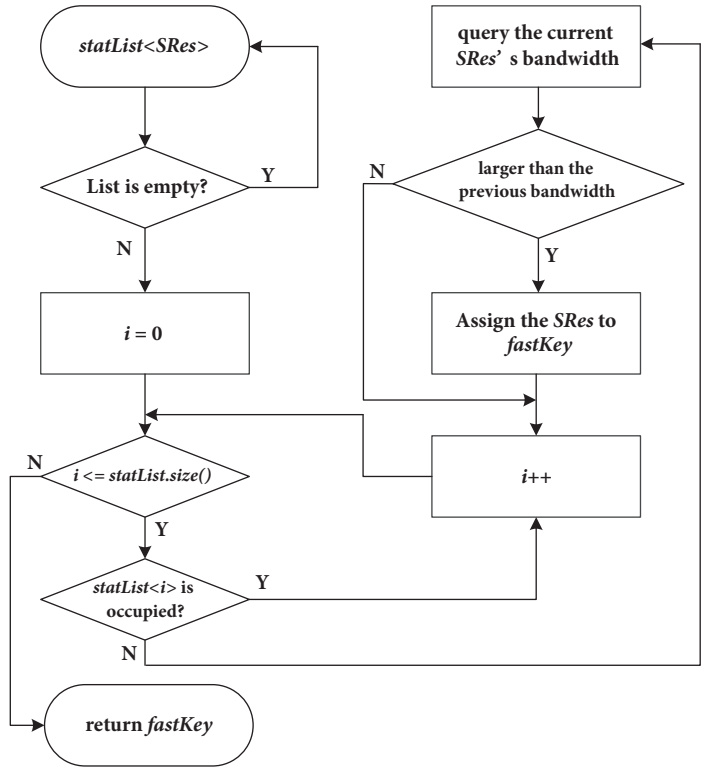


Figure 8: Main circulation flow chart of communication medium selection algorithm.

is empty, it indicates that the preset shared resources are all occupied by the user operation, and the current device state is not suitable for cleaning operation. If *statList* is not empty, then the state list is traversed.

Step 2. The traversal begins with the zero value in the *statList*. If the current state of *SRes* occupancy is true, the next value of *statList* will be judged. If the current state of *SRes* occupancy is false, then query the current *SRes* bandwidth.

Step 3. Compare the bandwidth of the current *SRes* with the previous bandwidth. If the current *SRes*'s bandwidth is large, assign the current bandwidth to the *fastKey*. If the current bandwidth value is small, then the first *SRes*'s bandwidth is still *fastKey*.

Step 4. No matter what *SRes* bandwidth value *fastKey* uses in the previous step, continue traversing *statList* until the loop end condition is satisfied.

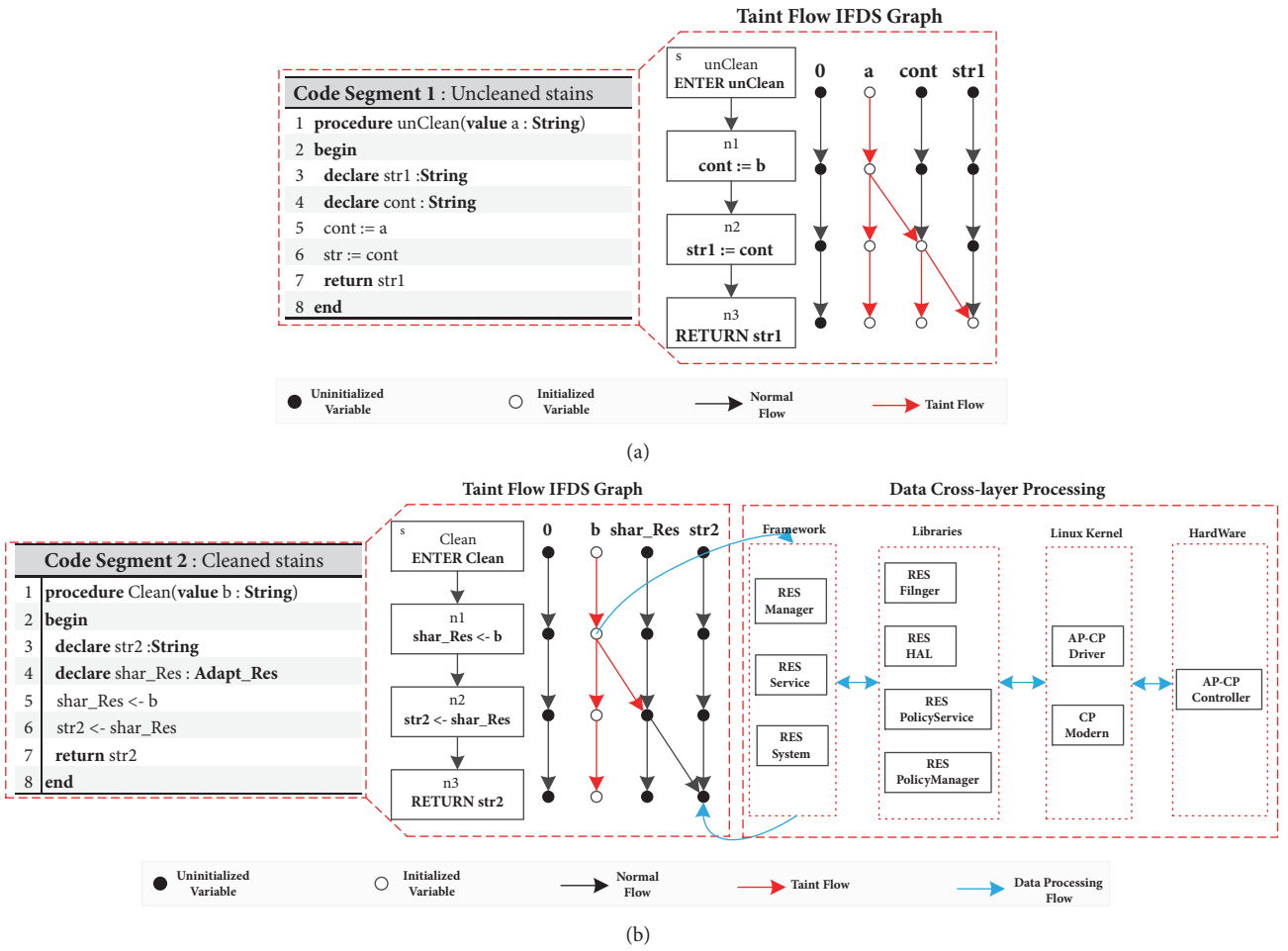


FIGURE 9: Taint analysis for different mode of transmission. (a) Transfer information directly via variables. (b) Transfer information via covert channel.

4.4. Taint Cleaning through Covert Channel. In the discussion of this section, readers need to know the definition of covert channel in Section 3.1 [26] and the IFDS algorithm [28].

Data flowing to the lower system layer are hard to be tracked by static taint tracking method (STT). Thus taint carried by EM_h cannot be tracked continuously by STT in the process of flowing to *SRes* at the bottom of operation system. The same reason can be obtained; Source of information read by EV_l from *SRes* cannot be tracked by STT. This communication is not allowed because the two entities do not communicate via security policy. At this point $\langle SRes, EM_h, EV_l \rangle$ has formed a covert channel communication triples.

Figure 5 shows the principle of EM_h, EV_l to get rid of the taint tracking by covert channel. The tainted data flow is shown in red directed line in this figure. On the left of Figure 9(a), the code segment 1 shows the process by which EM_h (**a**) propagates the information it carries through variable **cont** to EV_l (**str1**). On the right of Figure 9(a), the IFDS Graph is the results of data-flow analysis based on graph reachability of **cont** and **str1** using IFDS algorithm [28]. In Figure 9(b) the code segment 2 shows the communication entities **b** and **str2** realize the purpose of transmitting the key information by the covert channel through modification and reading *SRes* synchronous. Hydra-Bite needs to go through the framework layer, the library layer, and the kernel layer until the hardware layer to modify and read *SRes*. Take volume as an example; to change the volume, the volume manager(AudioManger) needs to call the system volume service(AudioService) to enter the volume system(AudioSystem), which can operate the AP-CP driver in the hardware abstraction layer (HAL). At this point STT is difficult to tag shar_Res(EM_h) and str2(EV_l).

5. Experimental Results

To verify the effectiveness of the Hydra-Bite method, this paper verifies the threat posed by the Hydra-Bite method to the privacy information itself through key information acquisition experiment and demonstrates the functional accessibility of Hydra-Bite. The experiment of run-time overhead is used to verify that the Hydra-Bite method is sufficient to transmit enough information in a limited time, which demonstrates the practicability of the method in time

Table 2: Test equipment and corresponding information.

Serial Number	Android Kernel Version	Android API	Market Share	Device Model
1	Android4.0	15	0.40%	Galaxy Note II
2	Android4.1	16	1.70%	MI 2
3	Android4.3	18	0.70%	MI 2S
4	Android4.4	19	12.00%	MI 3
5	Android5.0	21	5.40%	MI 4LTE-CU
6	Android5.1	22	19.20%	MI 4-LTE
7	Android6.0	23	28.10%	OPPO-A57
8	Android7.0	24	28.50%	MI 6
Total	—	—	96.00%	—

Table 3: The results of Co_Apps read-receive-send K_Info.

Permission	Android 4.0	Android 4.1	Android 4.3	Android 4.4	Android 5.0	Android 5.1	Android 6.0	Android 7.0
SendInfo Read the Key Information								
Location	▲	▲	▲	▲	▲	▲	★	★
Device State	▲	▲	▲	▲	▲	▲	▲	▲
Contact	▲	▲	▲	▲	▲	▲	★	★
SMS Message	▲	▲	▲	▲	▲	▲	★	★
WiFi State	▲	▲	▲	▲	▲	▲	▲	▲
Call Log	▲	▲	▲	▲	▲	▲	▲	▲
GetInfo Receive the Key Information								
Location	▲	▲	▲	▲	▲	▲	▲	▲
Device State	▲	▲	▲	▲	▲	▲	▲	▲
Contact	▲	▲	▲	▲	▲	▲	▲	▲
SMS Message	▲	▲	▲	▲	▲	▲	▲	▲
WiFi State	▲	▲	▲	▲	▲	▲	▲	▲
Call Log	▲	▲	▲	▲	▲	▲	▲	▲
GetInfo Send the Key Information								
Internet	▲	▲	▲	▲	▲	▲	▲	▲
SMS	▲	▲	▲	▲	▲	▲	★	★

overhead. The antikilling performance experiment is used to verify that Hydra-Bite method can void existing mainstream killing method, which demonstrates the tolerance of Hydra-Bite method in the face of the killing method.

5.1. Key Information Acquisition Experiment

(1) Key Information Acquisition Experiment's Set. In this section, to evaluate the performance of capturing key information, the Hydra-Bite method will be implemented in 8 different Android versions. Those versions are Android 4.0, Android 4.1... Android 7.0. Experimental device models and corresponding Android kernel version, Android API, and market share are shown in Table 2, where the market share uses the Google official website for week 1, 2018 statistics: developer.android.com.

(2) Key Information Acquisition Experiment's Results and Discussion. App_R and App_S are constructed to evaluate the Hydra-Bite's ability of capturing key information. App_R is constructed to read and clean the key information. App_S is constructed to receive the cleaned information from App_R and send to receive devices. In the 8 versions of Android listed in Table 2, this section tested the capturing ability of Co_Apps to six key information items listed in Table 3 which are geographic location, device status, equipment state, etc. Those key information items are marked by the Android system as requiring dangerous permission to access. The experimental results are shown in Table 3: "▲" and "★," respectively, represent that this operation is successful and this operation needs to be granted permission dynamically.

In Table 3, data, respectively, indicate the results of the operation: App_R to get key information, App_S to receive App_R's messages, and App_S to send key information outside. Before Android 6.0, Co_Apps can read key information directly and send out. After Android 6.0, there are inquiries when Co_Apps reads and sends. The reason for the above results is that the system after Android 6.0 adopts dynamic

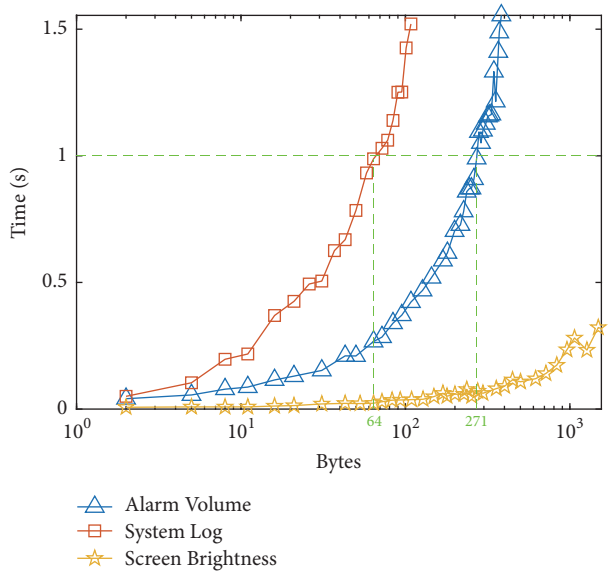


Figure 10: Overhead of taint cleaning.

permission granting mechanism and lets the user decide whether or not to grant app permission at run-time.

5.2. Running Time Overhead Performance Experiment

(1) Run-Time Overhead Performance Experiment's Settings. Right in the Hydra-Bite method EM_h the taint cleaning operation is carried out via covert channels, where access to the underlying resource is required twice per byte operation, which can be time-consuming. This paper illustrates the practicality of the Hydra-Bite approach in terms of time cost by examining the time overhead of the taint cleaning process.

In the experiment of running overhead performance, this section uses the device model MI_NOTE to test the time overhead of three covert channels which are "alarm volume," "screen brightness," and "system log." The device is equipped with Android 6.0 system and other hardware parameters that affect performance are listed below. The experimental device uses a 4-core CPU, whose basic frequency is 2.45GHz. The running memory of device is 3.00GB. And the kernel's version is 3.4.0-gf4b741d-00639-ge918701.

(2) Run-Time Overhead Performance Experiment's Results and Discussion. Run-time overhead performance experiment's results are shown in Figure 10. The results of each covert channel are marked with different symbols and colors. The green dotted line represents the amount of data that can be cleaned within a second.

It can be seen that when coded information has the number of bytes, system log covert channel's time overhead is the most expensive, because Android system processes generate a large number of logs. Filtering out encoded information from these logs is time-consuming. Screen brightness covert channel time overhead is minimum, because the value access and the brightness change are completed by different system modules, and this "separation" makes it easier to avoid waiting for hardware response. Cleaning rates of the above three covert channels are 78B/s, 280B/s, and 5.5KB/s, which are enough to transmit a certain amount of information.

Table 4: Part of antivirus engine in VirusTotal.

Serial Number	Antivirus Engine	Country
1	Alibaba	CHN
2	Antiy-AVL	CHN
3	Baidu	CHN
4	BitDefender	ROU
5	Arcabit	POL
6	CAT-QuickHeal	IND
7	Comodo	USA
8	Jiangmin	CHN
9	Kaspersky	USA
10	Kingsoft	CHN
11	McAfee	USA
12	Microsoft	USA
13	Qihoo-360	CHN

Table 5: Source of malicious sample.

Serial Number	Origin	Country	Quantity
1	VirousShare	USA	109
2	MalGenome	USA	73
3	GitHub	USA	26
4	zeltser.com	USA	19
5	bbs.pediy.com	CHN	17
6	bbs.kafan.cn	CHN	15
7	Google Group-	USA	14
9	bbs.duba.net	CHN	14
10	52pojie.cn	CHN	13
11	bbs.mumayi.net	CHN	12
12	Others	—	26
—	Total	—	338

5.3. Permission Set Split Effect Experiment

(1) Permission Set Split Effect Experiment's Setup. In this part, VirusTotal platform is used as an effective detection tool for collaborative application group. The platform is a multiengine file scanning tool created by Sistemas in 2004 [21]. And VirusTotal detects uploaded files through multiple security engines to determine whether there is malicious behavior. Some of the antivirus engines used by VirusTotal are listed in Table 4.

This paper chooses 10 Apps to be split from the malicious App set from Table 5 as samples. These samples all have redundancy of permission, and they apply multiple key information read permissions and sending class permissions. The sending class permissions, read class permissions, and VirusTotal detection rates of samples are listed in Tables 6, 7, and 8, respectively. "▲" indicates the samples apply for the

Table 6: Malicious sample's permission attributes for the transfer category.

Number of Sample	MD5 Value of Sample	Internet	SEND_SMS	CALL_PHONE
Mal_1	744c9f9ef5a3ad2559174523f1fd664d	▲	▲	▲
Mal_2	844bc220827f50539c67d09c3998a0da	▲	▲	■
Mal_3	899c92f0db1ec69e091795f4ddd251df	▲	▲	▲
Mal_4	4914c06560cdc3dfaca7c81eea9a33eb	▲	■	■
Mal_5	5192ad05597e7a148f642be43f6441f6	▲	■	▲
Mal_6	5895bcd066abf6100a37a25c0c1290a5	▲	▲	■
Mal_7	8947eae5c65df02d9c538b12ddaf636f	▲	■	▲
Mal_8	2908873c8ab99faa94ffe596499bd8f9	▲	■	▲
Mal_9	4884112ac7e599bd4dc20ccc91ce870c	▲	▲	■
Mal_10	375151412aff0b21d72207f08665d16d	▲	▲	▲

Table 7: Malicious sample's permission attributes for the read category.

Number of Sample	MD5 Value of Sample	Fine Location	Device State	Contacts	SMS content	WiFi State	Call Log
Mal_1	744c9f9ef5a3ad2559174523f1fd664d	▲	▲	■	▲	▲	▲
Mal_2	844bc220827f50539c67d09c3998a0da	▲	▲	▲	▲	■	▲
Mal_3	899c92f0db1ec69e091795f4ddd251df	▲	▲	■	▲	▲	▲
Mal_4	4914c06560cdc3dfaca7c81eea9a33eb	■	▲	▲	▲	▲	▲
Mal_5	5192ad05597e7a148f642be43f6441f6	▲	▲	▲	▲	▲	▲
Mal_6	5895bcd066abf6100a37a25c0c1290a5	▲	▲	▲	■	▲	▲
Mal_7	8947eae5c65df02d9c538b12ddaf636f	■	▲	▲	▲	■	■
Mal_8	2908873c8ab99faa94ffe596499bd8f9	▲	▲	▲	■	▲	■
Mal_9	4884112ac7e599bd4dc20ccc91ce870c	■	▲	▲	▲	▲	▲
Mal_10	375151412aff0b21d72207f08665d16d	▲	▲	▲	▲	■	▲

Table 8: Detection results of samples in VirusTotal platform.

Number of Sample	MD5 Value of Sample	VirusTotal Detection Result	VirusTotal Alarm Rate
Mal_1	744c9f9ef5a3ad2559174523f1fd664d	33/58	52.34%
Mal_2	844bc220827f50539c67d09c3998a0da	33/57	57.89%
Mal_3	899c92f0db1ec69e091795f4ddd251df	33/55	60.00%
Mal_4	4914c06560cdc3dfaca7c81eea9a33eb	37/57	64.91%
Mal_5	5192ad05597e7a148f642be43f6441f6	49/62	79.03%
Mal_6	5895bcd066abf6100a37a25c0c1290a5	49/62	79.03%
Mal_7	8947eae5c65df02d9c538b12ddaf636f	45/62	72.58%
Mal_8	2908873c8ab99faa94ffe596499bd8f9	32/54	59.26%
Mal_9	4884112ac7e599bd4dc20ccc91ce870c	32/57	56.14%
Mal_10	375151412aff0b21d72207f08665d16d	42/56	75.00%

permission and "■" indicates the samples do not apply for the permission. MD5 values for the corresponding samples on the VirusTotal platform are also listed.

Hydra-Bite splits the permission set according to whether there is redundancy or whether it is classified or not and uses "_NE_NTS, _E_NTS, _NE_TS, _E_TS" markers in the suffix. The meaning is shown in Table 9.

(2) Permission Set Split Effect Experiment's Results and Discussion. The experimental results of cooperative application group's antikilling performance evaluation are shown in Figure 11. The detection result in Figure 11 refers to the average alarm rate of Apps in the application group. The results of each sample are marked with different symbols and colors. Four types of results are separated from the

Table 9: The meaning of the suffix of the split result.

Serial Number	Marker	Meaning
1	_NE_NTS	**No** Elimination of Redundancy & **No** Split by Taxonomy
2	_E_NTS	Elimination of Redundancy & **No** Split by Taxonomy
3	_NE_TS	**No** Elimination of Redundancy & Split by Taxonomy
4	_E_TS	Elimination of Redundancy & Split by Taxonomy

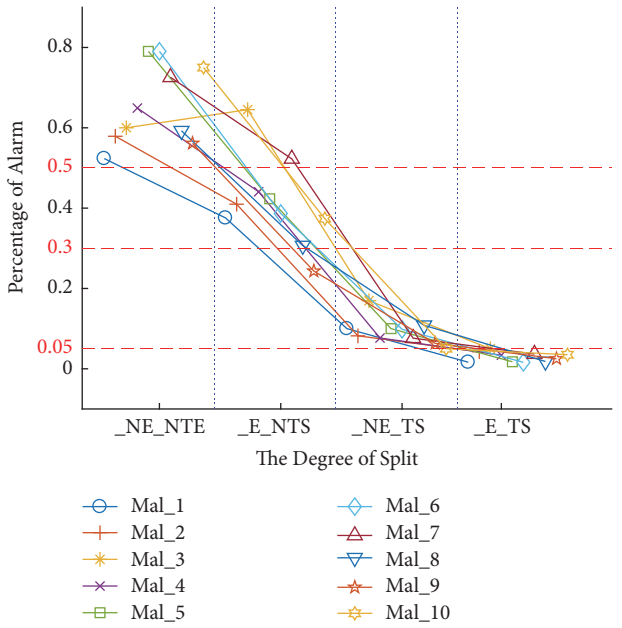


Figure 11: Alarm rate of different degree of resolution for samples.

blue vertical dotted lines. The red dotted line represents the threshold of different split methods. The following can be seen:

(1) The permission set is separately redundant or split by taxonomy, which can improve the antikilling performance of cooperative app group.

(2) The permission set is redundant and split by taxonomy, which can get the best antikilling performance of cooperative app group.

(3) The permission set is separately split by taxonomy which will enhance antikilling performance more than the result produced by being separately redundant.

The reason is that the antivirus engines are sensitive to potential information disclosure. Through simple elimination of permission set's redundancy, individual App within the cooperative application group reconstructed by HydraBite, there is still a potential risk of information disclosure for the Apps.

5.4. Antikilling Performance Experiment

(1) Antikilling Performance Experiment's Setup. In the antikilling performance experiment, our experimental samples are divided into 6 parts:

(1) Malware samples are the same as the contents of Tables 6–8.

(2) DroidBench samples are randomly selected from the DroidBench test set, and a microbenchmark suite is proposed and has been described in detail in the literature [9].

(3) Samples without taint clean and transmission directly: the sample set is shown in the 1st category, Table 10. After reading the key information, the samples do not clean the taint tagged by security methods. They send information to the external device directly through the corresponding external transmission permission. There is not collaboration between this kind of samples, so the column IPC is filled with "■."

(4) Samples with taint clean and transmission directly: the sample set is shown in the 2nd category, Table 10, adding a taint cleaning module to the previous samples. Three covert channels of volume, system log, and screen brightness are used to clean the taint. They send information to the external device directly with no IPC between them.

(5) Cooperative application samples without taint cleaning. The sample set is shown in the 3rd category, Table 10. In 1st category, the samples' permission sets are split and refactored to this sample set. The App in the group communicates directly with Intent. "_R" suffix represents an application that owns information read permissions. "_S" suffix represents an application that owns information send permissions.

(6) Cooperative application samples with taint cleaning: the sample set is shown in the 4th category, Table 10, adding a taint cleaning module to the previous samples.

Experimental environment is set as follows:

(1) *Android 6.0 Device*. The experimental sample sets are installed in the device described in Section 5.2 (1) to test whether the samples can be installed smoothly.

(2) *VirusTotal Platform*. The VirusTotal platform is mentioned in Table 6 and the literature [24]. This section uses VirusTotal to test the samples' antivirus performance.

(3) *Androguard [27]*. Carry out "permission-API" detection, and report dangerous samples of API calling combination. The Python version that builds Androguard is 2.7.10, and the mapping file is provided by PScout [13]. The running environment of Androguard: Androguard runs in Win 10 Professional Edition with the register width of 64 bits. The necessary Python, JDK, and SDK version are 2.7.10, 1.8, and 18, respectively.

(4) *FlowDroid [9]*. This tool's running environment is listed below. The system environment for the tool is Win 10 Professional Edition, with 64 bits' register width. In the system, the JDK version and Android API are 1.8. Besides, some of the official FlowDroid Support packages are unavailable

Table 10: 3–6 sample sets' attribute.

Serial Number of Type	Type of Sample	Name of Sample	Target Information	IPC Transmission Mode	Sending Mode
1	Non-Cooperative Transfer Case without Taint Cleaned	Capture1	IMEI Number	■	SMS Message
		Capture2	Location	■	Internet
		Capture3	Contact	■	SMS Message
2	Non-Cooperative Transfer Case with Taint Cleaned	SoundClean	IMEI Number	■	SMS Message
		LogClean	Location	■	Internet
		ScreenClean	Contact	■	SMS Message
3	Co_Apps without Taint Cleaned	NSound_R	IMEI Number	Intent	■
		Sound_S	■	■	SMS Message
		NLog_R	Location	Intent	■
		Log_S	■	■	Internet
		NScreen_R	Contact	Intent	■
		Screen_S	■	■	SMS Message
4	Co_Apps with Taint Cleaned	CSound_R	IMEI Number	Intent	■
		Sound_S	■	■	SMS Message
		CLog_R	Location	Intent	■
		Log_S	■	■	Internet
		CScreen_R	Contact	Intent	■
		Screen_S	■	■	SMS Message

due to an update or lack of resources. This paper changes all slf4j package provided officially into a 1.8.0 beta version.

(2) Antikilling Performance Experiment's Results and Discussions. To evaluate the surviving performance of Hydra-Bite in the real device, antivirus engine, "permission-API" mapping detection, and taint tracking detection, this section uses the above settings in Section 5.4 (1) to carry out experiments, the results are shown in Table 11. The meanings of each column in Table 11 are as follows.

"Source" column records the number of samples' information read permissions and APIs.

"Sink" column records the number of samples' transmission permissions and APIs.

"Android 6.0" column records samples and whether they triggered alarm in the installation experiment.

"VirusTotal" column records samples' alarm number on the VirusTotal platform and the hole number of participating engines.

"Androguard" and "FlowDroid" columns record experiment results by the two tools. The symbols' meanings that appear in Table 11 are described in the following.

"▲": Detect the insecure object and alert.

"▲": false positive, alert for an object that meets the security rules.

"★": Do not detect the insecure object.

"★": Detect the potential insecure object and do not alert, because there is no object to cooperate with it.

This section analyzes the experimental results in Table 11 as follows:

(1) *Malware Samples*. All these samples trigger alerts in the Android6.0 installation experiment. And their alarm rate in the VirusTotal platform is higher than other sample sets. Because of packers, obfuscation technology, Androguard and FlowDroid cannot analyze them.

(2) *DroidBench Samples*. With the obvious action of capturing key information, this sample set has a high alarm rate in VirusTotal platform and low successful installation rate in Android system.

(3) *Samples without Taint Clean and Transmission Directly*. The samples do not clean the taint tagged by security methods. And they send information to the external device directly. The above behavior triggers more alerts on the VirusTotal platform, with an average alarm rate of 29.70%. "Androguard" and "FlowDroid" also generate alarms for them.

(4) *Samples with Taint Clean and Transmission Directly*. Because the taint is cleaned, FlowDroid does not produce an alarm for it. The alarm rate triggered by this group of samples on the VirusTotal platform has been reduced, average alarm rate is 15.04%. And Androguard generates alarms for them because of no collaborative application group.

(5) *Cooperative Application Samples without Taint Cleaning*. Because the taint is not cleaned, FlowDroid produce an alarm for the information reading App.

(6) *Cooperative Application Samples with Taint Cleaning*. "Androguard" and "FlowDroid" do not generate alarms for this group of samples, because of the cooperative application and the cleaned taint. Moreover, the alarm rate caused by this set of samples is significantly reduced on the VirusTotal platform, and the average alarm rate is 5.85%.

It can be seen that, in the FlowDroid, the Source point of the send App are detected, but the FlowDroid does not alarm it. The reason is that it receives the information without

Table 11: Test results of anti-killing performance.

Sample Origin	Test Case	Source (Permission/ Sensitive API)	Sink (Permission/ Sensitive API)	Android 6.0 Warning (T/F)	VirusTotal [21] (Warnings/ Detector)	Androguard [22] (Total Permissions/ Sensitive APIs)	FlowDroid [9] (Source/Sink)
Malicious Sample (Table 5)	Mal_1	5/■	3/■	T	33/58	■	■
	Mal_2	5/■	2/■	T	33/57	■	■
	Mal_3	5/■	3/■	T	33/55	■	■
	Mal_4	5/■	1/■	T	37/57	■	■
	Mal_5	6/■	2/■	T	49/62	■	■
	Mal_6	5/■	2/■	T	49/62	■	■
	Mal_7	3/■	2/■	T	45/62	■	■
	Mal_8	4/■	2/■	T	32/54	■	■
	Mal_9	5/■	2/■	T	32/57	■	■
	Mal_10	5/■	3/■	T	42/56	■	■
DroidBench Test Case [9]	Merge1	1/1	1/1	T	27/54	4/▲ ▲	▲/▲
	DirectLeak1	1/1	1/1	T	27/55	4/▲ ▲	▲, ▲/▲
	ArrayAccess1	1/1	1/1	F	28/61	4/▲ ▲	▲, ▲/▲
	Button3	1/1	1/1	T	9/57	5/▲ ▲	▲/★
	ContentProvider1	1/1	1/1	T	25/55	4/▲ ▲	▲ ∗19, ▲/▲, ▲
	FieldSensitivity1	1/1	1/1	F	23/54	4/▲ ▲	▲, ▲/▲
	Loop1	1/1	1/1	T	33/59	4/▲ ▲	▲, ▲/▲
	ImplicitFlow1	1/1	0	F	12/57	2/▲ ★	▲ ▲/▲
	ActivityLifecycle2	1/1	1/1	T	32/62	4/▲ ▲	▲, ▲/▲
	Reflection1	1/1	1/1	T	28/55	4/▲ ▲	▲ ∗2, ▲/▲
	Echoer	1/1	1/1	T	17/62	3/▲ ★	▲/▲
	IntentSink2	1/1	1/1	T	30/61	2/▲ ▲	▲ ∗3, ▲/▲, ▲
	EventOrdering1	1/1	1/1	F	28/59	6/▲ ▲	▲, ▲/▲
	Executor1	1/1	1/1	T	21/58	4/▲ ★	▲/★
	JavaThread2	1/1	1/1	F	22/60	4/▲ ▲	▲, ▲/▲
	AsyncTask1	1/1	1/1	T	18/55	3/▲	▲ ∗2, ▲/▲
Non-Cooperative Transfer Case without Taint Cleaned (Table 10)	Capture1	1/1	1/1	F	14/59	2/▲ ▲	▲/▲
	Capture2	1/1	1/1	F	21/60	2/▲ ▲	▲/▲
	Capture3	1/1	1/1	F	17/56	2/▲ ▲	▲/▲
Non-Cooperative Transfer Case (Table 10)	SoundClean	1/1	1/1	F	10/62	4/▲ ▲	★/★
	LogClean	1/1	1/1	F	11/61	6/▲ ▲	★/★
	ScreenClean	1/1	1/1	F	13/62	5/▲ ▲	★/★
Co_Apps without Taint Cleaned (Table 10)	NSound_R	1/1	1/1	F	2/62	3/★	▲/▲
	Sound_S	1/1	1/1	F	5/60	2/★	★/★
	NLog_R	1/1	1/1	F	2/59	4/★	▲/▲
	Log_S	1/1	1/1	F	3/62	2/★	★/★
	NScreen_R	1/1	1/1	F	3/63	4/★	▲/▲
	Screen_S	1/1	1/1	F	6/57	2/★	★/★
Co_Apps with Taint Cleaned (Table 10)	CSound_R	1/1	1/1	F	2/62	3/★	★/★
	Sound_S	1/1	1/1	F	5/60	2/★	★/★
	CLog_R	1/1	1/1	F	2/59	4/★	★/★
	Log_S	1/1	1/1	F	3/62	2/★	★/★
	CScreen_R	1/1	1/1	F	3/63	4/★	★/★
	Screen_S	1/1	1/1	F	6/57	2/★	★/★

taint and sends it directly. Therefore Hydra-Bite can clean taint which is tagged by FlowDroid, it can resist detection tools based on "permission-API," and it has a high successful installation rate and a low VirusTotal alarm rate, and the results showed that Hydra-Bite method has enough threat to user privacy in antikilling performance.

6. Conclusion

In this paper, the key information disclosure through Hydra-Bite privacy leak path is researched. The purpose is to alert researchers to promote the progress of security work against collusion attacks and taint cleaning. The Hydra-Bite method is a malicious application variant that threatens user privacy in the context of application-scale operations. Hydra-Bite splits and reorganizes the traditional privacy stealing Trojans into a collaborative application group through the permission split module and uses the taint cleaning module to wash the taint tagged by the static taint tracking method on the communication entity which carrying the key information through the Android covert channel sends the cleaned key information to other devices through the collaborate sending module. The principle analysis and experimental results show that Hydra-Bite is less controlled to current security mechanisms in terms of performance and antikilling performance compared to traditional privacy stealing Trojans. Our next study will focus on improving existing static taint tracking mechanisms to tag key information with "more viscous" taints.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work presented in this paper is supported by the National Natural Science Foundation of China (nos. U1636219, 61602508, 61772549, U1736214, and 61572052), the National Key R&D Program of China (nos. 2016YFB0801303, 2016QY01W0105), Plan for Scientific Innovation Talent of Henan Province (no. 2018JR0018), and the Key Technologies R&D Program of Henan Province (no. 162102210032).

References

[1] M. Alazab, V. Monsamy, L. Batten, R. Tian, and P. Lantz, "Analysis of malicious and benign android applications," in *Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW '12)*, pp. 608–616, June 2012.

[2] H. L. Thanh, "Analysis of malware families on android mobiles: detection characteristics recognizable by ordinary phone users and how to fix it," *Journal of Information Security*, vol. 4, no. 4, pp. 213–224, 2013.

[3] A. J. Alzahrani and A. A. Ghorbani, "SMS mobile botnet detection using a multi-agent system," in *Proceedings of the the 1st International Workshop on Agents and Cyber Security*, pp. 1–8, Paris, France, May 2014.

[4] A. Castillo C, "Android malware past, present, and future," White Paper of McAfee Mobile Security Working Group, 2011.

[5] R. Schlegel, K. Zhang, X. Zhou et al., "Soundcomber: a stealthy and context-aware sound Trojan for smartphones," in *Proceedings of the Network and Distributed System Symposium (NDSS '11)*, pp. 17–33, 2011.

[6] F. Zhao, W. Shi, Y. Gan, Z. Peng, and X. Luo, "A localization and tracking scheme for target gangs based on big data of Wi-Fi locations," *Cluster Computing*, vol. 3, pp. 1–12, 2018.

[7] W. Q. Shi, X. Luo, F. Zhao, Z. Peng, Q. Cheng, and Y. Gan, "Geolocating a WeChat user based on the relation between reported and actual distance," *International Journal of Distributed Sensor Networks*, vol. 4, no. 14, 2018.

[8] W. Y. Liu, X. Y. Luo, Y. M. Liu et al., "Localization algorithm of indoor Wi-Fi access points based on signal strength relative relationship and region division," *Computers, Materials & Continua*, vol. 55, no. 1, pp. 71–93, 2018.

[9] S. Arzt, S. Rasthofer, C. Fritz et al., "Flowdroid: precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps," *ACM SIGPLAN Notices*, vol. 49, no. 6, pp. 259–269, 2014.

[10] W. Enck, M. Ongtang, and P. McDaniel, "On lightweight mobile phone application certification," in *Proceedings of 16th ACM Conference on Computer and Communications Security*, pp. 235–245, ACM, November 2009.

[11] M. Nauman, S. Khan, and X. Zhang, "Apex: extending Android permission model and enforcement with user-defined runtime constraints," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communication Security (ASIACCS '10)*, pp. 328–332, Beijing, China, April 2010.

[12] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: user attention, comprehension, and behavior," in *Proceedings of the 8th Symposium on Usable Privacy and Security (SOUPS '12)*, Washington, DC, USA, July 2012.

[13] K. W. Y. Au, Y. F. Zhou, Z. Huang, and D. Lie, "PScout: analyzing the Android permission specification," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '12)*, pp. 217–228, ACM, October 2012.

[14] E. Chin, A. P. Felt, K. Greenwood, and D. Wagner, "Analyzing inter-application communication in Android," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*, pp. 239–252, July 2011.

[15] P. P. F. Chan, L. C. K. Hui, and S. M. Yiu, "DroidChecker: analyzing android applications for capability leak," in *Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '12)*, pp. 125–136, April 2012.

[16] L. Lu, Z. Li, Z. Wu, W. Lee, and G. Jiang, "Chex: statically vetting Android apps for component hijacking vulnerabilities," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '12)*, pp. 229–240, October 2012.

[17] H. Bagheri, A. Sadeghi, J. Garcia, and S. Malek, "Covert: compositional analysis of Android inter-app permission leakage," *IEEE Transactions on Software Engineering*, vol. 41, no. 9, pp. 866–886, 2015.

[18] Y. Y. Ma, X. Y. Luo, X. Y. Li, Z. Bao, and Y. Zhang, "Selection of rich model steganalysis features based on decision rough set

α-positive region reduction," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[19] Y. Zhang, C. Qin, W. M. Zhang, F. Liu, and X. Luo, "On the fault-tolerant performance for a class of robust image steganography," *Signal Processing*, vol. 146, pp. 99–111, 2018.

[20] X. Y. Luo, X. F. Song, X. Y. Li et al., "Steganalysis of HUGO steganography based on parameter recognition of syndrome-trellis-codes," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13557–13583, 2016.

[21] V. Total, "VirusTotal-free online virus," *Malware and URL Scanner*, vol. 15, no. 2, pp. 226–241, 2012.

[22] Androguard, 2013, https://github.com/androguard/androguard.

[23] C. Marforio, H. Ritzdorf, A. Francillon, and S. Capkun, "Analysis of the communication between colluding applications on modern Smartphones," in *Proceedings of the Proceeding of the 28th Annual Computer Security Applications Conference (ACSAC '12)*, pp. 51–60, New York, NY, USA, December 2012.

[24] E. Miluzzo, M. Jadliwala, S. Balakrishnan et al., "Tapprints: your finger taps have fingerprints," in *Proceedings of the International Conference on Mobile Systems, Applications, and Services*, pp. 323–336, 2012.

[25] A. Maiti, M. Jadliwala, J. He et al., "Side-channel inference attacks on mobile keypads using smartwatches," https://arxiv.org/abs/1710.03656.

[26] S. Bugiel, L. Davi, A. Dmitrienko, S. Heuser, A. Sadeghi, and B. Shastry, "Practical and lightweight domain isolation on Android," in *Proceedings of the the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, p. 51, Chicago, Ill, USA, October 2011.

[27] R. A. Kemmerer, "Shared resource matrix methodology: an approach to identifying storage and timing channels," *ACM Transactions on Computer Systems*, vol. 1, no. 3, pp. 256–277, 1983.

[28] T. Reps, S. Horwitz, and M. Sagiv, "Precise interprocedural dataflow analysis via graph reachability," in *Proceedings of the 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 49–61, January 1995.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5938152, 11 pages
https://doi.org/10.1155/2018/5938152

Research Article

Task-Oriented Multilevel Cooperative Access Control Scheme for Environment with Virtualization and IoT

Jian Dong,[1,2] Hui Zhu ,[1] Chao Song,[1] Qiang Li,[3] and Rui Xiao[1]

[1]*State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710017, China*
[2]*Science and Technology on Communication Networks Laboratory, Shijiazhuang 050081, China*
[3]*Hangzhou Research Institute, NetEase Network Co., Ltd., Hangzhou 310000, China*

Correspondence should be addressed to Hui Zhu; zhuhui@xidian.edu.cn

Received 9 March 2018; Revised 16 May 2018; Accepted 23 June 2018; Published 10 July 2018

Academic Editor: Kim-Kwang Raymond Choo

With the development of cloud computing technology and the proliferation of the Internet of Things (IoT) terminals, more and more scenes need the collaboration of virtual machines and IoT terminals to resolve. However, there are many severe challenges on the security of virtual machines and IoT terminals. Based on Bell-LaPadula Model (BLP), a task-oriented multilevel cooperative access control scheme virtualization and reality BLP, named VR-BLP, is proposed. Specifically, tasks are created for each user of the platform and tasks and users are divided into multiple levels to provide more granularities to limit access between virtual machines and IoT terminals. Moreover, with network isolation cooperating with process isolation and shared memory isolation mechanisms, VR-BLP is implemented to enhance the security isolations between tasks. Performance evaluations show that VR-BLP enhanced the security of environment with virtualization and IoT without causing significant performance penalty.

1. Introduction

Since it was created in 2006, cloud computing [1–3] has been growing more and more popular in today's society. With wide popularity and broad application, it is playing a more and more important role in the development of the society, not only making enterprises get more income while saving more costs, but also providing more convenient online services for ordinary consumers. As the basis of cloud computing, virtualization [4–6] technologies including system virtualization and network virtualization have been widely used on cloud platform, helping cloud computing to develop faster and faster. On the other hand, with more and more connected devices expected to be in use, the Internet of Things (IoT) [7] has become a critical focus area for many enterprises.

To address security challenges in cloud environment, many access control schemes are proposed, such as RBAC [8] and TBAC [9]. These schemes partially enhance the security of access control in cloud but do not apply to resource management for environment with virtualization and IoT. When integrating virtualization and IoT to build a bigger platform which could allocate virtual machines and IoT terminals to users at the same time, enterprises and consumers could benefit more from virtualization and IoT [10]. For service provider enterprises, the hybrid platform could provide more kinds of combinations of services to clients, save the costs to build two kinds of platforms which are cloud computing platforms and IoT platforms, and achieve more efficient utility of the resources and the energies. For service buyer enterprises, the hybrid platform could provide more kinds of services meeting the needs of their own business, saving their money and energies to maintain two kinds of systems. For ordinary consumers, they could buy more flexible service according to their needs and budget. In short, it is of great use to integrate virtualization and IoT together.

Figure 1 shows a scene in which there exist multiple users and multiple tasks in a hybrid virtualization and IoT platform. In the hybrid platform, computing and storage resources form virtual resource pool through virtualization technologies, and IoT terminals form IoT terminal resource pool through network connections and network equipment form network equipment resource pool. Authorized users of the system could create a set of tasks and every task

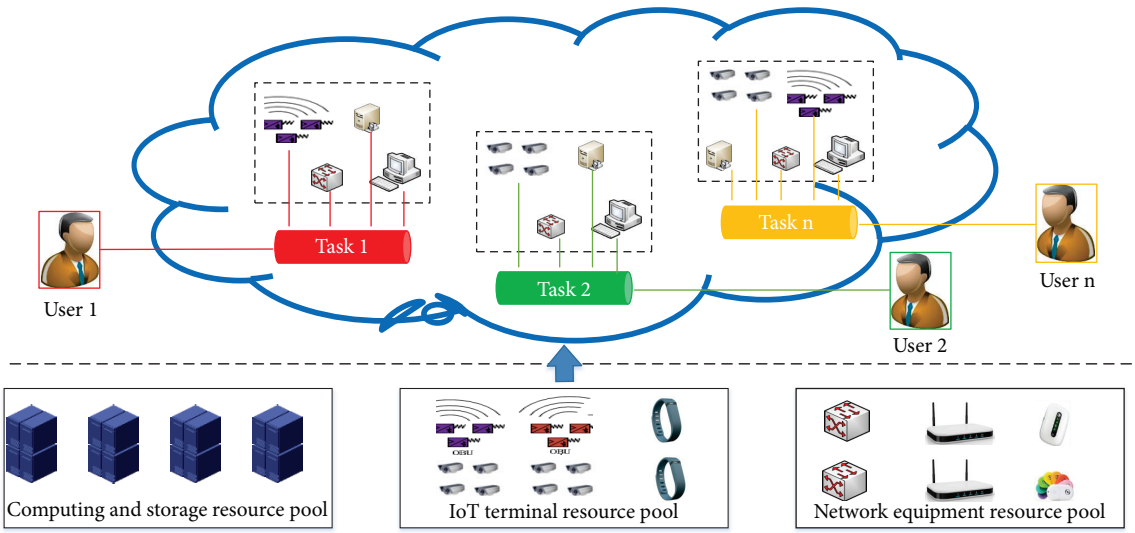


Figure 1: Multiuser and multitask scenes over virtualization and IoT.

contains a set of resources from resources pool. However, the integration of virtualization and IoT results in security issues which could become a critical problem. Virtualization and IoT may incur new vulnerabilities to the hybrid virtualization and IoT platform [11]. Even though virtualization and IoT security have attracted considerable interest in recent years and several solutions have been proposed, the flourish of secure solutions for virtualization and IoT still faces many challenges in the balance between information sharing and privacy preservation [12]. Security issues on virtualization and IoT have been the vital barrier to the development of the integration of virtualization and IoT.

In order to construct a task-oriented secure isolation mechanism for environment with virtualization and IoT, a new multilevel cooperative access control scheme named VR-BLP is proposed in this paper. And our main contributions are threefold.

(1) A task-oriented multilevel cooperative access control scheme for environment with virtualization and IoT, named VR-BLP, is proposed to enhance the security isolations between users and tasks.

(2) Network isolation cooperates with process isolation and shared memory isolation to enhance security isolations between virtual machines and IoT terminals.

(3) Performance evaluations show that VR-BLP is an efficient multilevel access control scheme for environment with virtualization and IoT.

The remainder of this paper is organized as follows. In Section 2, we survey the related works. In Section 3, we introduce the preliminaries of this paper. In Section 4, we present the architecture of our proposed VR-BLP scheme and provide the security proof of the VR-BLP scheme. Then, we present the implementation in Section 5, followed by evaluations in Section 6. Finally, we draw our conclusions in Section 7.

2. Related Work

The BLP [13, 14] model is the first access model in the sense of mathematical which was proposed by Bell and LaPadula in 1973. It is a state machine model based on a simulated military security strategy. With the BLP model becoming more and more popular, it was applied to a lot of different scenes to achieve more secure isolations of resources. Multiple level security (MLS) [15] has always been a focus of attention since the usage of computers in military and intelligence systems. With the development of cloud computing, scholars have done a lot of researches about the multiple level security of clouds.

A MUSHI system [16] toward multiple level security cloud with strong hardware level isolation was designed to provide hardware level isolation and protection to individual guest virtual machine (VM) execution. With MUSHI, a user could maintain confidentiality and integrity of his/her VM in a multicore environment even in the presence of malicious attacks from both within and outside the cloud infrastructure. A BLP-based multilevel security model [17] of workflow deployment over an architecture for federated clouds on the background of medical data security was proposed, which divided the medical service transactions into several states, and one state can be transformed to another by specific operations. By defining security operations, the model proves the system could keep secure. A BLP-based multilevel security model [18] for private cloud was proposed and was proved secure by mathematical method. The model used mandatory access control method to control user's operation and can guarantee that users cannot leak sensitive data after they read them. A Centralized Pervasive Computing Environment/Multilevel Security (CPCE/MLS) system [19] was designed to provide the security guarantee of the pervasive computing environment by introducing the server-storage terminals and implementing the multilevel security access control mechanism based on BLP model, process creation

supervision, and an auditing mechanism. A multilevel secure file sharing server [20] was designed to satisfy the requirements for certifiable, scalable, and multilevel cloud security and this paper showed how the secure file server can be used to create a high-assurance, MLS storage cloud. The file server was built on mature technology that was previously certified and deployed across domains and that supports high-performance, low-to-high, and high-to-low file sharing with verifiable security.

Though these works are very meaningful and valuable, they do not consider the security issues in a task-oriented environment with virtualization and IoT. To satisfy the requirements for environment with virtualization and IoT, a task-oriented cooperative access scheme VR-BLP is proposed and implemented with network isolation cooperating with process isolation and shared memory isolation mechanisms, to enhance the security isolations between tasks.

3. Preliminaries

In this section, we will introduce the knowledge related to the design and the implementation of our VR-BLP scheme.

3.1. BLP. While giving the definition of the subject, the object, the security level function, the state, the state transition rules, and so on, BLP model defines that a system is secure if and only if the system always satisfies the simple security property, the $*$-property, and the discretionary security property. Given a secure initial state, the state of the system retains security if every state transition satisfies the simple security property, the $*$-property, and the discretionary property. Through managing these state transitions, BLP prevents the leakage of confidential information in the process of information sharing. The subject of higher clearance level and larger category could access to the object with lower classification and smaller category with certain access attributes. The attributes include the read access, the write access, the append access, the execute access, and the control access. About the state transition rules, the BLP model designs a set of request elements that the subjects could make, including get, give, release, rescind, change, create, and delete. The BLP model develops a set of rules which are proof to satisfy the simple security property and the $*$-property. Therefore, a system which has implemented the BLP model is secure and could protect the privacy between information sharing.

3.2. LSM. LSM [21] is a framework in the Linux kernel that supports various computer security models and LSM has nothing to do with any separate security implementation. This framework is licensed under the GNU General Public License and it has been a part of the official Linux kernel since Linux 2.6. By providing a general purpose framework for security policy modules, LSM allows many different access control schemes to be implemented as loadable kernel modules and hence enables these security policies to develop independently. A quantity of existing access control implementations, including SELinux, Domain, and Type Enforcement (DTE) and Linux Intrusion Detection System (LIDS) has already been adapted to use the LSM framework.

3.3. SDN. SDN [22–24] uses stratification ideas to separate data and control. The control layer mainly includes a logic-centric and programmable controller so that it can grasp the global network information and it is convenient for operators and researchers to manage network configurations and the deployment of new protocols. There is a switch at the data layer which provides simple data forwarding to match data packets, so it can be quickly processed to meet the growing demands for traffic. The two layers use an open unified interface to interact. The controller sends unified standard rules to the switch through standard interfaces. The switch only needs to act according to these rules. Therefore, the SDN technology can effectively reduce the equipment load, help network operators control the infrastructure, and reduce overall operating costs. It has gradually become one of the most promising network technologies.

4. VR-BLP Scheme

In this section, we redefine the architecture, elements, security properties, and state transition rules of the BLP model for better applying to the circumstance of the environment with virtualization and IoT. And we propose a task-oriented multilevel access control scheme for environment with virtualization and IoT to construct a secure isolation between users and tasks.

4.1. Elements of VR-BLP Model. Before introducing the formal definitions of the task-oriented multilevel access control scheme for virtualization and IoT, we define the basic elements of the VR-BLP model as below.

Subject and Object. In VR-BLP model, one user is a subject and one task is an object. A user creates a set of tasks and each task creates a set of resources. Through mapping resources to tasks, users access resources by accessing tasks.

Security Functions Sets. F is a security function set and an arbitrary element of F contains three components $f_m(s)$, $f_c(s)$, and $f(o)$. $f_m(s)$ represents the highest clearance level of the subject s, $f_c(s)$ represents the current clearance level of the subject s, and $f(o)$ represents the classification of the object o.

Access Attribute Set. The access attribute set A contains three elements, $A = \{r, a, w\}$, where r stands for read-only, a stands for write-only, and w stands for read-write. A subject could only access an object with an access attribute included in the access attribute set A.

Access Matrix. The access matrix M contains a set of access attributes which record how a subject could access to an object. A subject could only access an object with an access attribute stored in the access matrix.

Current Access Set. $B = S \times O \times A$ represents the current set. An arbitrary element of B is written b. $b = (s_i, o_j, x)$ indicates the subject s_i has access to the object o_j in mode $x \subseteq m_{ij}$.

Current State of the System. $v = (b, M, f) \in V$ stands for an arbitrary state of the system in which $b = (s_i, o_j, x) \in B, x \subseteq m_{ij}, f = (fs, fo) \in F$.

Request Elements. The request elements include read-only, write-only, and read-write.

Requests. The requests $R = S \times RA \times O \times X$, where $X = A \cup F$. An arbitrary element of R is written R_k.

Decisions. The decisions include yes and no which stand for the decisions that rules make.

Relation. W stands for a relation that will be the union of partial functions which constitute the rules of operation of the system with respect to preservation of the simple security property and the $*$-property.

Rules. A rule is a function $\rho : \mathrm{R} \times V \longrightarrow D \times V$ that represents what the response and the state change are when a request and a state are inputted. It decides that what the system should react to the request of users and makes the system change according to the specific situation.

4.2. Security Properties

(1) Simple Security Property. The state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f)$ satisfies the simple security property, if and only if for $\forall(s, o, x) \in b$

(i) $x = a$

(ii) $x = r$ or $x = w \Longrightarrow f_m(s) \geq f(o)$

The simple security property means that, for $\forall(s, o, x) \in b$, when the subject s tries to access the object o with the access attribute a, the state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f)$ satisfies the simple security property. When the subject s tries to access the object o with the access attribute r, the state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f)$ satisfies the simple security property if the highest clearance level of the subject s is greater than or equal to the classification of the object o. When the subject s tries to access the object o with the access attribute w, the state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f)$ satisfies the simple security property if the highest clearance level of the subject s is greater than or equal to the classification of the object o.

(2) $*$-Property. The state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f)$ satisfies $*$-property, if and only if $\forall(s, o, x) \in b, s \in S/S^*$, where S^* is trusted subjects

(i) $x = r \Longrightarrow f_c(s) \geq f(o)$

(ii) $x = a \Longrightarrow f_c(s) \leq f(o)$

(iii) $x = w \Longrightarrow f_c(s) = f(o)$

The $*$-property means that, for $\forall(s, o, x) \in b$, $s \in S/S^*$, where S^* is trusted subjects, when the subject s tries to access the object o with the access attribute r, the state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f)$ satisfies the $*$-property if the current clearance level of the subject s is greater than or equal to the classification of the object o. When the subject s tries to access the object o with the access attribute a, the state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f)$ satisfies the $*$-property if the current clearance level of the subject s is lesser or equal to the classification of the object o. When the subject s tries to access the object o with the access attribute w, the state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f)$ satisfies the $*$-property if the current clearance level of the subject s is equal to the classification of the object o.

(3) Discretionary Security Property. The state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f)$ satisfies discretionary property, if and only if $\forall(s_i, o_j, x) \in b$, $x \in m_{ij}$.

(4) Fundamental Security Property. The system is secure if and only if the initial state of the system is secure and every state transition satisfies the simple security property, the $*$-property, and the discretionary property.

In our scheme, the subjects are user and the objects are task. Tasks are created by users and are assigned with different classifications. Users have different clearance levels given by the system as we designed. Our purpose is to make users with higher clearance levels able to access tasks with lower classifications. While tasks contain a set of resources, users with higher clearance levels could actually read data from resources with lower classifications, whether these resources are virtual machines or IoT terminals. Users with higher clearance levels could not write data to locations that belong to resources with lower classifications, whether these resources are virtual machines or IoT terminals. On the contrary, users with lower clearance levels could not read data from resources with higher classifications, whether these resources are virtual machines or IoT terminals. Users with lower clearance levels could write data to locations that belong to resources with higher classifications, whether these resources are virtual machines or IoT terminals.

4.3. State Transition Rules.

A rule is a function $\rho : \mathrm{R} \times V \longrightarrow D \times V$. As shown in Figure 2, the interpretation of a rule is that, given a request and a state, a rule decides a response and a state change. R is a set of request; D is a set of decisions. The result of decisions is one of the sets of (yes, no). Yes represents that the request is allowed to execute, and No represents that the request is denied to execute.

R_k is an arbitrary element of R, D_m is an arbitrary element of D, and a rule ρ is security preserving if and only if $\forall(R_k, v) \in R \times V$, $\rho(R_k, v) = (D_m, v^*)$ and v is secure $\Longrightarrow$ v^* is secure. Based on operations for environment with virtualization and IoT, we defined three state transition rules.

Rule 1. Read-only:

Given state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f) \in \mathrm{V}$, $s \in S \setminus S^*$,where S^* is trusted subjects, the handling process to request $R_k = (s_i, o_j, r)$ is as follows.

If $f_m(s_i) \geq f(o_j) \wedge r \in m_{ij}$, then set $f^* = f \cup \{\max(f_c(s_i), f(o_j))\}$, $b^* = b \cup \{(s_i, o_j, r)\}$

$$\rho 1(R_k, v) = (yes, (b^*, M, f^*))$$

Else

$$\rho 1(R_k, v) = (no, (b, M, f))$$

Rule 1 describes what the response is and what the system state is when the subject s_i requests to access the object o_j with access attribute r. When the request is submitted, the access control modules check if the request is allowed. First, the access control modules check the highest clearance level of the subject s_i and compare it with the classification of the object o_j.If the highest clearance level of the subject s_i is greater than or equal to the classification of the object o_j, then the access control modules check if the access attribute r satisfies the requirement $r \in m_{ij}$. Only when the two requirements are satisfied, the request is allowed to execute. Otherwise, the request is denied. If the request is allowed to execute, the access control modules calculate the next

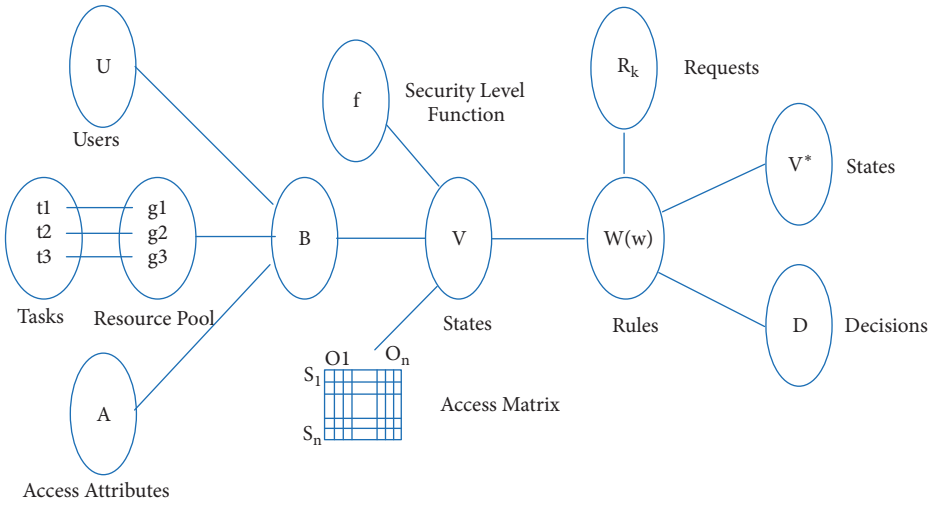


Figure 2: Proposed VR-BLP scheme.

system state. As a result of Rule 1, the system state $v = (\mathrm{b}, \mathrm{M}, f)$ becomes $v^* = (b^*, M, f^*)$, where $f^* = f \cup \{\max(f_c(s_i), f(o_j))\}$, $b^* = b \cup \{(s_i, o_j, r)\}$. The new security function generated after Rule 1 executes is the greater one of the two security functions $f_c(s_i)$ and $f(o_j)$. The new security function sets f^* is the combinations of the previous security function sets f and the greater one of the two security functions $f_c(s_i)$ and $f(o_j)$. The new current access set b^* is the combinations of the previous current access sets b and the present current access set (s_i, o_j, r). Finally, if Rule 1 is allowed to execute, the response is yes and the system state $v = (\mathrm{b}, \mathrm{M}, f)$ becomes $v^* = (b^*, M, f^*)$, where $f^* = f \cup \{\max(f_c(s_i), f(o_j))\}$, $b^* = b \cup \{(s_i, o_j, r)\}$.

Rule 2. Write-only:

Given state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f) \in \mathrm{V}$, $s \in S \setminus S^*$,where S^* is trusted subjects, the handling process to request $R_k = (s_i, o_j, a)$ is as follows.

If $f_m(s_i) \geq f(o_j) \wedge f_c(s_i) \leq f(o_j) \wedge a \in m_{ij}$, then set $b^* = b \cup \{(s_i, o_j, a)\}$

$$\rho 2(R_k, v) = (yes, (b^*, M, f))$$

Else

$$\rho 2(R_k, v) = (no, (b, M, f))$$

Rule 2 describes what the response is and what the system state is when the subject s_i requests to access the object o_j with access attribute a. When the request is submitted, the access control modules check if the request is allowed. First, the access control modules check the highest clearance level of the subject s_i and compare it with the classification of the object o_j.If the highest clearance level of the subject s_i is greater than or equal to the classification of the object o_j, then the access control modules check if the current clearance level of the subject s_i is lesser or equal to the classification of the object o_j. If the current clearance level of the subject s_i is lesser or equal to the classification of the object o_j, then the access control modules will check if the access attribute a satisfies the requirement $a \in m_{ij}$. Only when the three requirements are satisfied, the request is allowed to execute. Otherwise, the request is denied. If the request is allowed to execute, the access control module will calculate the next system state. As a result of Rule 2, the system state $v = (\mathrm{b}, \mathrm{M}, f)$ becomes $v^* = (b^*, M, f)$, where $b^* = b \cup \{(s_i, o_j, a)\}$. The new current access sets b^* is the combinations of the previous current access sets b and the present current access set (s_i, o_j, a). Finally, if Rule 2 is allowed to execute, the response will be yes and the system state $v = (\mathrm{b}, \mathrm{M}, f)$ becomes $v^* = (b^*, M, f)$, where $b^* = b \cup \{(s_i, o_j, a)\}$.

Rule 3. Read-Write:

Given state $\mathrm{v} = (\mathrm{b}, \mathrm{M}, f) \in \mathrm{V}$, $s \in S \setminus S^*$, where S^* is trusted subjects, the handling process to request $R_k = (s_i, o_j, r)$ is as follows.

If $f_m(s_i) \geq f(o_j) \wedge f_c(s_i) \leq f(o_j) \wedge w \in m_{ij}$, then set $f^* = f \cup \{f(o_j)\}$, $b^* = b \cup \{(s_i, o_j, w)\}$

$$\rho 3(R_k, v) = (yes, (b^*, M, f^*))$$

Else

$$\rho 3(R_k, v) = (no, (b, M, f))$$

Rule 3 describes what the response is and what the system state is when the subject s_i requests to access the object o_j with access attribute a. When the request is submitted, the access control modules check if the request is allowed. First, the access control modules check the highest clearance level of the subject s_i and compare it with the classification of the object o_j.If the highest clearance level of the subject s_i is greater than or equal to the classification of the object o_j, then the access control modules check if the current clearance level of the subject s_i is lesser or equal to the classification of the object o_j. If the current clearance level of the subject s_i is lesser or equal to the classification of the object o_j, then the access control modules will check if the access attribute w satisfies the requirement $w \in m_{ij}$. Only when the three requirements are satisfied, the request is allowed to execute. Otherwise, the

request is denied. If the request is allowed to execute, the access control modules calculate the next system state. As a result of Rule 3, the system state $v = (\mathrm{b}, \mathrm{M}, f)$ becomes $v^* = (b^*, M, f^*)$, where $f^* = f \cup \{f(o_j)\}, b^* = b \cup \{(s_i, o_j, w)\}$. The new security function generated after Rule 1 executes is the security function $f(o_j)$. The new security function sets f^* is the combinations of the previous security function sets f and the security function $f(o_j)$. The new current access sets b^* is the combinations of the previous current access sets b and the present current access set (s_i, o_j, r). Finally, if Rule 3 is allowed to execute, the response is yes and the system state $v = (\mathrm{b}, \mathrm{M}, f)$ becomes $v^* = (b^*, M, f^*)$, where $f^* = f \cup \{f(o_j)\}$, $b^* = b \cup \{(s_i, o_j, w)\}$.

4.4. Security Proof of the Model

Proof (Rule 1 keeps secure). Let the initial state $v = (b, M, f) \in$ V be secure. For the request $R_k = (s_i, o_j, r) \in R$, after executing Rule 1, we get a new state v^*; then $v^* = v$ or $v^* = (b^*, M, f^*)$. If $v^* = v$, then v^* is secure since v is secure, else if $v^* = (b^*, M, f^*)$: $v^* - v = ((s_i, o_j, r), M, f^*)$, where $f^* = f \cup \{\max(f_c(s_i), f(o_j))\}$, and $f_m(s_i) \geq f(o_j)$, so $v^* - v$ satisfies the simple security property. Since v satisfies the simple security property, v^* satisfies the simple security property. Since $f^* - f = \max(f_c(s_i), f(o_j)) \geq f(o_j)$, then $v^* - v$ satisfies the star-property. From the above, we prove that Rule 1 keeps secure. □

Proof (Rule 2 keeps secure). Let the initial state $v = (b, M, f) \in$ V be secure. For the request $R_k = (s_i, o_j, r) \in R$, after executing Rule 2, we get a new state v^*, then $v^* = v$ or $v^* = (b^*, M, f)$. If $v^* = v$, then v^* is secure since v is secure, else if $v^* = (b^*, M, f)$: $v^* - v = ((s_i, o_j, r), M, f)$, where $f_m(s_i) \geq f(o_j)$, so $v^* - v$ satisfies the simple security property. Since v satisfies the simple security property, v^* satisfies the simple security property. Since $f_c(s_i) \leq f(o_j)$, then $v^* - v$ satisfies the star-property. Since v satisfies the star-property, v^* satisfies the star- property. From the above, we proof that Rule 2 keeps secure. □

Proof (Rule 3 keeps secure). Let the initial state $v = (b, M, f) \in$ V be secure. For the request $R_k = (s_i, o_j, r) \in R$, after executing Rule 1, we get a new state v^*; then $v^* = v$ or $v^* = (b^*, M, f^*)$. If $v^* = v$, then v^* is secure since v is secure, else if $v^* = (b^*, M, f^*)$: $v^* - v = ((s_i, o_j, r), M, f^*)$, where $f^* = f \cup \{f(o_j)\}$, and $f_m(s_i) \geq f(o_j)$, so $v^* - v$ satisfies the simple security property. Since v satisfies the simple security property, v^* satisfies the simple security property. Since $f^* - f = f(o_j)$, then $v^* - v$ satisfies the star-property. From the above, we prove that Rule 3 keeps secure. □

5. Implementation of VR-BLP

This section will mainly introduce the application scenarios of VR-BLP scheme and the specific design architecture of our scheme.

5.1. Application Scenarios. As shown in Figure 3, users are divided into different groups with different clearance levels and each user creates a set of tasks allocated with classifications the same as the user's clearance level. The policy making module guides security label module how to add different security label to different tasks according to the tasks' classifications and it notices the access control module to execute state transition rules to isolate resources.

5.2. System Modules. The system modules consist of three parts: the policy making module, the security label module, and the access control module.

5.2.1. Policy Module. Users are assigned with different clearance levels according to the needs of the design, and tasks are assigned with classifications which are equal to the clearance level of the user who create those tasks. Since tasks contain a set of resources, resources are mapping to tasks resulting in that users achieve the multilevel security access control of resources. Therefore, the security label module adds different security labels to tasks and the access control modules isolate resources through network isolation, memory isolation, and process isolation. The policy making module makes policies and send these policies to the security module and the access control module to guide them to isolate resource meeting our expectations.

5.2.2. Security Label Module. The security label module receives security policies from the policy making module and attach security labels to tasks and resources. For tasks, they are attached with security labels according to their classifications. For resources, they are attached with the same security labels as the task which creates them. In this way, resources could be isolated from others by access control module through their security labels easily.

5.2.3. Access Control Module. The access control module receives policies from the policy making module and achieves the isolation of resources through process isolation, shared memory isolation, and network isolation. For process isolation and shared memory isolation, they are implemented by a security module named KMAC that we designed based on LSM. Through attaching tasks' security labels to virtual machines and virtual machines' disk images, KMAC does not allow one virtual machine request to access a disk whose security label is different from the virtual machine's security label. Through attaching tasks' security labels to virtual machines and virtual machines' shared memory, KMAC does not allow one virtual machine request to access a shared memory whose security label is different from the virtual machine's security label. For network isolation, we use SDN and traditional network technologies to isolate resources. When a user requests to access a task with access attribute read-only, write-only, or read-write, the request is analyzed by the security switch and the ACL rules in the security switch decides if the request is legal. According to decisions that the security switch makes, the request is allowed or denied to execute.

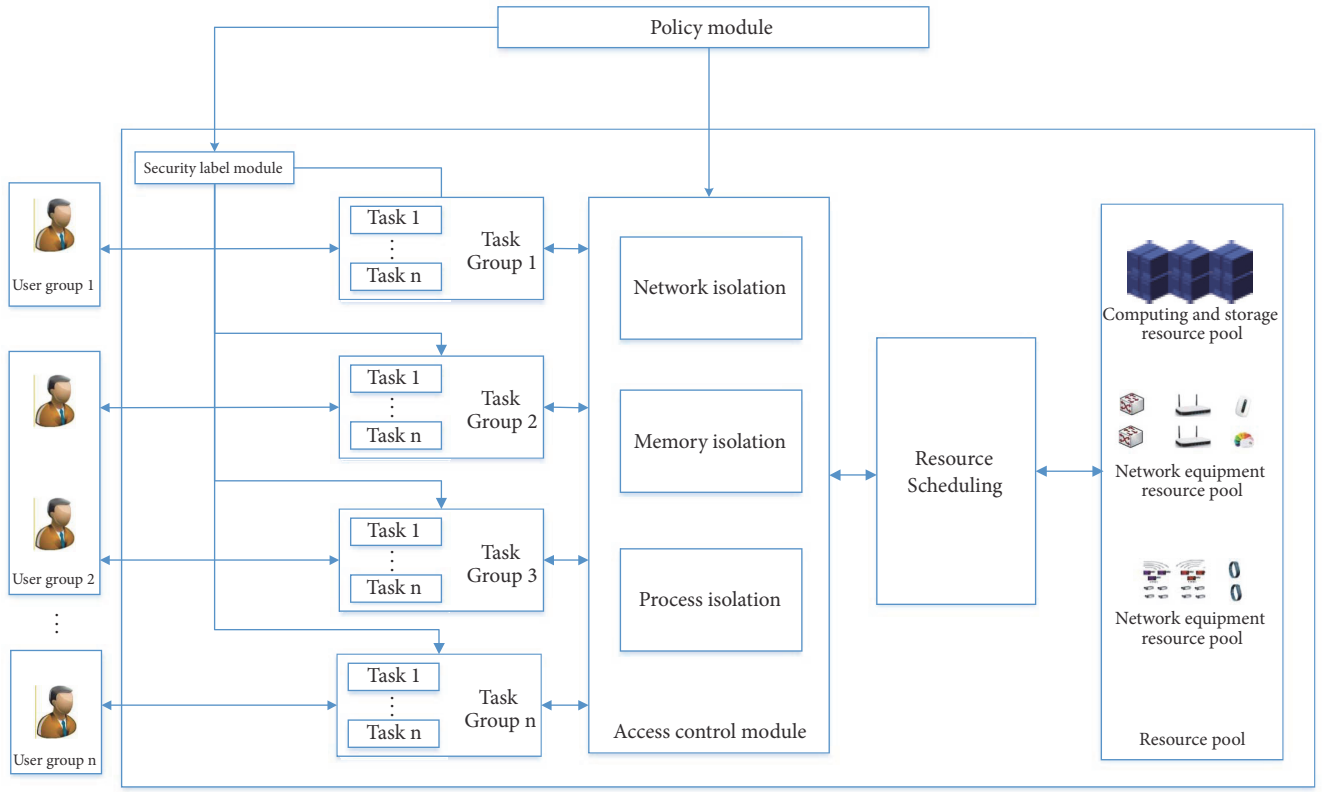


Figure 3: Application framework of VR-BLP.

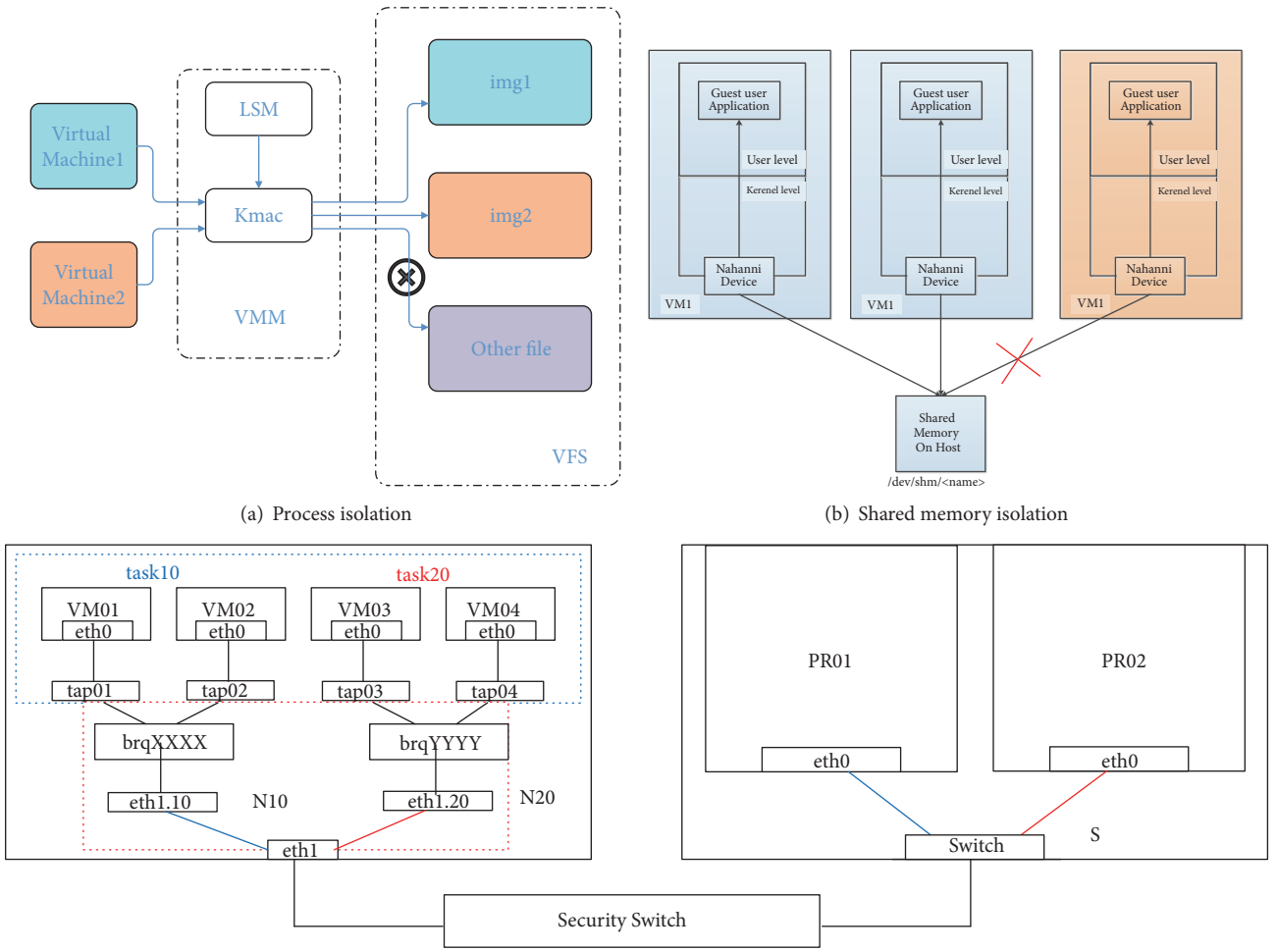
To strengthen the security isolation of virtual machines, a KMAC module based on LSM is designed. For security reason, KMAC module must be compiled into the Linux kernel as a LSM module. KMAC strengthen the security isolation of the process and the shared memory between virtual machines. As shown in Figure 4(a), a virtual machine is a QEMU process in the host machines. To strengthen the security isolation between QEMU processes, the disk images of each process are isolated. When a QEMU process starts, KMAC module allocates the same unique security label to the QEMU process and its disk image. After that, the QEMU process cannot access other processes with different security labels. As shown in Figure 4(b), mandatory access control of shared memory between virtual machines is implemented in the KMAC module. By using ivshmem, a virtual PCI device is added to a virtual machine to create a piece of shared memory. By using inode_create and file_mmap function of the LSM module, we could control virtual machines' access to shared memory. When a QEMU process starts, KMAC module allocates the same unique security label to the QEMU process and its shared memory. After that, the QEMU process cannot access other processes' shared memory with different security labels.

As shown in Figure 4(c), by using SDN, virtual machines of task10 are attached to virtual network N10 with a VLAN id 10 on the OpenStack platform. Virtual machines of task20 are attached to virtual network N20 with a VLAN id 20. For physical resources, VLAN10 and VLAN20 are created on the physical layer 3 switch S. Physical resources of task10 are attached to VLAN10 and physical resources of task20 are attached to VLAN20. Resources in VLAN10 could only communicate with those resources in VLAN10 and resources in VLAN20 could only communicate with those resources in VLAN20. Therefore, Resources in task10 could only communicate with those resources in task10 and resources in task20 could only communicate with those resources in task20. The security switch receives policies from the policy making module and make corresponding ACL rules to control users' request to tasks. According to the state transition rules in VR-BLP scheme, the security switch gets users' request and makes decisions.

6. Evaluation of VR-BLP

6.1. Environments of Evaluation. As shown in Figure 5, two workstations which have 32 cores, 32GB RAM, and dual network interface card serve as a controller node and a compute node of the OpenStack platform. Four PCs with wireless network adapters connected by wireless routers are used to simulate IoT terminals. A switch connects the wireless router with two workstations.

6.2. Device Configuration for Evaluation. In Figure 6(a), four tasks of one user are created and task1, task2, task3, and task4 are assigned with security levels 1, 2, 3, and 4 respectively, while security level $4 > 3 > 2 > 1$. Each task contains two



(a) Process isolation

(b) Shared memory isolation

(c) Network isolation

Figure 4: Resources isolation.

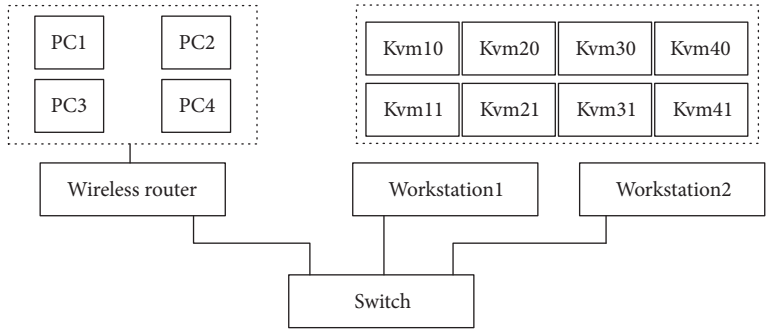


Figure 5: Topology of test environment.

KVM virtual machines and a PC and these resources are all allocated with a unique IP address. Resources of task1 are in the VLAN101 network. Resources of task2 are in the VLAN102 network. Resources of task3 are in the VLAN103 network. Resources of task4 are in the VLAN104 network. In Figure 6(b), four users are assigned with security levels 1, 2, 3, and 4, respectively. User1, user2, user3, and user4 all create a task1 with the same security level within one user group. Each task1 contains two KVM virtual machines and a PC. These resources are all allocated with a unique IP address. Resources of user1 are in VLAN105 network. Resources of task2 are in VLAN201 network. Resources of task3 are in VLAN301 network and resources of task4 are in VLAN401 network.

Tasks		IP Address	VLAN ID
Task1	Kvm10	10.1.1.13	101
	Kvm11	10.1.1.4	
	PC1	10.1.1.14	
Task2	Kvm20	10.1.2.29	102
	Kvm21	10.1.2.17	
	PC2	10.1.2.61	
Task3	Kvm30	10.1.3.18	103
	Kvm31	10.1.3.23	
	PC3	10.1.3.34	
Task4	Kvm40	10.1.4.51	104
	Kvm41	10.1.4.67	
	PC4	10.1.4.34	

(a) Policies of access control between different tasks

Users			IP Address	VLAN ID
User1	Task1	Kvm10	10.1.1.12	105
		Kvm11	10.1.1.3	
		PC1	10.1.1.10	
User2	Task1	Kvm20	10.2.1.9	201
		Kvm21	10.2.1.14	
		PC2	10.2.1.6	
User3	Task1	Kvm30	10.3.1.8	301
		Kvm31	10.3.1.15	
		PC3	10.3.1.11	
User4	Task1	Kvm40	10.4.1.5	401
		Kvm41	10.4.1.19	
		PC4	10.4.1.16	

(b) Policies of access control between different users

Figure 6: Policies of test environment.

6.3. Result Analysis. In this section, the above isolation experiments are conducted. The check mark in Tables 1 and 2 means that one virtual machine or PC could access another virtual machine or PC. As shown in Table 1, resources of task1 could access resources of task1. Resources of task2 could access resources of task1 and task2. Resources of task3 could access resources of task1, task2, and task3. Resources of task4 could access resources of task1, task2, task3, and task4. Therefore, the experimental results show that the VR-BLP scheme achieves multilevel security isolation between different tasks.

As shown in Table 2, resources of user1 could access resources of user2. Resources of user2 could access to resources of user1 and user2. Resources of user3 could access resources of user1, user2, and user3. Resources of user4 could access resources of user1, user2, user3, and user4. Therefore, the experimental results show that the VR-BLP scheme achieves multilevel security isolation between different users.

At last, as shown in Figure 7, we test the impacts of VR-BLP on the performance of the host and the guest in the case of single CPU and double CPU, respectively, using UnixBench [25]. As we can see from the results, our scheme has little impact on the performance of the host and the guest with an overhead of no more than 2%.

7. Conclusion

In this paper, a task-oriented multilevel cooperative access control scheme based on BLP, named VR-BLP, has been proposed. Specifically, the access control between virtual machines and IoT terminals achieves fined-grained isolation by dividing tasks and users into multiple levels. Moreover, VR-BLP enhances the security isolations between tasks through the cooperation of network isolation, process isolation, and shared memory isolation. Performance evaluations show that VR-BLP enhanced the security of environment with virtualization and IoT with only a small performance loss.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Table 1: Access control between different tasks.

Tasks		Access to Tasks			
		Task1	Task2	Task3	Task4
Task1	Kvm10	√			
	Docker11	√			
	PC12	√			
Task2	Kvm20	√	√		
	Docker21	√	√		
	PC22	√	√		
Task3	Kvm30	√	√	√	
	Docker31	√	√	√	
	PC32	√	√	√	
Task4	Kvm40	√	√	√	√
	Docker41	√	√	√	√
	PC42	√	√	√	√

Table 2: Access control between different users.

Users			Access to Users			
			User1	User2	User3	User4
User1	Task1	Kvm10	√			
		Docker11	√			
		PC12	√			
User2	Task1	Kvm20	√	√		
		Docker21	√	√		
		PC22	√	√		
User3	Task1	Kvm30	√	√	√	
		Docker31	√	√	√	
		PC32	√	√	√	
User4	Task1	Kvm40	√	√	√	√
		Docker41	√	√	√	√
		PC42	√	√	√	√

Conflicts of Interest

The authors declare that they have no conflicts of interest.

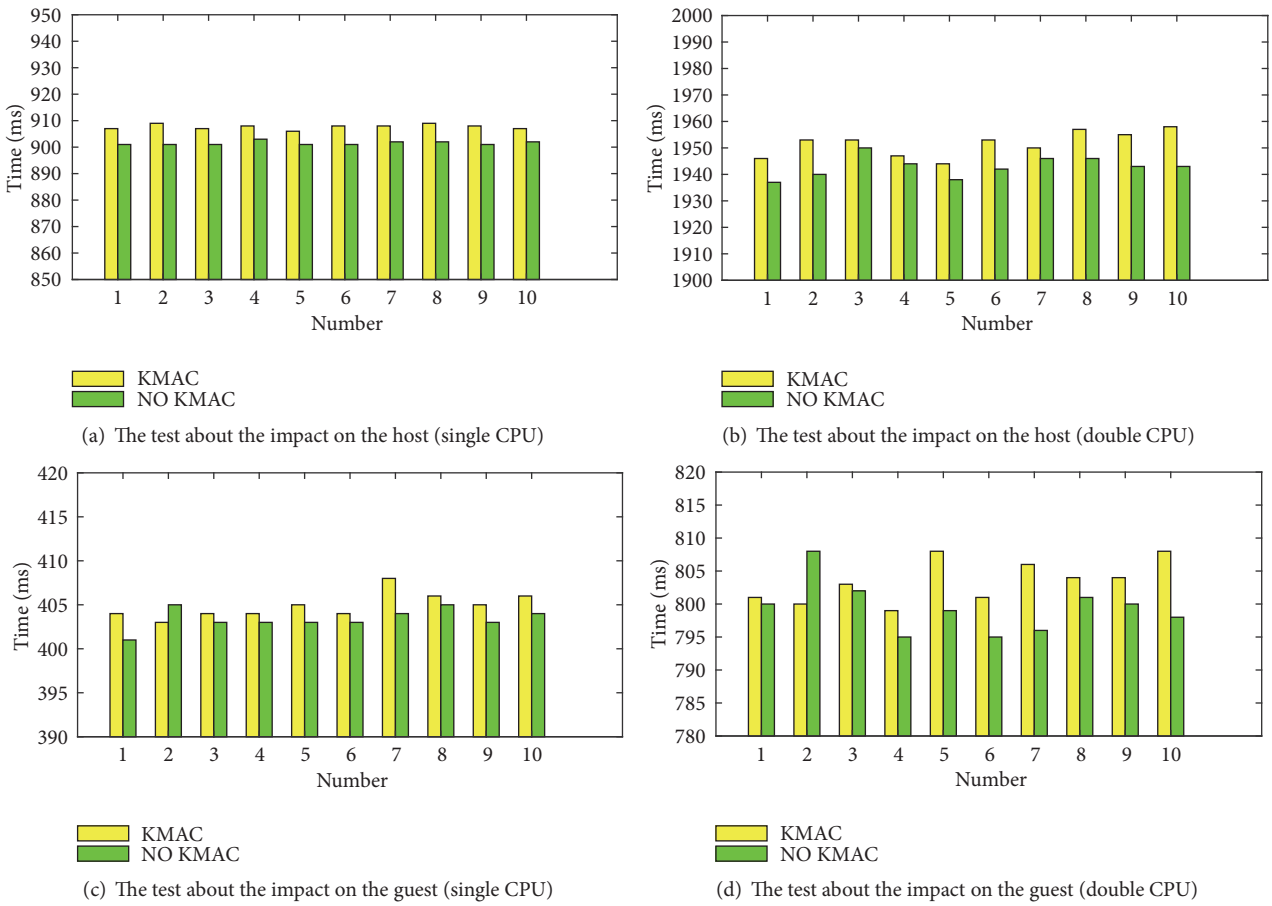


Figure 7: Performance comparison of VR-BLP.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2016YFB0800804), National Natural Science Foundation of China (61672411 and U1401251), Research Foundations for Science and Technology on Communication Networks Laboratory (no. KX172600023), and China 111 Project (no. B16037).

References

[1] D. Kapil, P. Tyagi, S. Kumar, and V. P. Tamta, "Cloud Computing: Overview and Research Issues," in *Proceedings of the 2017 International Conference on Green Informatics (ICGI)*, pp. 71–76, Fuzhou, China, August 2017.

[2] 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.

[3] E. Aguiar, Y. Zhang, and M. Blanton, "An overview of issues and recent developments in cloud computing and storage security," *High Performance Cloud Auditing and Applications*, pp. 3–33, 2014.

[4] R. Uhlig, G. Neiger, D. Rodgers et al., "Intel virtualization technology," *The Computer Journal*, vol. 38, no. 5, pp. 48–56, 2005.

[5] P. Barham, B. Dragovic, K. Fraser et al., "Xen and the art of virtualization," in *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, pp. 164–177, Bolton Landing, New York, NY, USA, 2003.

[6] N. Jain and S. Choudhary, "Overview of virtualization in cloud computing," in *Proceedings of the 2016 Symposium on Colossal Data Analysis and Networking, CDAN 2016*, March 2016.

[7] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[8] H.-C. Chen, M. A. Violetta, and C.-Y. Yang, "Contract RBAC in cloud computing," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 1111–1131, 2013.

[9] R. K. Thomas and R. S. Sandhu, "Task-based authorization controls (TBAC): A Family of models for active and enterprise-oriented authorization management," in *Proceedings of the IFIP WG11.3 Workshop on Database Security*, pp. 166–181, 1998.

[10] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[11] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[12] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, "An analysis of security issues for cloud computing," *Journal of Internet Services & Applications*, vol. 4, no. 1, pp. 1–13, 2013.

[13] D.-E. Bell and L.-J. LaPadula, *Secure Computer Systems: A Mathematical Model. Volume II*, MITRE Corporation, Bedford, MA, USA, 1973.

[14] D.-E. Bell and L.-J. LaPadula, *Secure computer systems: Mathematical foundations*, MITRE Corporation, Bedford, MA, USA, 1973.

[15] D. McCullough, "Specifications for multi-level security and a hook-up," in *Proceedings of the IEEE Symposium on Security and Privacy*, p. 161, 1987.

[16] N. Zhang, M. Li, W. Lou, and Y. T. Hou, "MUSHI: toward multiple level security cloud with strong hardware level isolation," in *Proceedings of the 2012 IEEE Military Communications Conference, MILCOM 2012*, pp. 1–6, Orlando, FL, USA, November 2012.

[17] L. Freitas and P. Watson, "Formalizing workflows partitioning over federated clouds: multi-level security and costs," *International Journal of Computer Mathematics*, vol. 91, no. 5, pp. 881–906, 2014.

[18] X. Haiwei, Z. Yunliang, G. Zhien, and D. Yiqi, "A multilevel security model for private cloud," *Chinese Journal of Electronics*, vol. 23, no. 2, pp. 232–235, 2014.

[19] Z. Tan, D. Liu, X. Zhuo, Y. Dai, and L. T. Yang, "Implementation and performance analysis of multilevel security system in pervasive computing environment," *The Journal of Supercomputing*, vol. 66, no. 3, pp. 1243–1259, 2013.

[20] M. R. Heckman, R. R. Schell, and E. E. Reed, "A multi-level secure file sharing server and its application to a multi-level secure cloud," in *Proceedings of the 34th Annual IEEE Military Communications Conference, MILCOM 2015*, pp. 1224–1229, Tampa, FL, USA, October 2015.

[21] C. Wright, C. Cowan, J. Morris, S. Smalley, and G. Kroah-Hartman, "Linux security modules: General security support for the Linux Kernel," in *Proceedings of the Foundations of Intrusion Tolerant Systems, OASIS 2003*, pp. 213–226.

[22] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: a comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[23] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: a survey," *IEEE Communications Magazine*, vol. 51, no. 11, pp. 24–31, 2013.

[24] A. Gember, R. Grandl, J. Khalid, and A. Akella, "Design and implementation of a framework for software-defined middlebox networking," in *Proceedings of the ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 467-468, China, August 2013.

[25] "UnixBench," https://github.com/kdlucas/byte-unixbench.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7849065, 15 pages
https://doi.org/10.1155/2018/7849065

Research Article

Resetting Your Password Is Vulnerable: A Security Study of Common SMS-Based Authentication in IoT Device

Dong Wang[1], **Xiaosong Zhang**[1], **Jiang Ming**[2], **Ting Chen**[1], **Chao Wang**[3] and **Weina Niu**[1,4]

[1] *University of Electronic Science and Technology of China, China*
[2] *The University of Texas at Arlington, USA*
[3] *ADLab of Venustech, China*
[4] *College of Cybersecurity, Sichuan University, China*

Correspondence should be addressed to Xiaosong Zhang; johnsonzxs@uestc.edu.cn

Received 8 March 2018; Revised 28 May 2018; Accepted 4 June 2018; Published 4 July 2018

Academic Editor: Ximeng Liu

Firmware vulnerability is an important target for IoT attacks, but it is challenging, because firmware may be publicly unavailable or encrypted with an unknown key. We present in this paper an attack on Short Message Service (SMS for short) authentication code which aims at gaining the control of IoT devices without firmware analysis. The key idea is based on the observation that IoT device usually has an official application (app for short) used to control itself. Customer needs to register an account before using this app, phone numbers are usually suggested to be the account name, and most of these apps have a common feature, called *Reset Your Password*, that uses an SMS authentication code sent to customer phone to authenticate the customer when he forgot his password. We found that an attacker can perform brute-force attack on this SMS authentication code automatically by overcoming several challenges, then he can steal the account to gain the control of IoT devices. In our research, we have implemented a prototype tool, called *SACIntruder*, to enable performing such brute-force attack test on IoT devices automatically. We evaluated it and successfully found 12 zero-day vulnerabilities including smart lock, sharing car, smart watch, smart router, etc. We also discussed how to prevent this attack.

1. Introduction

The Internet of Things (IoT for short) paradigm is one of the most thrilling innovations of the recent years. Growing interest has spurred the commoditization of many devices for personal use, such as smart home devices, smart wearable devices, and smart car [1, 2]. The figure of online capable devices increased 31% from 2016 to 8.4 billion in 2017. Experts estimate that the IoT will consist of about 30 billion objects by 2020. It is also estimated that the global market value of IoT will reach $7.1 trillion by 2020 [3].

For an IoT device, it usually consists of three components, an electronically augmented hardware device that reports its status and processes user commands, a mobile device that is used to receive status and send commands, a cloud that is used to exchange messages between the hardware device and mobile device. When a customer deploys his device, he will (1) install the device, (2) download the official app and install it on his smart phone, (3) register an account via the app, (4) pair the device with his smart phone via the app, (5) start to control his device via the app. Usually, the device can be controlled remotely via the app. So, if there are vulnerabilities in the device, an attacker can also control it remotely. While prior works on IoT focused on cryptographic protocols analysis [4–6], limited work has studied on security vulnerabilities of implementation.

In this paper, we study the common SMS-based authentication that is used in IoT devices. We observed that many IoT devices support a common feature, called *Reset Your Password*, designed for a customer to change his account password if he forgets it. To authenticate the customer, IoT cloud will send an authentication code to customer phone that is registered as the account name via SMS. These SMS messages share the same structure, *[sender] [text with authentication code] [expiration]* (e.g., *Panasonic: Your verification*

code is 3895, validity period is 5 minutes). SMS authentication code is usually a 4-digital or 6-digital number, here is *3895*. Ideally, the code is just known by the IoT cloud and the device owner, so anyone who can present the code will be considered as the real device owner. After analyzing several IoT devices, we found this SMS-based authentication of *Reset Your Password* may be vulnerable. An attacker can perform brute-force attack via mutating the SMS code in password reset message and reset IoT account password, because the search space of the digital SMS code is very small.

However, there is a big challenge: *cryptographically consistent message* (Section 3). If we modify the SMS code in a password reset message, it will become *cryptographically inconsistent* because the message may contain a signature, and this will cause the message being discarded by IoT cloud. We found that we can treat IoT app as a black box and control its execution, then we can reuse the app code to generate cryptographically consistent messages when we mutate the SMS code. We also addressed several other challenges that prevent us performing the attack test automatically. We proposed the design of our prototype tool (Section 4) *SACIntruder*; it can be used to check whether a IoT device is vulnerable to the SMS-based authentication automatically. We implemented our tool and used it to find 12 zero-day vulnerabilities. For instance, we found a vulnerable smart lock and an attacker can enter into a victim's house without authorization. We also found a vulnerable car and an attacker can drive it away.

In short, we make the following major contributions:

(i) To the best of our knowledge, it is the *first* security study about the SMS-based authentication in IoT device, and we found it may be vulnerable. An attacker can perform brute-force test on SMS code to gain the control of IoT devices without any interaction of victims.

(ii) We designed a tool *SACIntruder* that can automatically perform brute-force attack test to check whether an IoT device is vulnerable to SMS code. Our tool addressed the big challenge about cryptographically consistent message generation and other several challenges such as UI identifying, parameter identifying, and time expiration.

(iii) We implemented our tool and evaluated it on IoT devices including smart lock, sharing car, smart watch, and smart router, and it found 12 zero-day vulnerabilities automatically. We already reported all of them to the *CNCERT/CC* [7] to help vendors to fix them, and they all have been fixed now.

The remainder of this article is structured as follows. In Section 2, we introduce the background knowledge of IoT security. Then, we use a home app as an example case study to understand our problem and present the overview of *SACIntruder* in Section 3. We present the detailed design in Section 4 and evaluate it in Section 5. We discuss how to prevent this attack with the goal of security and usability in Section 6. The survey of related work in Section 7 is followed by our conclusion in Section 8.

2. Background

Traditional embedded devices are offline, and they can be controlled just physically. In contrast, many IoT devices are online and can be accessed via the Internet. So, an attacker can gain the control of these devices remotely, if there are security flaws. The loose protection and pervasiveness of vulnerabilities [8, 9] make these devices very weak to attackers. For instance, there are more than 90 reports about independent IoT attack incidents from 2014 to 2016 [10].

For IoT attacks, firmware is always an important target, because security vulnerability in firmware usually can bypass all limitations such as the accessibility of the underlying system, and an attacker can find a large number of vulnerabilities by analyzing firmware because it contains all critical codes. In 2017, F-Secure security researcher analyzed the firmware of a Foscam IP camera and found 18 zero-day vulnerabilities including insecure default credentials, command injection, stack-based buffer overflow, etc. There are some works about detecting vulnerabilities in firmware, some utilize symbolic execution [11, 12] to detect flaws automatically, while others construct an emulation runtime for dynamic analysis [13–15]. However, firmware acquisition is a big challenge for detecting vulnerabilities via firmware analysis, because not every device firmware is publicly available. Even if available, it may be encrypted with an unknown cryptographic algorithm or data key. In addition, firmware is usually a compressed archived file, it is unable to decompress it without the knowledge of the archived file format. The diverse architectures of hardware chipset is another challenge for firmware analysis, because different chipset has different memory layout and instruction set.

Many IoT devices can be controlled by a customer through the official apps with his IoT account (e.g., a smart lock that enables its owner to open or close the door remotely). So, if an attacker can compromise the account, he can gain the control of the device via the app. There are some methods for cracking an account. Password Brute-force attack [16] is a traditional account attack by trying all possible passwords. After analyzing a lot of IoT apps, we observed that most IoT accounts have the same password strategy: at least 6 characters and each character can be *A-Z*, *a-z*, and *0-9*. So, the size of password search space will be $(26 + 26 + 10)^6 = 56$ billion, it will take quite a long time to test every possible password. Cross-Site Scripting [17] (XSS for short) and Cross-Site Request Forgery (XSRF for short) [18] can also be used to take over an account, but they are special for browser and most IoT apps are not built on a browser. So, XSS and XSRF rarely impact IoT account. Phishing [19] is another important method to steal an account, an attacker runs an evil website that is very similar to the target website and guides a victim to access it and input his credential. However, IoT customers are typically guided to install apps from the vendor cloud or the official app store by the device manual. Therefore, it is challenging for a phishing attacker to inject the download of IoT apps to get the victim credential. Man in the Middle [20] (MitM for short) attack is another common way to steal a user account. The recent work, Password Reset MitM [21] (PRMitM for short) attack, exploits the similarity

```
GET /ci/user/getVerifyCode?uid=-1&phone=1383815****
&imei= HTTP/1.1
Host: * * *.* * *house.com.cn
Connection: Keep-Alive
Accept-Encoding: gzip
User-Agent: okhttp/3.3.1
```

Box 1: Messages of the password reset for a home app. App Request for a SMS authentication code.

```
{"code":200,"msg":"","result":{"session_id":"f7b532
83-3c20-400d-b0ee-76a171036414","code_-1":-
1,"error_code":"0"}}
```

Box 2: Messages of the password reset for a home app. Cloud Response for the SMS authentication code.

```
POST /ci/user/fgt/pwd?password=e10adc3949ba59abbe56
e057f20F883e&code=7496&phone=1383815****&sign=d3db1
a89d68cd72cbd2 a3fcbf9822876 HTTP/1.1
Cookie: JSESSIONID=f7b53283-3c20-400d-b0ee-76a17103
6414
Content-Length: 0
Host: * * *.* * *house.com.cn
Connection: Keep-Alive
Accept-Encoding: gzip
User-Agent: okhttp/3.3.1
```

Box 3: Messages of the password reset for a home app. App Request for password reset.

```
{"code":0,"msg":"reset success"}
```

Box 4: Messages of the password reset for a home app. Cloud Response for password reset.

of the registration and password reset processes to launch a MitM attack to popular websites and mobile apps. However, PRMitM relies on victim's interaction heavily to retrieve everything that is essential for password reset. So, the success of PRMitM attack relies on several strong assumptions. First, it requires that the victim registers an account or inputs his mobile phone and SMS authentication code to the compromised website. Second, it assumes that many victims will ignore the details of the password reset message but just copy the SMS code into the compromised website. Third, an attacker has to analyze target website to get the knowledge of every challenge. The attack will not happen if any of the above assumptions does not meet.

As stated early, the search space of the SMS code used in *Reset Your Password* is very small and we can perform brute-force attack on it. Once our mutation meets the right value before the time expiration, we can reset the password of IoT device account. In contrast to our previous work [22], we have presented a new UI model, a new approach to identify parameters and an approach to reduce unnecessary requests. So, we can find new vulnerabilities that cannot be found via our previous work.

3. Overview

The goal of this work is to understand the SMS authentication of *Reset Your Password* in IoT devices and automatically identify whether a device is vulnerable. In this paper, we focus on IoT apps in Android platform that is the most popular mobile platform in the world [23]. We first use a running example to discuss our problem in Section 3.1; we discuss the major challenge in Section 3.2 and other implementation challenges in Section 3.3 and then give an overview of *SACIntruder* in Section 3.4.

3.1. A Running Example of Password Reset. To understand our problem better, we select an android IoT app that is designed to control smart home devices. Boxes 1–4 illustrate

the messages used for the whole progress of the password reset.

When a customer wants to reset his IoT account password, he will be guided by app UI to send a message to the cloud for an SMS code, then the cloud will generate a code, send it to customer phone, and respond with a message to the app. Boxes 1 and 2 present a pair of messages about request and response. In theory, there is no limitation for the message format between the app and cloud. After analyzing a lot of apps, we found that most of them are built on Hypertext Transfer Protocol (HTTP for short) [24]. One possible reason may be that Representational state transfer (REST for short) [25] is popular in the development domain, and REST uses HTTP as its low-layer transport protocol. The Uniform Resource Locator (URL for short) [26] of REST usually has full meaning. In the running example, *getVerifyCode* and *phone=1383815∗∗∗∗* mean requesting the cloud to send an authentication code to *1383815∗∗∗∗*.

After receiving the SMS authentication code, customer inputs the code and new password and clicks a UI component to send a message containing the code and password to the cloud, then the cloud verifies them to replace the old password, at last responds with a message to the app. Box 3 presents a password reset request message, it contains four key parameters, three of them (*password, code, phone*) are mapped to user inputs, while *sign* cannot be mapped to any input. Moreover, *sign* is usually generated by a cryptographic algorithm with arguments including *password, code, phone* and other app-specific data. It is easy to modify the *code*, but *sign* will prevent us from doing this. Because if we modify it, we must update *sign*, or the message will be cryptographically inconsistent and the cloud will discard it. Besides the cryptographically consistent message generation, several implementation challenges also need to be addressed, we discuss them in the following sections. Box 4 presents a password reset response message for successful password reset.

In addition, some IoT apps only support to login the account with SMS authentication code, the customer never owns a password. He will request an SMS code every time when he wants to login the account. This type of login can be classified as a special password reset that contains no new password. So, we can support these apps using the same brute-force attack method.

3.2. Cryptographically Consistent Message Generation. The SMS code in *Reset Your Password* is a digital number with small search space, it is easy for the brute-force attack. But many IoT apps use at least one cryptographic strategy to protect messages, some chose signature to ensure the integerity [27], while others chose encryption to ensure the confidentiality [28]. IoT cloud will check the confidential or integrity of a message. If not valid, discard the message. We must find an approach to generate cryptographically consistent messages while mutating the code.

A straightforward method is to extract the cryptographic algorithm from the app and then reimplement it in the outside of IoT apps. Some program analysis technologies such as symbolic execution [29–31] and taint analysis [32, 33] can be applied to do this. These technologies can analyze a program to determine what inputs cause each part of the program by monitoring the execution of every instruction and its referenced data, so they are widely studied to explore the internal status of a program. They work well for many program logic, except the cryptographic algorithm because of the well-known challenge named *path explosion* [34, 35]: the number of feasible logic paths in a program grows exponentially with an increase in program size and can even be infinite in the case of programs with unbounded loop iterations. Unfortunately, loop iterations are very common in the cryptographic algorithm. IoT apps are commercial software, they usually contain complex logic and developers prefer to deploy some protections to enhance the security. In addition, packer is widely used to protect an app by developers, it contains code obfuscation [36], resource encryption, antidebugging, antiemulation, etc. So, it is very expensive for program analysis to extract the cryptographic algorithm from IoT apps because of these protections.

Based on the fact that nowadays most apps use a standard cryptographic algorithm for encryption and signature, Zuo [37] utilizes API hooking to extract the cryptographic algorithm. He hooks 61 well-defined cryptographic APIs in Android SDK to intercept their arguments and return values, then reconstructs the control flow and data flow based on the API hooking log, and reexecutes them out of the app. After analyzing a lot of IoT apps, we find that this method lacks flexibility, because it just supports well-defined cryptographic APIs. In our running example, the signature is generated as *sign* ≔ *MessageDigest.getInstance("MD5").dgest(Base64.encodeToString("...password=e10adc3949ba59abbe56e057f20f883ecode=7496phone=1383815∗∗∗∗..."))*. Obviously, *getInstance* and *dgest* will be logged as they are well-known and being hooked, but the arguments will not be logged as *encodeToString* is not well-known and not being hooked. So, the data flow will be interrupted and the cryptographic algorithm for generating *sign* cannot be reconstructed from the API hooking log. In addition, private cryptographic algorithm is also not supported by this method.

In fact, our final goal is the output, not the code of the cryptographic algorithm. If we treat the whole IoT app as a cryptographic algorithm, user input as the arguments, the output as the password reset message, then we can utilize UI automation [38] to input every possible SMS code and request password reset, app will execute its code to calculate the *sign*. So, we can generate a cryptographically consistent message without extracting the cryptographic algorithm. Moreover, this approach is independent on the cryptographic algorithm, so it does not have the drawbacks of program analysis and API hooking, it can support complex app logic, code obfuscation, private cryptographic algorithm and so on. Based on UI automation, we can generate cryptographically consistent messages when mutating the SMS code, but we still need to address several other implementation challenges for performing brute-force attack test automatically.

3.3. Implementation Challenges and Solutions. There are three implementation challenges for utilizing UI automation to perform brute-force attack on SMS code automatically: (1)

identifying password reset UI, (2) identifying interesting parameter, (3) time expiration on the SMS authentication code, and (4) unnecessary brute-force requests. We must address all of them.

Identifying Password Reset UI. Before we can use UI automation to input phone, SMS code, new password to drive the app to generate the password reset message, we need to identify the password reset UI firstly. Unfortunately, there is no straightforward information to declare where is the password reset UI. We analyzed a lot of apps and found that most of password reset UI contain the common feature: (1) an editable UI component with a default description like *input your phone number* for guiding users to input phone, (2) an editable UI component with a default description like *input SMS code* for guiding users to input SMS code, (3) an editable UI component with a default description like *input new password* for guiding users to input password, and (4) a clickable button with a default description like *confirm* for guiding users to submit request. This is because developers usually need human-friendly text information to guide users to input password reset parameters in the right component. So, we can enumerate every UI of an IoT app and check which one contains this common feature to identify the password reset UI. In addition, some apps divide the logic of password reset into several parts, so they will use more than one UI to guide users to input all parameters. We can analyze a set of sequential UI to identify the root of password reset UI. Details about identifying password reset UI are presented in Section 4.1.

Identifying Interesting Parameters. Usually, there are more than four parameters in password reset message. But interesting instances are *phone, code, password, sign*, we need to identify them. Messages based on REST usually encodes a parameter as a key-value in the URL or as JSON/XML in the content [39, 40]. If the parameter key name is well-known, such as *verifyCode* and *password* used in our running example, it is easy to identify them by regular expression matching. But different apps can use different key names, *checkCode, vCode, ck* are also acceptable for the SMS code, so it is challenging to identify them automatically. However, the input of IoT apps is controlled by us via UI automation, so we can use two different values for a parameter and analyze the two corresponding messages to identify the parameter (e.g., in our running example, we input *code1* and *code2* for the *code*, we can find both values in the messages and infer the key name for SMS code is *code*). Details about identifying interesting parameters are presented in Section 4.2.

Time Expiration. IoT clouds always assign an expiration limit (varies from 2 ~ 30 minutes) on the SMS code. The brute-force test must be as fast as it can, or the authentication code will become unusable before being mutated to the right one. If the password reset message is not encrypted or contains no signature, we can mutate code based upon a captured message directly and send mutated messages to the cloud at a fast speed. Or we move to use UI automation to generate cryptographically consistent messages. In fact, the speed of UI automation is very slow, if we serially mutate the code and request the cloud in a real-time environment, we will meet the expiration. However, we can use an offline environment to record all request messages and then replay all of them to the cloud via a high-performance computer. Because these request messages contain all possible values for the SMS code, so the password will be reset successfully. Details about time expiration are presented in Section 4.3.

Unnecessary Requests. During the attempts of password reset with mutated codes, if the account password is reset successfully, we can discard the rest of messages to reduce unnecessary requests. But the response contents are diverse, there are no standards and documents for whether the password reset response is successful or not. We observed that the response usually contains a status message such as *your code is invalid* or *too many unsuccessful attempts*. Different messages have a different length, this will make the length of the whole response message different. So, we can monitor the response length. If it is changed, it means a new status, such as *success of password reset* or *being blocked by the cloud*. Then we try to login the account with the predefined password, if success and we discard the rest messages, if failure and we infer this IoT device account is not vulnerable.

By addressing above challenges, we can use UI automation to generate cryptographically consistent message while mutating the code to all possible values. In addition, if there is no signature or encryption in password reset message, we can directly mutate the code based on a captured password reset message to perform brute-force attack test.

3.4. SACIntruder. To check whether a given IoT device is vulnerable to the SMS-base authentication, we design a tool named *SACIntruder*, to automatically perform brute-force attack on the SMS code. The only knowledge we need is the phone number that is used to register as the device account. PMitM [21] presented a method about how to get victim's phone, and how to get the phone is beyond the scope of this paper. Moreover, IoT app may choose a binary protocol to build a request message. We only focus on HTTP protocol here, because we observed that a large number of Android apps use REST API that is built on the HTTP. We consider HTTPS as HTTP, because HTTPS uses SSL as its lower transport protocol and we can use a self-signed certificate to bypass HTTPS.

The whole progress of our brute-force attack supported by *SACIntruder* is presented in Figure 1. There are four steps: (1) *SACIntruder* starts every activity of an IoT app to identify password reset UI. (2) *SACIntruder* drives the app to input victim phone and request for SMS code, then it replaces the victim phone with a test phone via message interception, this replacement will prevent the cloud send a code to the victim at the very beginning. (3) *SACIntruder* drives the app to input parameters for password reset and request the cloud, then it identifies interesting parameter. If the message contains encryption or signature, it will stop forwarding the following message to the cloud, and continue driving the app to input all possible SMS code and save all messages into a database.

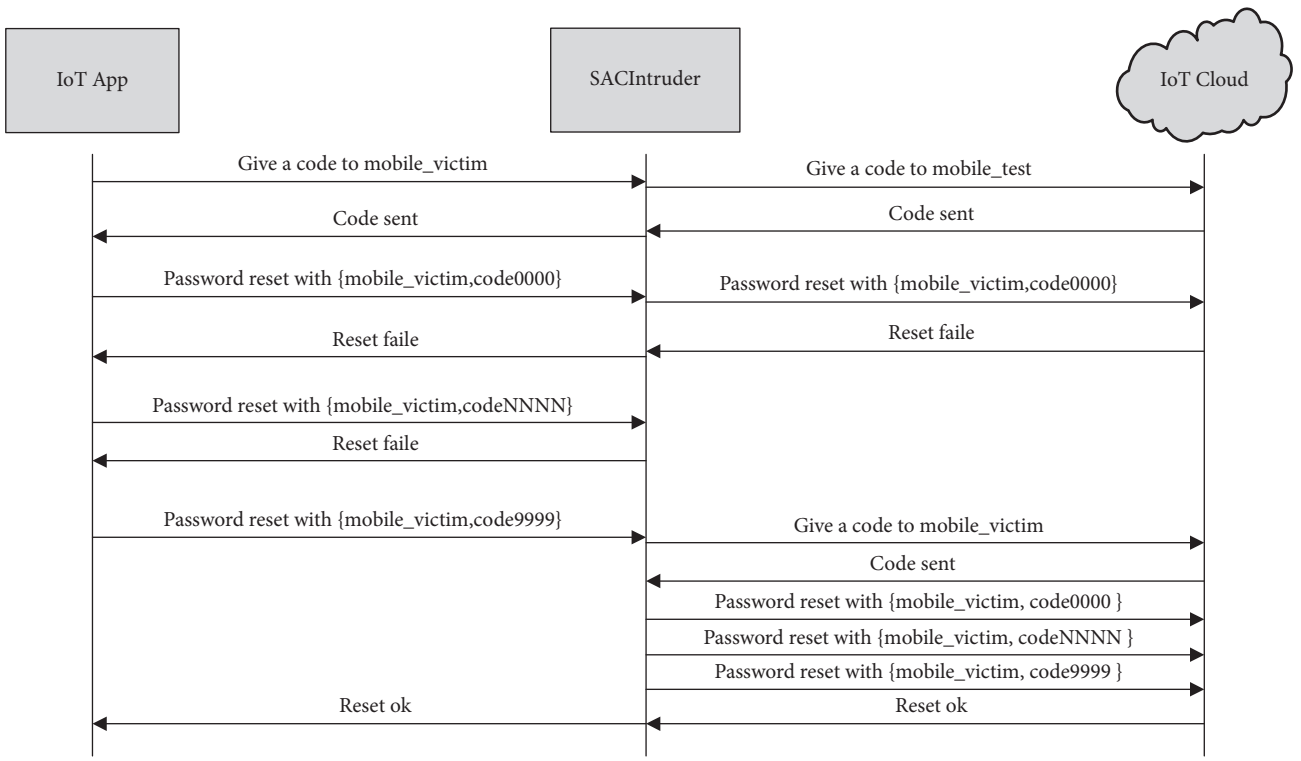


Figure 1: Password reset via brute-force attack to SMS authentication code: code0000 is a SMS authentication code whose value is 0000 and codeNNNN is an instance of all possible SMS authentication code whose value ranges from 0000 to 9999.

(4) At last, *SACIntruder* begins to replay the message for SMS code, and all messages for password reset to the cloud. If the account is vulnerable, its password will be reset successfully.

An overview of *SACIntruder* is presented in Figure 2: its inputs are an IoT app and a victim phone and the output is whether the IoT app is vulnerable. There are three key components:

(i) **MessageGenerator** using UI automation to control the execution of IoT app, it performs static analysis and dynamic analysis to identify UI and drives app to generate cryptographically consistent messages.

(ii) **ParameterIdentifier** using different inputs to identify parameters in the password reset message, it contains a network proxy to intercept the communication between the app and cloud. It cannot input data to the app, so it communicates with MessageGenerator to achieve this goal.

(iii) **RecordReplayer** uses a database to record all password reset messages and replay them to the cloud parallelly at last; it is running on a high-performance computer for fast speed.

4. Detailed Design

In this section, we present the detailed design of the three key components of SACIntruder. We first describe how does *MessageGenerator* generate a password reset message in Section 4.1, then explain how does *ParameterIdentifier* identify interesting parameters in Section 4.2, and then present how does *RecordReplyer* bypass the time expiration on SMS code in Section 4.3.

4.1. Password Reset Request Message Generation. MessageGenerator is responsible for password message generation, it utilizes UI automation to control the execution of IoT apps. First, it runs the app and identifies the password reset UI by enumerating every activity and checking the common feature in Section 3.3. As stated previously, there are two password reset UI models. Figure 3 is the classic UI model, named *Single-stage Password Reset*, because all parameters are inputted in a single UI. And Figure 4 is another model, named *Multistage Password Reset*, because all parameters are divided into two or three groups and inputted in several UI. Then, *MessageGenerator* enters the right UI, inputs parameters, and triggers request to generate password reset messages.

4.1.1. Single-Stage Password Reset Request Message Generation. For the *Single-stage Password Reset*, all UI elements used to receive these parameters are defined in an activity. First, we identify the UI via analyzing every activity independently and then input every parameter to this activity to drive the app to generate password reset messages.

Static Method to Identify UI. In Android, every activity must be declared in a file named *AndroidManifext.xml* that can be

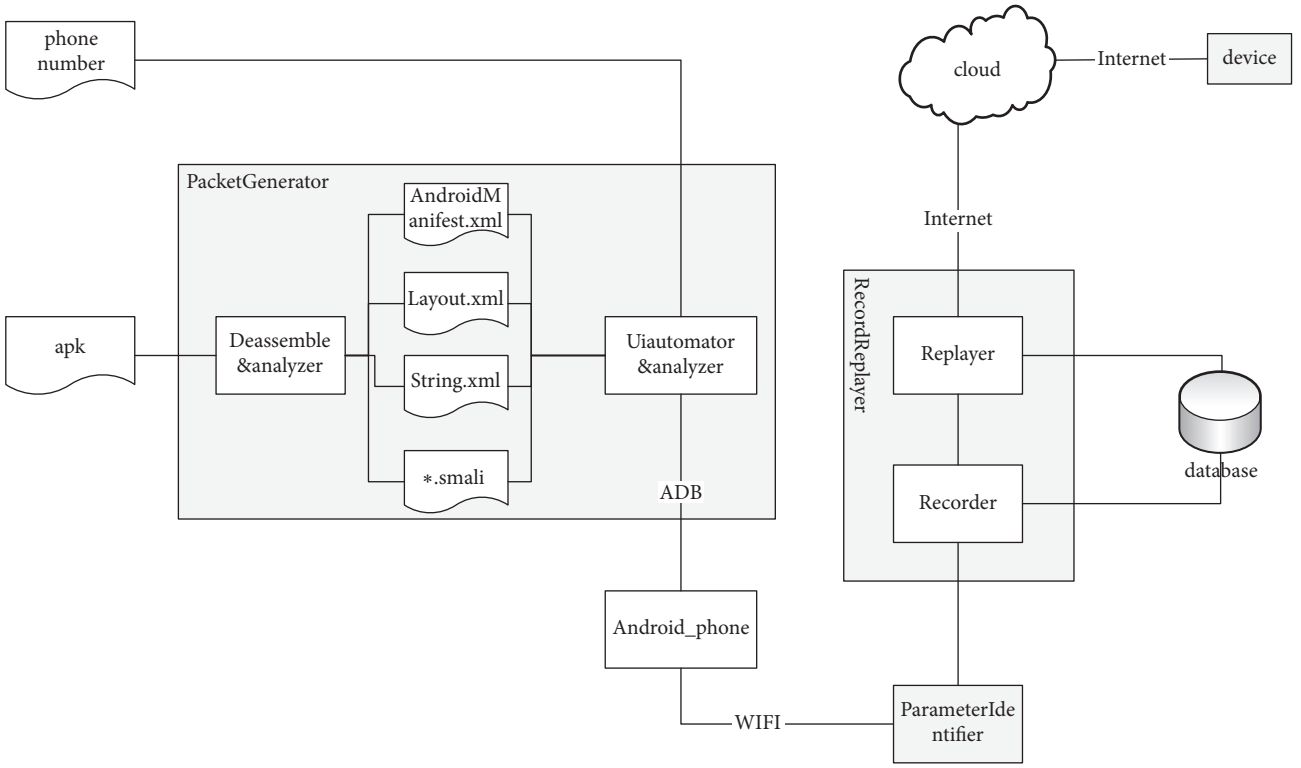


Figure 2: An overview of SACIntruder.

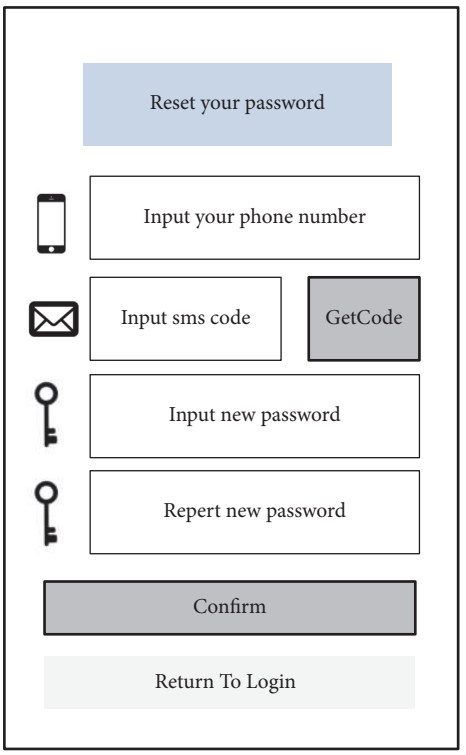


Figure 3: Single-stage password reset UI model.

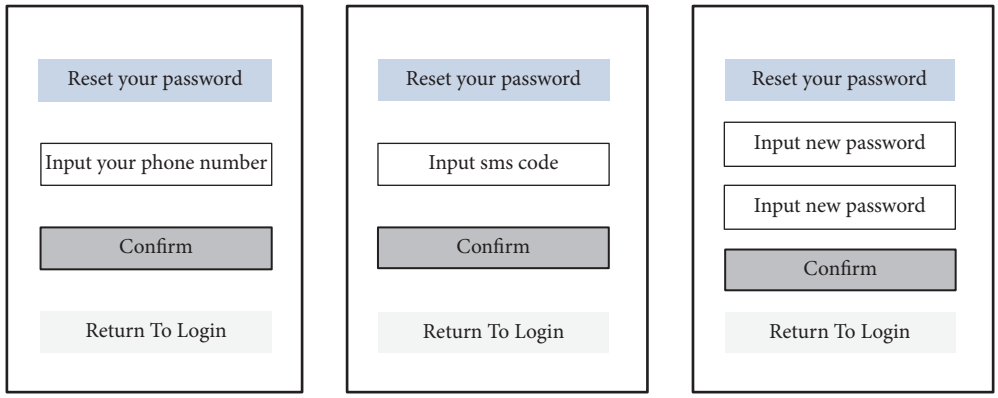


Figure 4: Multistage password reset UI model.

extracted from the app binary file. Some other works such as AppsPlayground, SMV-Hunter, and Gui Ripping [41–43] use this manifest to start dynamic UI exploration. But dynamic exploration is usually slow, so we first use static method. For an activity, all UI elements (e.g., *Buton, ImageButon, CheckBox*) are defined in a layout, and the activity uses *setContentView* to load the layout. In theory, a layout can be a standalone file or a piece of codes about dynamic layout generation. After analyzing a lot of apps, we found that most apps use a standalone file. So, we analyze the layout file of an activity to infer whether it is the right UI via checking the common human-friendly information.

Because an activity loads its layout by calling API *setContentView*, we need to analyze the code of an activity to get its layout. The source code of an IoT app is always not available, but it is not a problem. Because Android app is very similar to the Java archive, it is very easy to disassemble the app. We can use *apktool* [44] to disassemble the app. Every part of the app can be extracted to basic elements, such as string pool files, code files for every class, layout files, image files and so on. In these elements, a file named *public.xml* is very important, because the disassembled code of an activity will not use a name to reference its layout but an integer number that is defined in this file. Another file named *strings.xml* is also important, because the human-friendly information in password reset UI are usually defined here. There are in total five major steps in order to identify the right UI via static method:

(i) Disassemble an IoT app with *apktool*. If the app is not protected by a packer [45], we can get every part of the app. Otherwise, we can only get a subset of all parts, and we will move to dynamic method.

(ii) Get the name list of all activities from the manifest. Again, every activity must be declared in this file. If the app contains a password reset activity, we can analyze every activity to find it out. In addition, we get the package name of the app from the manifest, combine package name and activity name to generate the full name of the activity code file. The output of this step is an array of *activityFullname*.

(iii) Find the layout via disassembled code of every activity. The entry point of an activity is a callback function named *onCreate*. We scan the disassembled code of every activity's *onCreate* procedure to get the referenced layout. Again, the layout in the disassembled code has been converted to an integer. The output of this step is an array of *(activityFullname, layoutInteger)*.

(iv) Get the layout filename via *public.xml*; Android uses this file to map the integer to a name. The output of this step is an array of *(activityFulname, layoutName)*.

(v) Analyze the referenced layout file to infer whether it is the password reset activity. A layout file is an XML file that contains every UI element definition with type name and default value. Figure 3 is a typical *Single-stage Password Reset* UI that contains three input elements, two submit elements, and some human-friendly strings. If the layout file contains elements definition like this, we can infer it is the right UI. The output this step is *activityFullname*.

Dynamic Method to Identify UI. If an IoT app is protected by a packer, the real code and layout will be hidden from static analysis except for the manifest. But manifest just contains names of all activities, we cannot get the layout referenced by an activity from it. So, we move to dynamic method.

In Android, every running activity is managed by the activity manager. All activities are maintained via a stack, and they are arranged in the order according to the time when each activity is opened and only the top activity in the stack is painted on the screen [46]. To paint the screen, activity manager keeps a screen layout that contains the top activity and other UI elements such as system virtual home key. Moreover, Android allows dumping the layout of the current screen to support debugging. So, we can run every activity of an app and dump the current screen layout to get the layout of the running activity without regarding the packer or dynamic layout loading. There are in total four major steps in order to identify the right UI via dynamic method:

(i) Get all activity names from the manifest. This step is the same as the previous static method.

(ii) Start every activity of the IoT app. This will bring an activity to the top in the system activity stack. We use an Android phone and turn its debugging support on, then we use command *am start package/activity* to start an activity.

(iii) Dump the current screen layout. There is an android shell command named *uiautomator* [47] that is designed for dumping layout of the current screen. We use it to dump the layout and save it as a file in the phone, then we use Android Debug Bridge [48] (ADB for short) command to download the layout file from the phone.

(iv) Analyze the layout to infer whether it is the password reset UI. Unlike traditional standalone layout, screen layout already loaded the values of associated strings that are defined in another file. Another difference is it may contain some UI elements which do not belong to the app such as the status bar on the top of current screen or a popup window. To reduce the noise, we first exclude these elements, and then we check the common feature used in the static method. If found, we infer this activity is the password reset UI.

Contrast to the previous static method, dynamic method is much slower, because it runs the app and uses debug interface to get the key information. But it is very useful when the app is protected by a packer [45], or the app is implemented to load the activity layout dynamically via code.

Generate Password Reset Message. After identifying the password reset UI, we can use UI automation to input parameters for driving the app to generate a password reset message. The method is straightforward: enumerating parameter elements in the layout, inputting data and clicking the submit button to generate the message. Some apps contain one password elements, while two password elements for some other apps. It is easy to identify this difference, because the two elements for password are very similar based on their description text.

To capture the messages, we can use Wireshark, tcpdump and libpcap [49, 50]. However, we just focus on the application layer content that is consumed by the app and the cloud. We found that Android supports WIFI proxy that means all HTTP/HTTPS requests from app can be redirected to the WIFI proxy. Meanwhile, there are plenty of open-source HTTP proxy servers, we can utilize them to analyze the content of the HTTP layer. So, we use the WIFI proxy of Android and HTTP proxy server to capture the generated message.

4.1.2. Multistage Password Reset Request Message Generation. Many IoT apps use *Single-stage Password Reset*, but some apps use *Multistage Password Reset*. We directly use dynamic analysis to identify this password reset UI, because we cannot infer it by analyzing a single activity. Then we input parameters to drive the app to generate password reset message.

Identify the UI. A sample of *Multistage Password Reset* model is presented in Figure 4. There are three activities, the first one just receives the phone number, if it is confirmed, the app will switch to next one and an SMS code request message will be sent to the cloud; the second one receives the SMS authentication code, if confirmed, the app will switch to next one; the third one receives the password, if confirmed, the password reset message will be sent to the cloud.

It is challenging to recognize this type of password reset UI, because every single activity does not contain sufficient information. But it is obvious that if there are some sequential activities in the app, and every one of these activities contains a different part of the whole parameters for password reset, we can infer it is the right UI. So, we use a state machine:

(i) First, we define a 3-bit-vector, the first bit means the presence of a phone number UI element, the second bit means the presence of an SMS code, the third bit means the presence of a password. And we define four states: *s0* is the initial state, *s1* means that phone number has been met, *s2* means that phone and SMS code have been met, and *s3* is the final state that means all parameters have been met.

(ii) Second, we run every activity and dump the current screen layout. Then we check the presence of phone number, SMS code, and password, and change the state according to the current state.

(iii) Third, if we reach the *s3*, we infer the activity that changes the state to *s1* is the password reset UI.

Generate Password Reset Message. Here we use the same method used in the *Single-stage Password Reset* model to generate the messages and capture them. The only difference is we need to input parameters in several UI, because each UI just receives a part.

4.2. Identifying Interesting Parameter in Password Reset Message. Usually, there are more than four parameters in the password reset message. But we are just interested in *phone, code, password, sign*. We need a principled approach to parse the request message and identify our interesting parameters.

4.2.1. Parsing Parameters in Request Message. Again, we focus on IoT app built on REST [51] that uses HTTP as its transport protocol. According to the HTTP protocol specification, each request message consists of (1) a request line (e.g., *GET /ci/user/getVerify Code?uid=-1&phone=1383815∗∗∗∗&imei= HTTP/1.1*), (2) extra request-headers (e.g., *Host: ∗ ∗ ∗. ∗ ∗∗house.com.cn*), (3) an empty line, and (4) optional message body. There are two main methods for HTTP request, *GET* and *POST*. For a *GET* request, it can host a small piece of data in the URL. For a *POST* request, it can host a lot of data in the message body. And both methods are popular in IoT apps.

According to the URL specification [52], data can be encoded as a *k-v* block in the URL parameters. The first *k-v* is connected to the base URL via symbol "?", *k-v* is connected via the symbol "=", and each *k-v* pair is concatenated via "&". To parse these parameters, we first parse the path segment via scanning the reserved path symbol "?" to get the first *k-v* (*uid=-1* in our running example). Then we use "&" to parse more *k-v* (*phone=1383815∗∗∗∗* and *imei=*). It is quite

straightforward to index the parameter name and its value, and we store them in a pair *(name, value)*.

Parameters can also be in the message body when the method is *POST*. There is no limitation on the format of the body, because requester can specify its format type via a *Content-Type* request header. But the best practice of REST development suggests developers to use JSON and XML as the format for message body and we only parse these two popular formats. JSON and XML have a hierarchy tree structure, which means that each value can be tracked by the path from the root of the tree. And we still store them in a pair *(name, value)*.

4.2.2. Identifying Interesting Parameter with Different Input Value. After parsing parameters, we need to know whether a parameter is interesting, such as the signature. We do not care those uninteresting parameters (e.g., *app version* and *mobile version*), because they are irrelevant to the procedure logic of password reset and we just keep their original values. There are four interesting parameters: *phone, code, password, sign*. For the phone number, it is used in SMS code request message and password reset request message. SMS code is the most important parameter; we want to mutate it to all possible values. Signature is a hidden parameter that is automatically generated by the app logic. If signature or encryption is found, we utilize UI automation to drive the app to generate cryptographically consistent messages.

The values of these parameters are from the user input in the password reset UI, and we can control the input value via UI automation. So, we change the input value of a UI element with different values, then we analyze the generated messages to locate the two different input values. By this way, we infer the interesting parameters without the knowledge of their key name:

(i) For the *phone*, we input two different phone numbers, same code, and same password in the password reset UI. Then we trigger the UI to send the password reset message and parse these two messages. If a parameter with the same key contains both different input values, we infer the parameter is the phone number. For our running example in Boxes 1–4, we use two inputs *(phone1, password, code)* and *(phone2, password, code)* to drive the app to generate two messages. In the following analysis, we can find a parameter whose key name is *phone* contains our two different values *(phone1, phone2)*. So we get the phone parameter without any knowledge about its key name.

(ii) For the *code*, we input two different SMS codes, same phone number, and same password. Then we infer the SMS code like the phone number.

(iii) For the *sign*, we input two different SMS codes, same phone number, and same password. Unlike the SMS code, we do not have any input value for signature, so we do not know what should be searched in the two messages. Cryptographic algorithms have a desirable property named avalanche effect [53], it means that a little difference in the inputs will cause a dramatic difference in the outputs. Signature is built on a cryptographic algorithm, and we can use this common property to infer the *sign*. We parse these two messages, if we find a parameter that has a very high score of difference in its two values, we infer it is the *sign*. We use Euclidean distance [54] as the score and we also use this method to check whether the whole message is encrypted.

4.3. Time Expiration Bypass. We utilize UI automation to generate cryptographically consistent messages and support all types of algorithm including private versions. But it is very slow for UI automation, because every password reset request will indeed (1) get parameters from the UI elements and check validation, (2) perform encryption or signature, (3) create a new TCP connection to the cloud, (4) build an HTTP request message containing the data from step2, (5) send the message to the cloud, (6) wait for the response message and parse it, and (7) synch of UI events. In addition, for an Android phone, capability for computing and networking is very limited. So, if we directly use UI automation to drive the app to perform brute-force on the SMS authentication code, we will fail because of time expiration of the SMS code.

After analyzing a lot of IoT apps, we found that we can use an offline-style method, the core idea is that we just drive the app to generate all messages in an offline environment and replay all of them in an online environment:

(i) Intercept the SMS code request message from the IoT app and record it, replace the phone number by another one used for experimentation, forward the modified message to the cloud, receive the response message and record it, at last forward the response to the app. Then the app can send password reset message.

(ii) Intercept the password reset request message from the IoT app, forward it to the cloud if it is the first one, receive the response and record it, forward the response to the app. In the following requests, we will not forward the requests to the cloud and use the recorded response message to emulate the cloud response. In this way, we generate all password reset messages in an offline environment without communication to the cloud.

(iii) After all possible messages have been generated, we perform the brute-force test. First, we replay the recorded SMS code request message to the cloud to generate an SMS code for the real phone number. Second, we replay all password reset messages to the cloud. We use socket pool and thread pool to maximize the speed. And we monitor the response length, if it is changed, we stop replaying and use the predefined password to try account login. If successful, we infer the IoT device account is vulnerable.

5. Evaluation

We have implemented *SACIntruder* based on several open-source tools, our message generation is built on *uiautomator*

Table 1: Summary of IoT Apps under testing.

Type	Vendor	AndroidApp	UI Model
Watch	ToyCloud	com.watch∗ ∗ ∗.www	Single-stage Password Reset
Lock	Panasonic	com.∗ ∗ ∗.digitallock	Single-stage Password Reset
SharingCar	Panda	com.∗ ∗ ∗.usecar	Single-stage SMS Login
SharingCar	win-sky	com.∗ ∗ ∗.drivevi	Single-stage SMS Login
Router	ximo	com.∗ ∗ ∗.router	Single-stage Password Reset
HomeGate	HuiJu	com.∗ ∗ ∗.devices	Single-stage Password Reset
Robot	lejurobot	com.∗.zelos	Multi-stage password Reset
Car Cmera	DUDU	com.∗ ∗ ∗.∗ ∗ ∗.laucher	Single-stage Password Reset
HomeGate	BroadLink	com.∗ ∗ ∗.rmt	Multi-stage password Reset
Car	DasAuto	com.∗ ∗ ∗.faw.vw.∗ ∗ ∗	Multi-stage password Reset
IP Camera	uniview	com.∗ ∗ ∗.ezview	Multi-stage password Reset
Car	DongFeng	com.∗ ∗ ∗.windlink	Single-stage Password Reset

Table 2: Summary of discovered vulnerabilities.

AndroidApp	CNVD	Public
com.watch∗ ∗ ∗.www	CNVD-2017-02059	Yes
com.∗ ∗ ∗.digitallock	CNVD-2017-03908	Yes
com.∗ ∗ ∗.usecar	CNVD-2017-04583	Yes
com.∗ ∗ ∗.drivevi	CNVD-2017-06343	Yes
com.∗ ∗ ∗.router	CNVD-2017-15081	Yes
com.∗ ∗ ∗.devices	CNVD-2017-03909	Yes
com.∗.zelos	CNVD-2017-01003	Yes
com.∗ ∗ ∗.∗ ∗ ∗.laucher	CNVD-2017-09696	Yes
com.∗ ∗ ∗.rmt	CNVD-2017-12023	Yes
com.∗ ∗ ∗.faw.vw.∗ ∗ ∗	CNVD-2017-25143	Yes
com.∗ ∗ ∗.ezview	CNVD-2017-12075	Yes
com.∗ ∗ ∗.windlink	CNVD-2017-15147	Yes

[55] and *mitmproxy* [56], our RecordReplayer deploys *SQLite* [57] as its persistent data storage. And we wrote python code to drive UI automation and intercept messages, we wrote C code to replay recorded messages.

5.1. Experiment Setup

IoT Devices. We selected 12 representative IoT devices from different categories, including car, sharing car, robot, smart lock, smartwatch and smart router, etc. All these devices have an official Android app used to manage them. The detailed specifications of these IoT apps are described in Table 1. In particular, we summarize app information and their UI model. There are three types of UI model, *Single-stage Password Reset*, *Single-stage SMS Login*, *Multistage Password Reset*. *Single-stage SMS Login* can be considered as a special type of *Single-stage Password Reset*.

Testing Environment. Our IoT UI automation runs on an Ubuntu 12.04 PC with Intel Core i7 quad-core 3.6 GHz CPU with 16G RAM, a wireless router *TP-LINK TL-WAR1200L 1200M* and an Android phone *OnePlus*. Both the phone and PC are connected to the same wireless router, the WIFI proxy of the OnePlus is configured to the PC. We did not test IoT account of other customers, we register an account with our experimental phone to simulate the victim account. During our testing, the SMS authentication code will be sent to our phone, but we never use it and we just perform brute-force on the SMS authentication code.

5.2. Evaluation Result. We found the official apps of these devices are vulnerable to the brute-force attack on SMS code, an attacker can steal the accounts of these devices to control them remotely. As shown in Table 2, we found 12 zero-day vulnerabilities, the third column indicates whether the vulnerability can be indexed publicly in the *China National Vulnerability Database* [58](CNVD for short). Again, all vulnerabilities we founded have been reported to CNCERT/CC [7] to help the vendor fix them.

There are 8 vulnerabilities about *Single-stage Password Reset and Login*, and six of them are about password reset. There are other 4 vulnerabilities about *multistage password Reset*. Our tool supports both single-stage UI model and multistage UI model.

5.3. Case Studies. **CNVD-2017-03908** is a password reset vulnerability about a smart lock that belongs to Panasonic.

First, our tool finds that the app uses *Single-stage Password Reset* model. Second, it drives the app to generate a password message. Third, it finds that there are no encryption and signature in the message, so it mutates the SMS code in the message directly on a computer that is faster than a smartphone. Fourth, the tool sends mutated messages to the cloud to reset password successfully. An attacker can use this vulnerability to steal the victim account, then he can open victim's door to do anything. Panasonic has fixed this issue now.

CNVD-2017-04583 is a login vulnerability about a sharing car that belongs to a company that owns many cars and leases them to the customers. First, our tool drives the app to generate a password message. Second, it finds that the message has a signature, it cannot directly mutate the SMS code. Third, it uses UI automation to drive the app to try all possible code to generate password messages and record them to a database. At last, it replays all messages to the cloud to generate a login token and forwards the token to the application. As a result, an attacker can use the car sharing service in victim's name. It has been fixed now.

CNVD-2017-15147 is a password reset vulnerability about a car intelligent interconnected system that can send commands to the *Electronic Control Unit* (ECU for short) to open/close door, window and *Car-Carrying Air-Conditioning*. The app also uses *Single-stage Password Reset* model, and the SMS code parameter uses a strange key name *checknum*, not traditional $***code$. Our tool can find it because we use different input values to infer parameter. An attacker can utilize this vulnerability to drive a car away. It has been fixed now.

CNVD-2017-12023 is a password reset vulnerability about a smart home controller that can control a lot of devices provided by broadlink, the app uses *multistage password reset* model. Our tool first finds the right UI by analyzing several sequential activities and drives the app to generate a password reset messages. Then our tool finds that the message is encrypted via inputting different values, so it uses UI automation to generate all password messages and record them to a database. At last, it replays all messages to the cloud to reset password successfully. An attacker can use this vulnerability to control home devices. It has been fixed now.

6. Discussions

Possible Countermeasures to Prevent This Attack. This paper shows that *Reset Your Password* of IoT apps may be vulnerable, because an attacker can brute-force attack SMS authentication code to crack IoT device user account without any victim's interaction. The core insights are (1) the search space of SMS code used in the password reset is much smaller than password, (2) the cloud does not limit the number of attempt for account management. Frequency limitations, such as IP-based strategy and Account-based strategy, may pose some problems. If an attacker and legal customers are behind the same NAT [59] gateway, IP-based strategy will block legal customers to access to their devices. Account-based strategy will block the device owner to access the device, if an attacker performs brute-force on his account. The best protection is to deploy *CAPTCHA* [59] in password reset message, because our method relies on pregeneration of all password reset messages containing every possible SMS code. To balance security and usability, we can just activate *CAPTCHA* when the number of unsuccessful attempts meets a threshold.

Can SACIntruder Works on iOS. SACIntruder can also be implemented on iOS, because the core insights are (1) *Reset Your Password* is a feature at application level, it is independent on the low-layer smartphone operating system. (2) Human-friendly information, such as "input your phone" in Reset Your Password UI, is designed to guide user to input easily. We can use this to identify the UI, no matter the app is running on Android or iOS. (3) SMS authentication code is also independent on the low-layer smartphone operating system. It is usually a 4-digital or 6-digital number. Its search space is small. The major difference in these two platforms is UI automation, because it is dependent on the low-layer smartphone operating system. On Android, we can use uiautomator to control third apps. But we cannot do this on iOS because of its app sandbox, we need a jailbroken iphone to bypass this limitation. Then we can write our code to control the whole iOS, such as capturing the screen to identify UI via image processing and inputting data to third apps.

Limitations. Our paper just focuses on HTTP/HTTPS protocols because of the popularity of REST. But there is no limitation for IoT apps, they can use any protocol even private version based on binary format. Our tool SACIntruder uses the WIFI proxy of Android phone to help packet interception, so an IoT app can detect the WIFI proxy to prevent the packet interception.

7. Related Work

Vulnerability Discovery in Embedded/IoT Device. Costin [60] used static analysis to analyze more than 30000 firmware images to find bugs including XSS, hardcoded private key-pairs and back-door. Cui and Stolfo [61] found more than 500000 publicly accessible devices containing default credentials via Internet scanning. Cui and Costello [62] found that remote firmware update functionality can be exploited by attackers to insert malware. Davidson [11] used KLEE symbolic execution engine to detect memory vulnerabilities in open-source firmware. Li [63] ported the QEMU emulator to detect vulnerabilities in SoC. Zaddach [42] combined emulator and a real device to detect vulnerabilities. Chen [13] ported QEMU to run the Linux-based firmware to detect vulnerabilities on a large scale. Wang [64] designed a fuzz framework RPFuzzer by sending normal network packets and monitor CPU and system logs to detect vulnerabilities in routers. Costin [14] analyzed the management web interface in devices to detect vulnerabilities. Chen [65] designed a fuzzing framework named IoTFuzzer that uses the rich protocol information in IoT official app to guard fuzzing. In contrast to these works, our paper focuses on vulnerability on the IoT account. If the account is vulnerable, an attacker can use the account to control device via the account.

Account Security. SQL injection, XSS, CSRF, and logic fault are usually being used to hack an account. Halfond [66] presented an extensive review of the different types of SQL injection attacks. Vogt [67] tracked the flow of sensitive information inside the web browser to prevent XSS. Barth [68] performed an experimentation about CSRF vulnerability in 283945 advertisements and presented a new variation on CSRF attacks. Dalton [69] presented a novel methodology based on Dynamic Information Flow Tracking to mitigate authentication and access control vulnerabilities. Pellegrino [70] used a black-box methodology to detect logic flaws in web applications based on automatic identification of a number of behavioral patterns. Wang [71] performed a security analysis of cashier-as-a-service based web stores and found several logic flaws that can allow an attacker to buy an item at an arbitrarily low price. Gelernter [21] presented the password reset MitM attack that exploits the similarity of the registration and password reset processes, this attack can be used to take over user accounts. Zuo [37] performed password brute-forcing via automatic forgery of cryptographically consistent request message, it hooked standard cryptographical API to get knowledge of algorithms. Contrast to these works, our paper focuses on the SMS authentication code for account management, an attacker can perform brute-force attacking on the code to steal IoT account without any interaction of victim.

Mobile App Analysis. Monkey [72] is a testing tool for dynamic exploring the app UI automatically. Machiry [73] proposed a system named Dynodroid for generating relevant inputs to unmodified Android apps. Rastogi [41] proposed a framework that automates the analysis of Android application, it integrated multiple components comprising different detection and automatic exploration techniques. Anand, Mirzaei and Zuo applied symbolic execution [74–76] to perform more systematic dynamic analysis, so they can retrieve more internal knowledge but heavy overhead. Cui [77] proposed a tool named Discoverer for automatic protocol reverse engineering from network traces. Beddoe [78] maintained a protocol informatics project, it is very useful for protocol reverse engineering. Our paper is particularly by Monkey and protocol informatics project, we use UI testing tool to explore the UI component and input test data, we use different input values to identify interesting parameters.

8. Conclusion

We have performed the first security study of *Reset Your Password* that is popular in IoT device account and found it may be vulnerable because of the SMS-based authentication. We have presented the design, implementation, and evaluation of SACIntruder, a tool that is able to automatically perform brute-force attack on SMS code to test whether an IoT account is vulnerable. We have tested SACIntruder with representative IoT devices from different categories, including car, sharing car, robot, smart lock, smart watch, and smart router and found 12 zero-day vulnerabilities. We reported all vulnerabilities to CNCERT/CC to help the vendor to fix them, and all of them have been fixed now.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request. The vulnerabilities found in this paper can be accessed in the CNVD.

Disclosure

A conference version of this paper was presented at the RESEC 2018.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported in part by the Science and Technology Project of State Grid Corporation of China, National Natural Science Foundation of China (Grant no. 61572115), and National Key Research and Development Plan (2017*YFB*0802900), and Project 2117*H*14243*A* and Sichuan Province Research and Technology Supporting Plan, China.

References

[1] Google. Android wear, 2018.

[2] G. Ho, D. Leung, P. Mishra, A. Hosseini, D. Song, and D. Wagner, "Smart locks: Lessons for securing commodity internet of things devices," in *Proceedings of the 11th ACM Asia Conference on Computer and Communications Security, ASIA CCS 2016*, pp. 461–472, Xi'an, China, June 2016.

[3] C.-L. Hsu and J. C.-C. Lin, "An empirical examination of consumer adoption of Internet of Things services: Network externalities and concern for information privacy perspectives," *Computers in Human Behavior*, vol. 62, pp. 516–527, 2016.

[4] J.-Y. Lee, W.-C. Lin, and Y.-H. Huang, "A lightweight authentication protocol for Internet of Things," in *Proceedings of the 3rd International Symposium on Next-Generation Electronics, ISNE 2014*, twn, May 2014.

[5] C. Doukas, I. Maglogiannis, V. Koufi, F. Malamateniou, and G. Vassilacopoulos, "Enabling data protection through PKI encryption in IoT m-Health devices," in *Proceedings of the 12th IEEE International Conference on BioInformatics and BioEngineering, BIBE 2012*, pp. 25–29, November 2012.

[6] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, "Performance evaluation of Attribute-Based Encryption: toward data privacy in the IoT," in *Proceedings of the proceedings of the 2014 1st IEEE International Conference on Communications (ICC '14)*, pp. 725–730, Sydney, Australia, June 2014.

[7] CNCERT/CC, National computer network emergency response technical team/coordination center of china, 2018.

[8] Lucian Constantin, Hackers found 47 new vulnerabilities in 23 iot devices at def con. CSO, 2016.

[9] Chris Brook, *Travel Routers, Nas Devices among Easily Hacked iot Devices*, 2017.

[10] N. Zhang, S. Demetriou, M. Xianghang et al., "Understanding iot security through the data crystal ball: Where we are now and where we are going to be," https://arxiv.org/abs/1703.09809.

[11] D. Davidson, M. Benjamin, T. Ristenpart, and J. Somesh, "Fie on firmware: Finding vulnerabilities in embedded systems using symbolic execution," in *Proceedings of the In USENIX Security Symposium*, pp. 463–478, 2013.

[12] G. Hernandez, F. Fowze, D. Tian, T. Yavuz, and K. R. Butler, "FirmUSB," in *Proceedings of the the 2017 ACM SIGSAC Conference*, pp. 2245–2262, Dallas, Texas, USA, October 2017.

[13] D. D. Chen, M. Egele, M. Woo, and D. Brumley, "Towards Automated Dynamic Analysis for Linux-based Embedded Firmware," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA.

[14] A. Costin, A. Zarras, and A. Francillon, "Automated dynamic firmware analysis at scale: A case study on embedded web interfaces," in *Proceedings of the 11th ACM Asia Conference on Computer and Communications Security, ASIA CCS 2016*, pp. 437–448, chn, June 2016.

[15] devttyS0. Embedded device hacking, 2017.

[16] L. R. Knudsen and M. J. Robshaw, "Brute Force Attacks," in *The Block Cipher Companion*, Information Security and Cryptography, pp. 95–108, Springer, Berlin, Heidelberg, 2011.

[17] K. Spett, *Cross-Site Scripting*, vol. 1, SPI Labs, 2005.

[18] J. Burns, *Cross site request forgery. An introduction to a common web application weakness, Information Security Partners*, 2005.

[19] M. Wu, R. C. Miller, and S. L. Garfinkel, "Do security toolbars actually prevent phishing attacks?" in *Proceedings of the CHI 2006: Conference on Human Factors in Computing Systems*, pp. 601–610, can, April 2006.

[20] N. Asokan, V. Niemi, and K. Nyberg, "Man-in-the-middle in tunnelled authentication protocols," in *Security protocols*, vol. 3364 of *Lecture Notes in Comput. Sci.*, pp. 28–48, Springer, Berlin, 2005.

[21] N. Gelernter, S. Kalma, B. Magnezi, and H. Porcilan, "The Password Reset MitM Attack," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy, SP 2017*, pp. 251–267, May 2017.

[22] D. Wang, J. Ming, T. Chen, X. Zhang, and C. Wang, "Cracking IoT Device User Account via Brute-force Attack to SMS Authentication Code," in *Proceedings of the the First Workshop*, pp. 57–60, Incheon, Republic of Korea, June 2018.

[23] Google. Android, the world's most popular mobile platform, 2012.

[24] R. Fielding, J. Gettys, J. Mogul et al., "Hypertext Transfer Protocol – HTTP/1.1," RFC Editor RFC2616, 1999.

[25] R. Battle and E. Benson, "Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST)," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 1, pp. 61–69, 2008.

[26] A. Warshavsky, A. Fiske, B. Cinarkaya, and R. Guest, "System, method and computer program product for performing one or more actions utilizing a uniform resource locator," *The US Patent*, vol. 8, pp. 990-144, 2015.

[27] S. G. Stubblebine and V. D. Gligor, "On message integrity in cryptographic protocols," in *Proceedings of the Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 85–104, May 1992.

[28] W. Chung-Ping and C-C. Jay Kuo, "Fast encryption methods for audiovisual data confidentiality," in *Proceedings of the In Multimedia Systems and Applications III*, vol. 4209, pp. 284–296, 2001.

[29] M. Aizatulin, A. D. Gordon, and J. Jan, "Extracting and verifying cryptographic models from C protocol code by symbolic execution," in *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS'11*, pp. 331–340, October 2011.

[30] M. Boreale, "Symbolic trace analysis of cryptographic protocols," in *Automata, languages and programming*, vol. 2076 of *Lecture Notes in Comput. Sci.*, pp. 667–681, Springer, Berlin, 2001.

[31] R. Corin and F. A. Manzano, "Efficient symbolic execution for analysing cryptographic protocol implementations," in *Engineering Secure Software and Systems*, vol. 6542 of *Lecture Notes in Computer Science*, pp. 58–72, Springer, Berlin, Germany, 2011.

[32] F. Gröbert, C. Willems, and T. Holz, "Automated Identification of Cryptographic Primitives in Binary Programs," in *Recent Advances in Intrusion Detection*, vol. 6961 of *Lecture Notes in Computer Science*, pp. 41–60, Springer, Berlin, Heidelberg, 2011.

[33] D. Evans and D. Larochelle, "Improving security using extensible lightweight static analysis," *IEEE Software*, vol. 19, no. 1, pp. 42–51, 2002.

[34] C. R. Ramakrishnan and J. Rehof, "Attacking path explosion in constraint-based test generation," in *Proceedings of the nternational Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 366, 351 pages, Springer.

[35] C. Cadar and K. Sen, "Symbolic execution for software testing: Three decades later," *Communications of the ACM*, vol. 56, no. 2, pp. 82–90, 2013.

[36] 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)*, IEEE, pp. 297–300, November 2010.

[37] C. Zuo, W. Wang, R. Wang, and Z. Lin, "Automatic Forgery of Cryptographically Consistent Messages to Identify Security Vulnerabilities in Mobile Services," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, 2016.

[38] S. Hao, B. Liu, S. Nath, W. G. J. Halfond, and R. Govindan, "PUMA: Programmable UI-automation for large-scale dynamic analysis of mobile apps," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2014*, pp. 204–217, usa, June 2014.

[39] D. Crockford, "The application/json Media Type for JavaScript Object Notation (JSON)," RFC Editor RFC4627, 2006.

[40] T. Bray, J. Paoli, and Michael Sperberg-McQueen. C., "Extensible markup language (xml)," *World Wide Web Journal*, vol. 4, no. 2, pp. 27–66, 1997.

[41] V. Rastogi, Y. Chen, and W. Enck, "AppsPlayground: automatic security analysis of smartphone applications," in *Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy (CODASPY '13)*, pp. 209–220, ACM, February 2013.

[42] D. Sounthiraraj, J. Sahs, G. Greenwood, Z. Lin, and L. Khan, "SMV-HUNTER: Large Scale, Automated Detection of SSL/TLS Man-in-the-Middle Vulnerabilities in Android Apps," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, 2014.

[43] A. Memon, I. Banerjee, and A. Nagarajan, "GUI ripping: Reverse engineering of graphical user interfaces for testing," in *Proceedings of the 10th Working Conference on Reverse Engineering, WCRE 2003*, pp. 260–269, November 2003.

[44] R. Winsniewski, *Android–apktool: A tool for reverse engineering android apk files*, 2012.

[45] Y. Zhang, X. Luo, and H. Yin, "DexHunter: Toward extracting hidden code from packed android applications," in *Proceedings of the European Symposium on Research in Computer Security*, pp. 293–311, 2015.

[46] C. Hu and I. Neamtiu, "Automating GUI testing for android applications," in *Proceedings of the 6th International Workshop on Automation of Software Test, AST 2011, Co-located with ICSE 2011*, pp. 77–83, May 2011.

[47] S. Gunasekaran and V. Bargavi, "Survey on automation testing tools for mobile applications," *International Journal of Advanced Engineering Research and Science*, vol. 2, no. 11, pp. 2349–6495, 2015.

[48] Android Developers. Android debug bridge, 2014.

[49] E. H. Weigle, "High-speed and high-fidelity system and method for collecting network traffic," *The US Patent*, vol. 7, pp. 783-739, 2010.

[50] A. Orebaugh, G. Ramirez, and J. Beale, *Wireshark & Ethereal network protocol analyzer toolkit*, Elsevier, 2006.

[51] L. Richardson and S. Ruby, *RESTful web services*, O'Reilly Media, Inc., 2008.

[52] T. Berners-Lee, L. Masinter, and M. McCahill, "Uniform Resource Locators (URL)," RFC Editor RFC1738, 1994.

[53] S. Ramanujam and M. Karuppiah, "Designing an algorithm with high avalanche effect," *IJCSNS International Journal of Computer Science and Network Security*, vol. 11, no. 1, pp. 106–111, 2011.

[54] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 227–248, 1980.

[55] Xiaocong, Python wrapper of android uiautomator testing framework, 2014.

[56] mitmproxy. An interactive tls-capable intercepting http proxy, 2016.

[57] S. Parkes, *Sqlite: An embeddable sql database engine*, 2011.

[58] CNCERT/CC. China national vulnerability database, 2018.

[59] G. Tsirtsis and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)," RFC Editor RFC2766, 2000.

[60] A. Costin, J. Zaddach, A. Francillon, D. Balzarotti, and S. Antipolis, "A large-scale analysis of the security of embedded firmwares," in *Proceedings of the In USENIX Security Symposium*, pp. 95–110, 2014.

[61] A. Cui and S. J. Stolfo, "A quantitative analysis of the insecurity of embedded network devices: Results of a wide-area scan," in *Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC 2010*, pp. 97–106, December 2010.

[62] A. Cui, M. Costello, and J. S. Stolfo, "When firmware modifications attack: A case study of embedded exploitation," in *Proceedings of the NDSS*, 2013.

[63] H. Li, D. Tong, K. Huang, and X. Cheng, "FEMU: A firmware-based emulation framework for SoC verification," in *Proceedings of the 8th IEEE/ACM International Conference on Hardware/Software-Co-Design and System Synthesis, CODES+ISSS 2010*, pp. 257–266, usa, October 2010.

[64] Z. Wang, Y. Zhang, and Q. Liu, "RPFuzzer: A framework for discovering router protocols vulnerabilities based on fuzzing," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 8, pp. 1989–2009, 2013.

[65] J. Chen, W. Diao, Q. Zhao et al., *Iotfuzzer: Discovering memory corruptions in iot through app-based fuzzing*, 2018.

[66] W. G. Halfond and A. Orso, "A classification of sql-injection attacks and countermeasures," in *Proceedings of the IEEE International Symposium on Secure Software Engineering*, vol. 1 of *IEEE*, pp. 13–15, 2006.

[67] V. Philipp, F. Nentwich, N. Jovanovic, E. Kirda, K. Christopher, and V. Giovanni, "Cross site scripting prevention with dynamic data tainting and static analysis," in *Proceedings of the NDSS*, vol. 2007, p. 12, 2007.

[68] A. Barth, C. Jackson, and J. C. Mitchell, "Robust defenses for cross-site request forgery," in *Proceedings of the 15th ACM conference on Computer and Communications Security, CCS'08*, pp. 75–87, usa, October 2008.

[69] D. Michael, C. Kozyrakis, and N. Zeldovich, *Nemesis: Preventing authentication & access control vulnerabilities in web applications*, 2009.

[70] G. Pellegrino and D. Balzarotti, "Toward Black-Box Detection of Logic Flaws in Web Applications," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, 2014.

[71] R. Wang, S. Chen, X. Wang, and S. Qadeer, "How to shop for free online security analysis of cashier-as-a-service based web stores," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy, SP 2011*, pp. 465–480, May 2011.

[72] Android Developers. Ui/application exerciser monkey, 2012.

[73] A. Machiry, R. Tahiliani, and M. Naik, "Dynodroid: an input generation system for android apps," in *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE '13)*, pp. 224–234, ACM, August 2013.

[74] S. Anand, M. Naik, M. J. Harrold, and H. Yang, "Automated concolic testing of smartphone apps," in *Proceedings of the 20th ACM SIGSOFT International Symposium on the Foundations of Software Engineering, FSE 2012*, November 2012.

[75] Z. Chaoshun and L. Zhiqiang, "Smartgen: Exposing server urls of mobile apps with selective symbolic execution," in *In Proceedings of the 26th International Conference on World Wide Web*, pp. 867–876, International World Wide Web Conferences Steering Committee, 2017.

[76] N. Mirzaei, S. Malek, C. S. Păsăreanu, N. Esfahani, and R. Mahmood, "Testing android apps through symbolic execution," *ACM SIGSOFT Software Engineering Notes*, vol. 37, no. 6, p. 1, 2012.

[77] C. Weidong, K. Jayanthkumar, and J. H. Wang, "Discoverer: Automatic protocol reverse engineering from network traces," in *In USENIX Security Symposium*, p. 14, 1, 2007.

[78] B. Marshall, *The Protocol Informatics Project*, 2004.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5945312, 11 pages
https://doi.org/10.1155/2018/5945312

Research Article

A New Type of Countermeasure against DPA in Multi-Sbox of Block Cipher

Shuaiwei Zhang and Weidong Zhong

Key Laboratory of Network & Information Security of People's Armed Police, Engineering University of People's Armed Police, Xi'an 710086, China

Correspondence should be addressed to Shuaiwei Zhang; zsw36277@163.com

Received 7 March 2018; Accepted 12 April 2018; Published 28 June 2018

Academic Editor: Ximeng Liu

The Internet of Things (IoT) provides the network for physical devices, like home appliances, embedded with electronics, sensors, and software, to share and exchange data. With its fast development, security of IoT has become a crucial problem. Among the methods of attack, side-channel attack has proven to be an effective tool to compromise the security of different devices with improving techniques of data processing, like DPA and CPA. Meanwhile, many countermeasures have risen accordingly as well, such as masking and noise addition. However, their common deficiency was that every single countermeasure might not be able to protect the key information completely after statistical analysis. Sensitive information will be disclosed during differential power analysis of Sbox, since it is the only nonlinear component in block cipher. Thus, how to protect Sbox effectively was the highlight of researches. Based on Sbox-reuse concept proposed by Bilgin, this paper put forward a new type of a countermeasure scheme against DPA in multi-Sbox of block cipher. We first converted the multi-Sbox into 4×4 permutations and then reused permutation with the algebraic degree of more than one so as to turn it into a special reusable Sbox and then numbered 4×4 permutation input. Finally, we made these inputs of permutations completely random by masking. Since it was necessary to make the collected power consumption curve subject to alignment process in DPA by chosen-plaintext attack, this scheme combined the concept from DPA countermeasures of masking and noise addition. After the experiment with the proposed implementation, successful prevention of the attacker from accurately aligning the power consumption curve of the target Sbox has been proven, and the level of security has been improved by adding more random noise to protect key information and decrease the accuracy of statistical analysis.

1. Introductions

The Internet of Things (IoT) has been undergoing a fast and vast development in recent decades, which improved the efficiency and accuracy of many tasks in our life and brings more economic benefit. However, it also gives rise to the issue of security [1–4] especially in electronic devices [5]. Since 1996, when Paul Kocher proposed the side-channel attack [6], which will make the IoT applications unsecured and vulnerable, many improvements of attack method have induced the researches in countermeasures. Not only the range of cryptology security has extended from the initial security simply based on mathematical theory to comprehensive security of mathematical theory together with cryptography implementation, but also a huge thwart to the security of hardware device needed to be overcome in IoT. From the beginning of the 20th century until now, research achievements in this field emerge endlessly, such as power analysis [7, 8], timing analysis [9], electromagnetic analysis [10, 11], fault injection [12, 13], more advanced template attack [14, 15], Glitch attack [16, 17], and machine learning attack [18–23], among which power analysis has become the research emphasis for its easy implementation, lower costs, and higher successful attacking rate especially in lightweight block cipher [24]. Power analysis consists of simple power analysis, differential power analysis, and high-order power analysis, which are all based on the concept of recovering key with power difference generated by logic circuit composed with CMOS when processing "0" or "1" bit. Thanks to the vigorous development of attack theory, researches looking into countermeasures theory against power attack have also been in full swing. Over the years of study on countermeasures, the theories are basically divided

into two categories. One is the countermeasure scheme based on algorithm, such as random masking, shuffling, and hiding, characterized by low costs but low security [25–27]. The other is based on circuit level technique, featuring higher security, and more implementation costs, including two major technologies: sense amplifier based logic (SABL) [28] and wave dynamic differential logic (WDDL) [29]. In 2006, Svetla proposed the secret sharing and multiparty secure computation-based threshold implementation scheme[30], a well-developed scheme that can resist high-order DPA attack and Glitch attack [31–33], which possesses higher security and lower implementation costs. Inspired by threshold implementation and based on the concept of reused Sbox of block cipher, Bilgin proposed a design with compact implementation of multi-Sbox in 2015 [34], which greatly reduced the cost in implementation of DES.

Based on the study mentioned above, our paper puts forward a new type of a countermeasure scheme against DPA attack using concept of reused Sbox in [34]. We first convert the multi-Sbox into 4×4 permutation and reuse the permutation with the algebraic degree of more than one in order to turn it into a special and reusable Sbox and then number the 4×4 permutation input. Finally, each group of 4×4 permutation enters into Sbox after random masking; the power consumption curve is randomized by scrambling the data input from Sbox to have a higher probability of invalidating DPA. The security and feasibility of this scheme are verified by DES algorithm in our experiment.

The novel contributions of this paper are as follows.

(1) In this paper, we put forward a new type of countermeasure against DPA and it is divided into two phases. The first phase is converting the multi-Sbox into 4×4 permutations and reusing the permutation with the algebraic degree of more than one to turn it into a special reusable Sbox. The next phase is generating random input, which makes input data of Sbox completely random.

(2) Compared to other DPA masking techniques, the proposed scheme uses the value of masking as a selector and controls the sequence of data input of the multi-Sbox, instead of applying XOR or modular multiplication onto value of masking and original data. This not only results in reduced number of masking, but also increases the difficulty of aligning each power consumption curve for the attacker, which indirectly increases the noise for resisting DPA attacks.

(3) The proposed scheme can be applied to many other cryptographic algorithms based on multi-Sbox; the only difference is that, in the first phase of converting Sbox, different principles of generating permutations from Sbox that correspond to different algorithms should be considered in order to have a special and reusable Sbox and then proceed with the phase of generating random input.

This paper is organized as follows. Section 2 includes preliminaries of DPA procedures, physical basis of power attack, and concept of compact implementation. Section 3 introduces our countermeasure scheme. In Section 4, the results of the experiments are presented for validation of our scheme. Section 5 shows the security analysis of our countermeasure scheme. Section 6 is dedicated to conclusions.

2. Preliminaries

2.1. Differential Power Analysis. Differential power analysis (DPA) [7] is a side-channel attack scheme in DES algorithm put forward by Paul Kocher in 1999, whose model is based on hamming weight. The author believes that register requires different power when storing "0" and "1", which leads to the disclosure of power information. Compared with simple power analysis, differential power analysis recovers keys with statistical differential technology instead of requiring algorithm details. However, it has to collect much more consumption curves. This paper offers a conclusion of the typical process of DES algorithm differential power analysis as follows.

(1) Choose m sets of plaintexts $M_1, M_2, M_3, ..., M_m$ and encrypt each of them with the same key K to measure each set of consumption curve and mark it as $T_i[j]$; among which, i refers to the sets of plaintexts measured ($1 \le i \le m$) and j means the sampling sites.

(2) A distinguisher $D(M_i, b, K_s)$ is chosen to represent b of the median at the end of the first group of Sbox, among which M represents plaintext and $0 \le K_s \le 2^6$ stands for 6-bit key entering into the Sbox corresponding to bit b.

(3) According to the predicted K_s and the speculated value of distinguisher $D(M_i, b, K_s)$, all the consumption curves with the distinguisher value of 0 and 1 are averaged to record differential power curve, as revealed in

$$\Delta D[j] = \frac{\sum_{i=1}^{m} D(M_i, b, K_s) T_i[j]}{\sum_{i=1}^{m} D(M_i, b, K_s)} - \frac{\sum_{i=1}^{m} (1 - D(M_i, b, K_s)) T_i[j]}{\sum_{i=1}^{m} (1 - D(M_i, b, K_s))} \quad (1)$$

(4) During the observation of the current differential power curve, if an obvious large peak appears, the speculation about 6-bit key is considered as correct; if there is no remarkable peak, such speculation is incorrect and should continue.

(5) The 6-bit key that corresponds to other Sbox is predicted with the same scheme; the last 8 checking bits are obtained by brute force.

2.2. Physical Basis of Power Attack. Due to the improved manufacturing process, logic gates made by CMOS process possess lower power consumption, less costs, and stronger antijamming capability compared to TTL circuit. Almost all the mainstream cipher chips and equipment adopt devices of CMOS process to construct circuit. For the convenience of analysis, the following part offers an introduction to the physical property of CMOS device regarding its power consumption. Take inverter as an example with its internal structure shown in Figure 1.

As shown in Figure 1, this structure consists of two enhanced MOSFET, namely, N channel structure and P channel structure. When the low logic level is input, P channel conducts and N channel is cut off with high logic level output; when the high logic level is input, N channel conducts and P channel is cut off with low logic level output. The total power

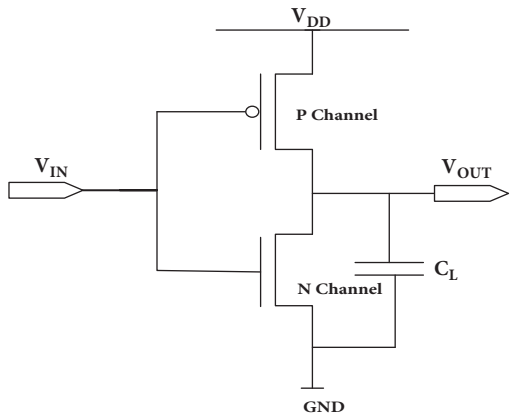


Figure 1: The internal structure of inverter.

consumption refers to the sum of static power and dynamic power which is

$$P_{total} = P_{stat} + P_{dyn} \quad (2)$$

When input V_{IN} of inverter stabilizes, the output V_{OUT} is also stable; under such circumstances, there are the conduction and the cut-off between P channel and N channel. It is found in actual measurement that a small amount of leak current I_{leak} is conveyed through the cut-off channel. Therefore, P_{stat} static power can be calculated according to the following:

$$P_{stat} = {I_{leak}}^2 V_{DD} \quad (3)$$

When the input V_{IN} of inverter changed, the output V_{OUT} changed accordingly. At this time, the dynamic power generated usually consists of two parts: one is P_{chrg}, power consumption of load capacitor C_L, while charge and discharge account for 85%; the other is P_{sc}, power consumption of top-down short-circuit current generated by the two concurrently conducting channels within very short period of time when the input level reaches $V_{DD}/2$ (accounts for 15%). Table 1 represents the constitution of the total power consumption of inverter with different inputs. Other logic gates based on CMOS process also have the above-mentioned consumption properties with much more complicated structure. Multielectrode MOS hopping superposition has made the generated dynamic power more obvious. Therefore, attackers can easily align the power consumption with the key, which serves as the principle of power attack after the hardware implementation of cryptographic algorithm.

2.3. Compact Implementation

2.3.1. Introduction. Sbox compact implementation is proposed by Bilgin based on threshold implementation in 2015 [34]. In threshold implementation, Sbox with algebraic degree of two will be implemented with at least three shares while Sbox is with algebraic degree of three with at least four shares. The circuit scale grows exponentially with the increasing number of shares. Therefore, researchers hope to replace the Sbox of higher algebraic degree with several serial Sbox of lower algebraic degree so as to ensure less resource consumption and less reduction of speed thanks to the employment of pipeline technology. Bilgin adopted the affine-equivalence technology to seek the public high-degree permutation of the eight Sbox in DES algorithm for reuse and then implemented the residual parts with algebraic degree of 1, thus reducing the hardware resources of Sbox by 50% [34].

2.3.2. Scheme Implementation. This scheme is dedicated to the 4×4 Sbox. As it can be seen as the permutations are of 4 bits, some of its properties deserve further study.

One permutation of n bits constitutes a symmetric group. An affine equivalence is defined as follows.

Definition 1. If there is a pair of affine permutation $A(x)$ and $B(x)$ which also meets $S_1 = B \circ S_2 \circ A$, $S_1(x)$ and $S_2(x)$ can be called affine equivalence.

The permutations that form affine equivalence in n bits permutations constitute a class. In this class, a permutation can be regarded as the representation element. The permutations in one class have the same algebra degree. At the same time, all the permutations are represented with $\mathscr{A}_{2^n}$ or $\mathscr{S}_{2^n} \backslash \mathscr{A}_{2^n}$.

Literature reveals that in 4-bit permutations, there are one affine class, six quadratic classes, and 295 cubic classes, among which all the affine class and quadratic classes all belong to $\mathscr{A}_{16}$; however, 144 out of 295 cubic classes belong to $\mathscr{A}_{16}$ and the remaining 151 are categorized into $\mathscr{S}_{16} \backslash \mathscr{A}_{16}$.

$\mathscr{M} = \{\mathscr{Q}_{004}, \mathscr{Q}_{012}, \mathscr{Q}_{293}, \mathscr{Q}_{294}, \mathscr{Q}_{299}, \mathscr{Q}_{300}\}$ is a set for 6 quadratic classes. It is proven in [19] that, in $\mathscr{A}_{16}$, permutations with any algebra degree can be represented by the elements from $\mathscr{M}$. The cubic class permutation in $\mathscr{S}_{16} \backslash \mathscr{A}_{16}$ can be represented by one or many secondary permutations in $\mathscr{A}_{16}$ and one-third of permutations in $\mathscr{S}_{16} \backslash \mathscr{A}_{16}$; however, the third permutation in $\mathscr{N} = \{\mathscr{Q}_{001}, \mathscr{Q}_{003}, \mathscr{Q}_{013}, \mathscr{Q}_{301}\}$ is often chosen to represent all because they possess some fine properties. Therefore, we aim to decompose different Sbox such that minimum number of nonlinear permutations is used to jointly describe all Sbox. Refer to [34] for more specific implementation of scheme.

3. Our Countermeasure Scheme

3.1. Classification of DPA Countermeasures Methods. DPA can speculate the key by subjecting the collected consumption curve to statistical difference. Therefore, the protection of any of the links can reduce the possibility of successful attack. Currently, the countermeasure methods for DPA usually fall into the following three categories.

(1) Countermeasures for the Leaked Information. In light of the low power consumption and fast speed, the mainstream hardware platforms all use chips based on CMOS process. It is defined by the working principle of CMOS gates that different power consumption will be generated when processing bit "0" and "1". Therefore, the countermeasures targeted the nature of disclosed information which is changing the processed "0" and "1" bit through certain technologies, such as adding mask.

Table 1: Constitution of the total power consumption of inverter with different inputs.

Initial State	Final State	Constitution of Total Power Consumption
0	0	P_{stat}
1	1	P_{stat}
0	1	$P_{stat} + P_{chrg} + P_{sc}$
1	0	$P_{stat} + P_{chrg} + P_{sc}$

(2) Countermeasures for the Implementation of Circuit Environment. As DPA is a method based on chosen-plaintext attack, it has high requirements for precision of measured consumption curve. If Signal to Noise Ratio (SNR) reduces, it will give rise to the high number of power consumption curves in attack and even result in the failure of attack. Therefore, the countermeasures for the implementation of circuit environment are to artificially introduce noise to the circuit in order to enhance attack difficulty and reduce the probability of successful attacks.

(3) Countermeasures for the Data Postprocessing. Data of the collected power consumption curve need to be aligned during the data postprocessing of DPA. The alignment is carried out by keeping the leaking points, which leak the sensitive information from different power consumption curves, aligning at the same point of time, to recover the key with a higher efficiency. The countermeasure of scrambling is employed to increase the difficulty of aligning different power consumption curves, in order to protect the circuit from leaking sensitive information.

This scheme is a combined countermeasure that includes countermeasures for the leaked information, the implementation of circuit environment, and data postprocessing. By utilizing the Sbox-reuse technology and randomly inputting data with masking, it can resist DPA because of raising random noise and preventing attackers from aligning the consumption curves corresponding with the key data with high probability in the data postprocessing.

3.2. Scheme Flow. In accordance with Nikova's theory, when the bit digit input $n \geq 4$, such permutation is secure. It is also noted that, in the existing cryptography scheme, the smallest Sbox is 4×4; under such circumstances, the minimum permutation of 4×4 in the Sbox framework turns out to be logical. The specific scheme flow is listed as follows.

(1) n independent parallel Sboxes are replaced by a special and reusable Sbox framework S', using the compact algorithm. The 4×4 Sbox in S' is numbered

$$[S_0(m_0), S_1(m_1) \cdots S_{n-1}(m_{n-1})] \Longrightarrow S', \tag{4}$$

in which m_{n-1} stands for the input of the $(n\text{-}1)^{\text{th}}$ 4-bit Sbox permutation, $S_{n-1}(m_{n-1})$ is the output of the $(n\text{-}1)^{\text{th}}$ 4-bit Sbox permutation, and S' is a special and reusable Sbox framework.

(2) A random number R_1 appears before the Sbox algorithm of circuit

$$R_1 = \left(r_1, r_2, \cdots r_{g(n)}\right) \tag{5}$$

Among which, $0 \leq R_1 \leq n-1$ and $g(n)$ stands for the binary bit digit that corresponds to n, the number of 4×4 Sbox participating in algorithm.

(3) The first 4×4 Sbox permutation entering S' is chosen based on R_1 value; the permutation is S_{R_1}.

(4) The random number R_1 and the input of 4×4 Sbox permutation entering S' are subjected to XOR operation with the input data as the random number of the next 4×4 Sbox permutation

$$m_{R_1} \oplus R_1 = R_2 \tag{6}$$

(5) Repeat Step (3) and Step (4); if the 4×4 Sbox that corresponds to the newly generated random number R_i has been chosen, then execute Step (6).

(6) R_i is subjected to XOR operation bit by bit, $R_i{}^*$ is obtained. Namely,

$$R_i{}^* = r_{g(n)\bullet(i-1)+1} \oplus r_{g(n)\bullet(i-1)+2} \oplus \cdots \oplus r_{g(n)\bullet(i-1)+g(n)} \tag{7}$$

(7) Choose a distinguisher $f(R_i{}^*)$.

$$f\left(R_i{}^*\right) = \begin{cases} S_{(R_i-1+n) \bmod n} & \text{if } R_i{}^* = 0 \\ S_{(R_i+1) \bmod n} & \text{if } R_i{}^* = 1 \end{cases} \tag{8}$$

If $R_i{}^*$, the result of bit-by-bit XOR operation of R_i is "0", the permutation $S_{(R_i-1+n) \bmod n}$ is chosen; if the result is "1", the permutation $S_{(R_i+1) \bmod n}$ is chosen. If the result is the selected 4×4 Sbox permutation, execute Step (7) until the 4×4 Sbox that has never been chosen appears and returns to Step (3).

(8) Repeat the above-mentioned steps until all n 4×4 Sbox permutations have all been chosen and entered the S'; Figure 2 is the flow of our scheme.

4. Experiments

This part mainly introduces the scheme implementation by using DES algorithm Sbox. Although it is known that DES algorithm of 56-bit key has been proven insecure in many applications, Triple-DES has been proven secure for its 112-bit key and widely applied to many electronic devices [35].

4.1. Implementation Steps of DES Algorithm Sbox Scheme. According to DES algorithm, its Sbox consists of *eight* parallel 6×4 Sboxes; in each Sbox, the first and sixth of its 6-bit input are used to determine four 4×4 permutaions. The 4-bit input consists of the second, third, fourth, and fifth of the 6-bit input; therefore, the *eight* 6×4 Sboxes actually consist of *thirty-two* 4×4 multi-Sbox. The DES algorithm Sbox is

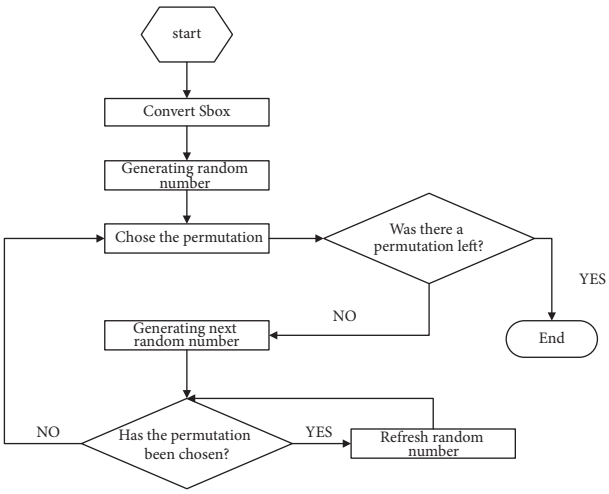


Figure 2: The flow of our scheme.

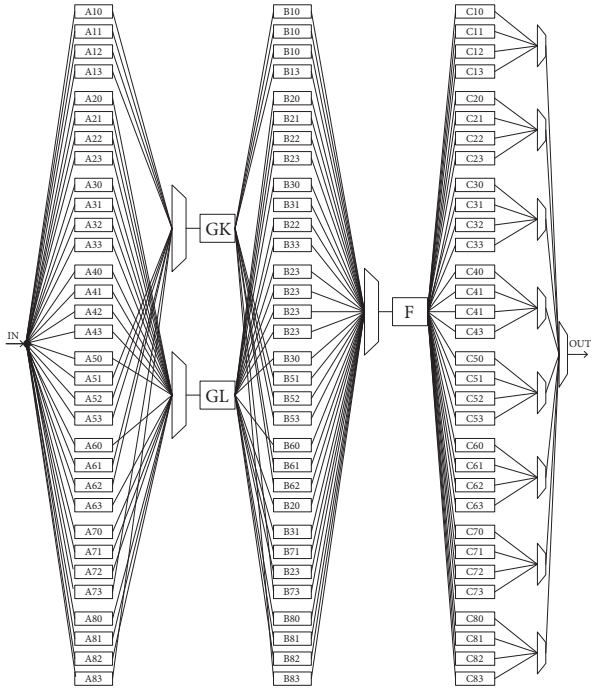


Figure 3: Special and reusable Sbox framework S'.

implemented according to the flows introduced in 3.2 with specific steps listed as follows.

(1) The *eight* 6×4 Sboxes in DES algorithm are converted into *thirty-two* 4×4 permutations. As suggested by Bilgin's reuse concept, n independent parallel Sboxes are converted into a special and reusable Sbox framework S'.

$$[S_0(m_0), S_1(m_1) \cdots S_7(m_7)] \Longrightarrow S' \tag{9}$$

The logic diagram after conversion is listed in Figure 3:

GK, GL, F, *Aij*, *Bij*, and *Cij* are known permutations. Refer to [34] for the specific permutations.

(2) As there are 8 Sboxes of 4×4 participating in DES algorithm, therefore, $n = 8$ and $g(n) = 3$. To satisfy the following algorithm requirements, we make $g(n)' = g(n) + 1 = 4$. $R_1 = (r_1, r_2, \cdots r_{g(n)'}) = (r_1, r_2, r_3, r_4)$ is the random number generated, $0 \le R_1 \le 15$.

(3) Suppose ${R_1}' = (r_2, r_3, r_4)$; the first 4×4 Sbox permutation entering S' is chosen based on the value of ${R_1}'$.

(4) The random number R_1 and the input of 4×4 Sbox permutation entering S' are subjected to XOR operation; the results obtained serve as the random number for the selection of the next 4×4 Sbox permutation.

$$m_{R_1'} \oplus R_1 = R_2 \tag{10}$$

(5) Repeat Step (3) and Step (4); if the 4×4 Sbox that corresponds to the newly generated random number R_i has been chosen, then execute Step (6).

(6) R_i is subjected to XOR operation bit by bit to obtain ${R_i}^*$.

$${R_i}^* = r_{3(i-1)+1} \oplus r_{3(i-1)+2} \oplus r_{3(i-1)+3} \oplus r_{3(i-1)+4} \tag{11}$$

(7) Choose a distinguisher function $f({R_i}^*)$

$$f({R_i}^*) = \begin{cases} S_{({R_i}'+7) \bmod 8} & \text{if } {R_i}^* = 0 \\ S_{({R_i}'+1) \bmod 8} & \text{if } {R_i}^* = 1 \end{cases} \tag{12}$$

If ${R_i}^*$, the result of bit-by-bit XOR operation of R_i is "0"; the permutation $S_{({R_i}'+7) \bmod 8}$ is chosen; if the result is "1", the permutation $S_{({R_i}'+1) \bmod 8}$ is chosen. If the result is the selected 4×4 Sbox permutation, execute Step (7) until the 4×4 Sbox that has never been chosen appears and returns to Step (3).

(8) Repeat the above-mentioned steps until all *eight* 4×4 Sboxes permutations have all been chosen and entered the S'. Finally, output all the parts of S simultaneously. The pseudocode of scheme is listed as Algorithm 1 where $S_{R_i'} = in$ means 4×4 Sbox $S_{R_i'}$ has never been chosen.

4.2. Experimental Results. The experiment environment of this scheme is presented in Table 2.

In accordance with 3.2, this scheme is subjected to experiment with the results listed as follows.

4.2.1. Resource and Operating Speed Result. On one hand, Tables 3 and 4 are the resources consumed by the algorithm in the FPGA platform between the scheme proposed in this paper and original scheme. It can be seen that the total logic elements of this scheme are 33k, which is roughly eightfold the original scheme. But considering the whole resources in FPGA chip (about 114480 logic elements), our scheme is still practical to operate.

On the other hand, the speed of our countermeasure implementation is up to *80M* and an average number of periods of 41 are needed to process one group of plaintext.

4.2.2. Security Result. Figures 4 and 5 are the DPA result comparison between original DES algorithm and our countermeasure scheme for each Sbox within right key (both are using fourth-order cumulate to make result more obviously). Apparently, after 800 power traces of DPA, we found that

Input: R_1, muti-Sbox
Output: S
(1) ***function***$(R_1$,*muti-Sbox, S)*
(2) *Convert muti-Sbox to* S'
(3) *Number the* 4×4 *Sbox start at* S_0
(4) ***input*** *Random masking* $R_1(r_1, r_2, r_3, r_4)$
(5) ***for*** $i = 1$ *to* 8
(6) ***do*** $R_i' \longleftarrow (r_{3i-1}, r_{3i}, r_{3i+1})$
(7) *Chose* $R_i th$ 4×4 *Sbox* $S_{R_i'}$
(8) ***if*** $(S_{R_i'} = in)$
(9) *Save* $S_{R_i'}$
(10) $R_i \longleftarrow (m_{R_i'} \oplus R_i)$
(11) ***go to*** *Line (7)*
(12) ***else***
(13) $R_i^* \longleftarrow (r_{3i-2}, r_{3i-1}, r_{3i}, r_{3i+1})$
(14) ***if*** $(R_i^* = 0)$
(15) *Chose* $S_{(R_i+7) \bmod 8}$
(16) ***go to*** *Line (8)*
(17) ***else if*** $(R_i^* = 1)$
(18) *Chose* $S_{(R_i+1) \bmod 8}$
(19) ***go to*** *Line (8)*
(20) ***end if***
(21) ***end if***
(22) ***end for***
(23) $S \longleftarrow [S_{R_1'} \| S_{R_2'} \| S_{R_3'} \| S_{R_4'} \| S_{R_5'} \| S_{R_6'} \| S_{R_7'} \| S_{R_8'}]$
(24) ***return*** S
(25) ***end function***

Algorithm 1: The pseudocode of scheme.

Table 2: Experimental environment.

Tools	Pattern
PC	Lenovo Thinkpad x240 core i7
System	Windows7
Software	Quatus prime 15.1, Modelsim15.1
FPGA	Altera EP4CE115F2317
Oscilloscope	Tektronix MSO5204B
Differential probe	Tektronix TDP3500
Regulated power supply	DH-1719

Table 3: Total logic elements of original scheme.

Number	Parameters	Values
1	Total logic elements	4137
2	Total combinational function	3856
3	Dedicated logic registers	1144
3	Total registers	1144
4	Total pins	194

Table 4: Total logic elements of this scheme.

Number	Parameters	Values
1	Total logic elements	33602
2	Total combinational function	30997
3	Dedicated logic registers	7385
3	Total registers	7385
4	Total pins	187

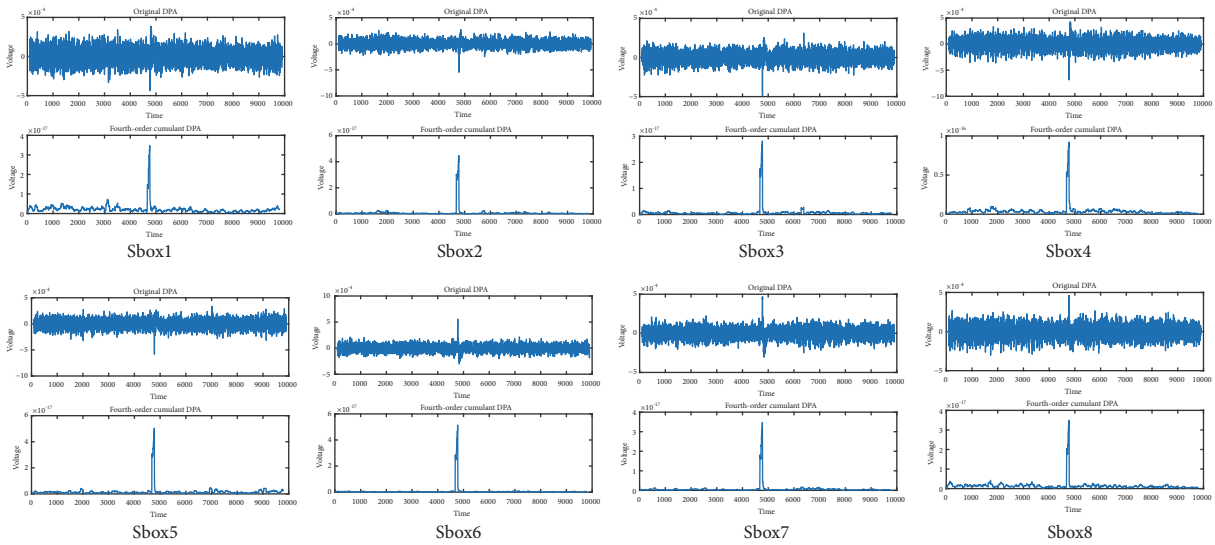


Figure 4: DPA using original scheme with 800 traces.

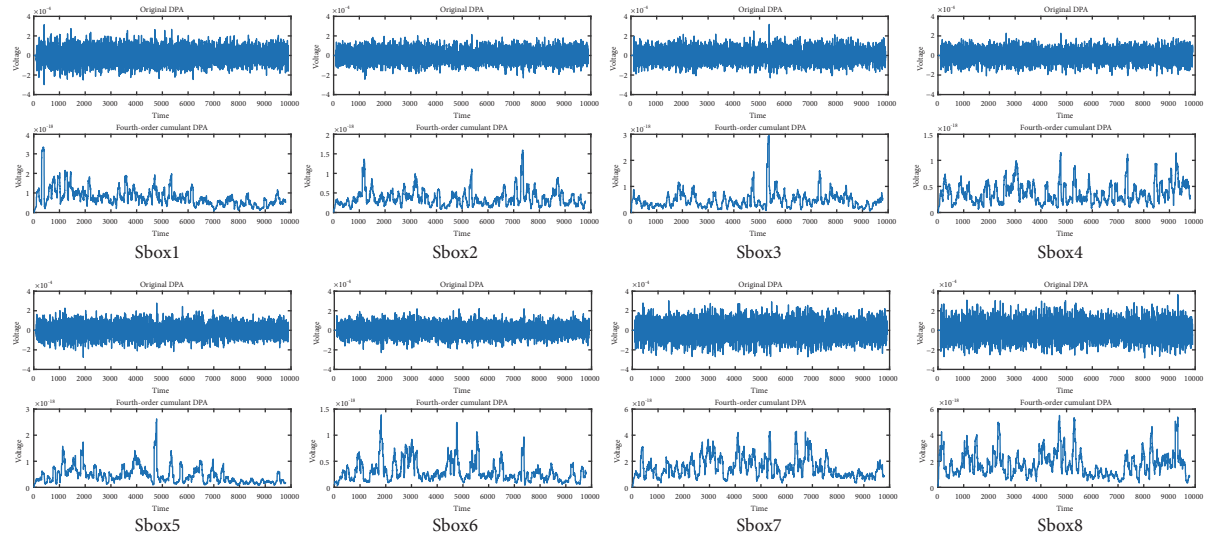


Figure 5: DPA using our scheme with 5000 traces.

there was one obvious peak in original DPA of DES algorithm for each Sbox. On the contrary, several peaks in our scheme with 5000 traces we found in Figure 5 were "ghost" peaks, which leads to wrong key corresponding to the target Sbox. Therefore, we conclude that our countermeasure scheme in Sbox of DES can improve the security of implementation against DPA.

5. Security Analysis

5.1. Theory of DPA Power Analysis. The DPA power attack is target at the output of register corresponding to the Sbox in cryptographic algorithms circuit. Although sensitive information might leak from the logic circuits inside the Sbox and be used by attackers for Glitch attack, we mainly focus on DPA, and our scheme is offering protection to registers.

Take 4 × 4 Sbox as an example with the specific circuit diagram shown in Figure 6, in which power region is at where attackers want to collect power consumption.

X_i represents the input of Sbox, Y_i stands for output of Sbox as well as the input of register, and Q_i is the output of register.

The internal structure of one register is shown as Figure 7.

One register consists of a few control components and one D trigger; the D trigger is composed of 6 NAND gates shown in Figure 8.

Therefore, in line with the analysis of 2.2, when an obvious large hopping takes place after D is input, CMOS transistors within *eight* NAND gates, *one* OR gate, and *one* NOT gate will instantaneously generate dynamic power consumption. Attackers can attack the device according to the power consumption collected and by means of DPA.

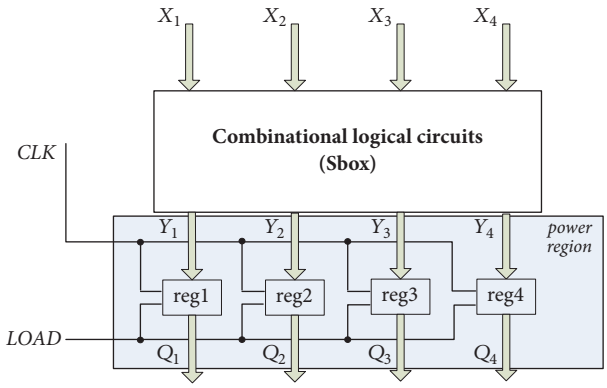


Figure 6: The corresponding register of Sbox.

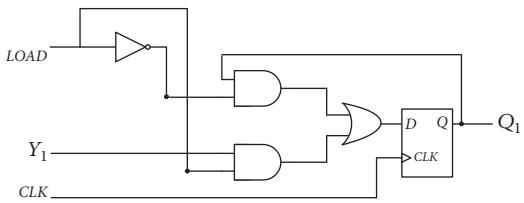


Figure 7: Internal structure of one register.

5.2. Analysis of the Security of Traditional Power Model. It is shown in 2.2 that, in cryptographic calculation circuit, the total power consumption is the sum of dynamic power and static power:

$$P_{total} = P_{stat} + P_{dyn} \tag{13}$$

Due to the output of register, different hopping corresponds to different power consumption and is represented by $P_{0\longrightarrow1}$, $P_{1\longrightarrow0}$, $P_{0\longrightarrow0}$, and $P_{1\longrightarrow1}$; and, obviously, $P_{0\longrightarrow0} = P_{1\longrightarrow1} = P_{stat}$. Therefore, as shown by 5.1,

$$P_{0\longrightarrow1} = a\left(P_{AND} + P_{OR} + P_{NOT}\right) + n + P_{stat} \tag{14}$$

$$P_{1\longrightarrow0} = a\left(P_{AND}{}' + P_{OR}{}' + P_{NOT}{}'\right) + n + P_{stat}, \tag{15}$$

in which a is a constant coefficient, P_{AND}, P_{OR}, and P_{NOT} are dynamic power consumption in logic gates, and n is noise. As abundant facts have proven that $P_{0\longrightarrow1} > P_{1\longrightarrow0}$, it is believed that

$$P_{0\longrightarrow1} = P_{1\longrightarrow0} + \varepsilon \tag{16}$$

As hamming weight model is adopted in DPA, therefore,

$$P_0 = \frac{\left(P_{1\longrightarrow0} + P_{0\longrightarrow0}\right)}{2} \tag{17}$$

$$P_1 = \frac{\left(P_{0\longrightarrow1} + P_{1\longrightarrow1}\right)}{2} \tag{18}$$

The following part offers an analysis of the DPA security. If attackers succeed in guessing the key, refer to Table 5.

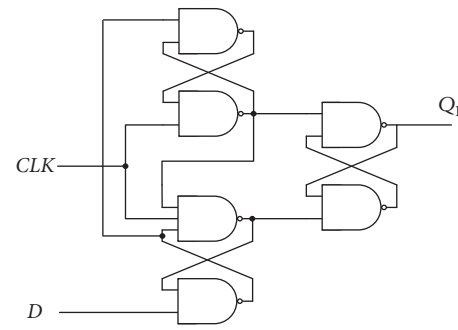


Figure 8: Internal structure of D trigger.

Table 5: Situation when attackers succeed in guessing the key.

	Guess value	True value	power
Possibility1	0	0	P_0
Possibility2	1	1	P_1

In accordance with DPA principle, power consumption with the guessed value of 1 minus the power consumption with the guessed value of 0 is represented as follows:

$$DP = P_1 - P_0 = \frac{\left(P_{0\longrightarrow1} + P_{1\longrightarrow0}\right)}{2} = \frac{\varepsilon}{2} \tag{19}$$

If attackers fail to guess the key, refer to Table 6.

Power consumption with the guessed value of 1 minus the power consumption with the guessed value of 0 is represented as follows:

$$DP = \left(P_0 + P_1\right) - \left(P_0 + P_1\right) = 0 \tag{20}$$

Therefore, the possibility of guessing the key correctly for the attackers is 1/16.

5.3. Analysis of the Security in Our Scheme. The proposed scheme combines the methods of conversion of Sbox and randomizes the input to resist DPA. Table 7 lists the situation of guessing key in our scheme.

As it is shown in the table, the attackers can only locate the position of leaking point on the power consumption curve of target Sbox, when the sequence of speculating Sbox and the key to the corresponding Sbox are both correct. In other cases, the positions of leaking points are random. Compared to conventional masking schemes, there are 3 advantages.

(1) Multi-Sboxes will rely on each other, due to existence of the selector for value of masking.

Keys of conventional cryptographic algorithms can be successfully recovered by DPA because their multi-Sboxes are parallel independently; DPA is able to successfully recover key from each single Sbox to get the corresponding key. However, the proposed scheme utilizes a special reusable Sbox, having random sequence of encrypting data in Sboxes each time, resulting in different success rate of recovering key from different Sboxes, shown in Table 8. Also depicted in Figure 9, the success rate of recovering key from corresponding Sbox with proposed scheme is decreasing exponentially compared to conventional method.

Table 6: Situation when attackers fail to guess the key.

	Guess value	True value	power
Possibility1	0	0	P_0
Possibility2	1	1	P_1
Possibility3	0	1	P_1
Possibility4	1	0	P_0

Table 7: Situation of guessing key in our scheme.

	Guess Sbox	Guess value	True value	power
Possibility1	correct	correct	sure	sure
Possibility2	correct	wrong	random	random
Possibility3	wrong	correct	random	random
Possibility4	wrong	wrong	random	random

Table 8: The success rate of recovering key corresponding n^{th} Sbox.

	Sequence of speculating Sbox	Guessing the value of key	Success rate
$Sbox^{1st}$	1/8	1/64	$(1/2)^9$
$Sbox^{2nd}$	1/8	$(1/64)^2$	$(1/2)^{15}$
$Sbox^{3rd}$	1/8	$(1/64)^3$	$(1/2)^{21}$
$Sbox^{4th}$	1/8	$(1/64)^4$	$(1/2)^{27}$
$Sbox^{5th}$	1/8	$(1/64)^5$	$(1/2)^{33}$
$Sbox^{6th}$	1/8	$(1/64)^6$	$(1/2)^{39}$
$Sbox^{7th}$	1/8	$(1/64)^7$	$(1/2)^{45}$
$Sbox^{8th}$	1/8	$(1/64)^8$	$(1/2)^{51}$

(2) It is difficult to align power consumption curves increases during data postprocessing.

Since the principle of DPA is to align the position of leaking points for sensitive information, the statistical differential method is then applied to recover the key. However, the positions of leaking points for sensitive information on different power consumption curves are not located within one period with a high possibility for proposed scheme; additional measures need to be applied to move power consumption curves during data postprocessing for attacks.

(3) Increased noise exists for DPA attack.

Since the noise generated during DPA attack can be eliminated with statistical differential method, the noise will be on superposition randomly while processing data of each single Sbox during process of encryption for the proposed scheme, as the method for inputting data is based on Sbox in series randomly. Moreover, this noise cannot be eliminated by statistical differential method; thus, even if the attackers moved the power consumption curves precisely and successfully recovered the keys corresponding to Sboxes, the attack will still end up in failure because of the interference of the noises in the result.

6. Conclusions

This paper proposed a countermeasure scheme of multi-Sbox against DPA attack, based on the multi-Sbox-reuse concept and random input for IoT applications security. Compared to other DPA masking techniques, the proposed scheme uses the value of masking as a selector and controls the sequence of data input of the multi-Sbox, instead of applying XOR or modular multiplication onto value of masking and original data. This not only results in reduced number of masking, but also increases the difficulty of aligning each power consumption curve for the attacker, which indirectly increases the noise for resisting DPA attacks. With the experiments, our scheme is supported correctly and accurately by experimental evidence of power data for DES algorithm processing in our DPA platform as Figure 10 has shown.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China (Grant no. 2017YFB0802000), National Natural

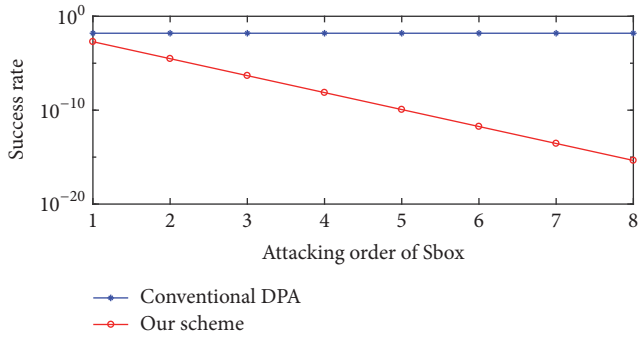


Figure 9: Comparison between conventional DPA and our scheme in success rate.

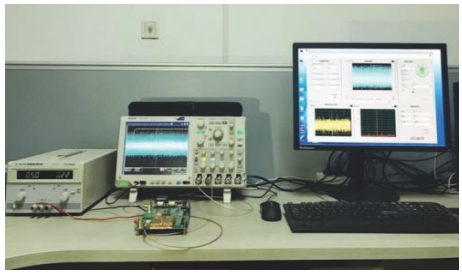


Figure 10: DPA platform.

Science Foundation of China (Grant nos. U1636114, 61772550, and 61572521), and National Cryptography Development Fund of China (Grant no. MMJJ20170112).

References

[1] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin, "Smart community: an internet of things application," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 68–75, 2011.

[2] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the internet of things: perspectives and challenges," *Wireless Networks*, vol. 20, no. 8, pp. 2481–2501, 2014.

[3] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: the road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[4] M. Chiang and T. Zhang, "Fog and IoT: an overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[5] A. A. Pammu, K.-S. Chong, W.-G. Ho, and B.-H. Gwee, "Interceptive side channel attack on AES-128 wireless communications for IoT applications," in *Proceedings of the 2016 IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2016*, pp. 650–653, Republic of Korea, October 2016.

[6] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Advances in Cryptology-CRYPTO '96*, Lecture Notes in Computer Science, pp. 104–113, Springer, Berlin, Germany, 1996.

[7] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conference*, pp. 388–397, Springer, Berlin, Germany, 1999.

[8] W. Wang, Y. Yu, F. Standaert, J. Liu, Z. Guo, and D. Gu, "Ridge-Based DPA: Improvement of Differential Power Analysis For Nanoscale Chips," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1301–1316, 2018.

[9] M.-L. Akkar and C. Giraud, "An implementation of DES and AES, secure against some attacks," in *Proceedings of the third International Workshop on Cryptographic Hardware and Embedded Systems, CHES 2001*, vol. 2162, pp. 309–318, Springer, Berlin, Germany, May 2001.

[10] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic Analysis: Concrete Results," in *Cryptographic Hardware and Embedded Systems — CHES 2001*, pp. 251–261, Springer, Berlin, Germany, 2001.

[11] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Power and electromagnetic analysis: Improved model, consequences and comparisons," *Integration, the VLSI Journal*, vol. 40, no. 1, pp. 52–60, 2007.

[12] G. Piret and J. Quisquater, "A differential fault attack technique against spn structures, with application to the AES and khazad," in *Cryptographic Hardware and Embedded Systems - CHES 2003*, vol. 2779 of *Lecture Notes in Computer Science*, pp. 77–88, Springer, Berlin, Germany, 2003.

[13] C. H. Kim and J.-J. Quisquater, "Faults, injection methods, and fault attacks," *IEEE Design & Test of Computers*, vol. 24, no. 6, pp. 544-545, 2007.

[14] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 13–28, Springer, Berlin, Germany, 2003.

[15] L. Lerman, R. Poussier, O. Markowitch et al., "Template attacks versus machine learning revisited and the curse of dimensionality in side-channel analysis: extended version," *Journal of Cryptographic Engineering*, pp. 1–13, 2017.

[16] S. Mangard, N. Pramstaller, and E. Oswald, "Successfully Attacking Masked AES Hardware Implementations," in *nternational Workshop on Cryptographic Hardware and Embedded Systems*, vol. 2005, pp. 157–171, Springer, Berlin, Germany.

[17] S. Mangard, T. Popp, and B. M. Gammel, "Side-channel leakage of masked CMOS gates," in *Topics in cryptology–CT-RSA 2005*, pp. 351–365, Springer, Berlin, Germany, 2005.

[18] S. Zhang, X. Yang, W. Zhong, and Y. Wei, "An improved combinational side-channel attack on S-box in block cipher," *Journal of Internet Technology*, vol. 17, no. 1, pp. 157–166, 2016.

[19] G. Hospodar, B. Gierlichs, E. De Mulder, I. Verbauwhede, and J. Vandewalle, "Machine learning in side-channel analysis: A first study," *Journal of Cryptographic Engineering*, vol. 1, no. 4, pp. 293–302, 2011.

[20] E. Cagli, C. Dumas, and E. Prouff, "Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures," in *International Conference on Cryptographic Hardware and Embedded Systems*, vol. 2017, pp. 45–68, Springer International Publishing, Champa.

[21] S. Hou, Y. Zhou, H. Liu, and N. Zhu, "Wavelet support vector machine algorithm in power analysis attacks," *Radioengineering*, vol. 26, no. 3, pp. 890–902, 2017.

[22] L. Lerman, Z. Martinasek, and O. Markowitch, "Robust profiled attacks: Should the adversary trust the dataset?" *IET Information Security*, vol. 11, no. 4, pp. 188–194, 2017.

[23] W. Shan, S. Zhang, and Y. He, "Machine learning based side-channel-attack countermeasure with hamming-distance redistribution and its application on advanced encryption standard," *IEEE Electronics Letters*, vol. 53, no. 14, pp. 926–928, 2017.

[24] S. Tang, W. Li, and J. Wu, "Power analysis attacks against FPGA implementation of KLEIN," *Security and Communication Networks*, 2017.

[25] M.-L. Akkar and C. Giraud, "An implementation of DES and AES, secure against some attacks," in *Proceedings of the third International Workshop on Cryptographic Hardware and Embedded Systems, CHES 2001*, vol. 2162, pp. 309–318, Springer, May 2001.

[26] M. Akkar, R. Bévan, and L. Goubin, "Two Power Analysis Attacks against One-Mask Methods," in *International Workshop on Fast Software Encryption*, vol. 2004, pp. 332–347, Springer, Berlin, Germany.

[27] A. A. Ding, L. Zhang, Y. Fei, and P. Luo, "A Statistical Model for Higher Order DPA on Masked Devices," in *nternational Workshop on Cryptographic Hardware and Embedded Systems*, pp. 147–169, Springer, Berlin, Germany, 2014.

[28] K. Tiri, M. Akmal, and I. Verbauwhede, "A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards," in *Proceedings of the 28th European Solid-State Circuits Conference (ESSCIRC '02)*, pp. 403–406, IEEE, September 2002.

[29] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, vol. 1, pp. 246–251, IEEE Computer Society, February 2004.

[30] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold implementations against side-channel attacks and glitches," in *nternational Conference on Information and Communications Security*, pp. 529–545, Springer, Berlin, Germany, 2006.

[31] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "Trade-Offs for Threshold Implementations Illustrated on AES," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 7, pp. 1188–1200, 2015.

[32] A. Shahverdi, M. Taha, and T. Eisenbarth, "Lightweight side channel resistance: threshold implementations of Simon," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 66, no. 4, pp. 661–671, 2017.

[33] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "Higher-order threshold implementations," in *nternational Conference on the Theory and Application of Cryptology and Information Security*, vol. 8874, pp. 326–343, Springer, Berlin, Germany, 2014.

[34] B. Bilgin, M. Knežević, V. Nikov, and S. Nikova, "Compact Implementations of Multi-Sbox Designs," in *International Conference on Smart Card Research and Advanced Applications*, pp. 273–285, Springer, 2015.

[35] Y. Ren, L. Wu, H. Li et al., "Key recovery against 3DES in CPU smart card based on improved correlation power analysis," *Tsinghua Science and Technology*, vol. 21, no. 2, pp. 210–220, 2016.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9576978, 15 pages
https://doi.org/10.1155/2018/9576978

Research Article

Cluster-Based Arithmetic Coding for Data Provenance Compression in Wireless Sensor Networks

Qinbao Xu, Rizwan Akhtar, Xing Zhang, and Changda Wang

School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China

Correspondence should be addressed to Changda Wang; changda@ujs.edu.cn

Received 8 March 2018; Revised 25 May 2018; Accepted 5 June 2018; Published 27 June 2018

Academic Editor: Kim-Kwang Raymond Choo

In wireless sensor networks (WSNs), data provenance records the data source and the forwarding and the aggregating information of a packet on its way to the base station (BS). To conserve the energy and wireless communication bandwidth, the provenances are compressed at each node along the packet path. To perform the provenances compression in resource-tightened WSNs, we present a cluster-based arithmetic coding method which not only has a higher compression rate but also can encode and decode the provenance in an incremental manner; i.e., the provenance can be zoomed in and out like Google Maps. Such a decoding method raises the efficiencies of the provenance decoding and the data trust assessment. Furthermore, the relationship between the clustering size and the provenance size is formally analyzed, and then the optimal clustering size is derived as a mathematical function of the WSN's size. Both the simulation and the test-bed experimental results show that our scheme outperforms the known arithmetic coding based provenance compression schemes with respect to the average provenance size, the energy consumption, and the communication bandwidth consumption.

1. Introduction

Wireless sensor networks (WSNs) are composed of a large number of low-cost, low-power, and randomly distributed wireless sensor nodes (nodes, for short), which are intended to monitor physical or environmental data from the detecting areas and cooperatively pass the data to the base station (BS) or a desired actuator through wireless communication. They are widely deployed in a vast number of different application areas, such as health monitoring, meteorology, and military operations. Because of the diversity of the environment and the large number of sensor types involved, in order to use the reliable information to make an accurate decision, it is essential to evaluate the trustworthiness of the received data at the base station (BS) of a WSN. In practice, there are some examples of significant losses because the faulty data are used [1].

In a multihop WSN, provenance of each data packet presents the history of the data acquisition and the actions performed on the data while the data are transmitted to the BS. Provenance provides the knowledge about how the data come to be in its current state, including where the data originated, how it was generated, and the operations it has undergone since its generation. Therefore, provenance plays an important role in assessing the data's trustworthiness [2, 3]. With the increase in packet transmission hops, the provenance's size expands rapidly and sometimes the size even largely exceeds the size of the data packet itself. WSNs are a kind of resource constrained network. Because of storage and computational resource constraints, sensor nodes deployed in WSN do not have the ability to manipulate the provenance if its size is very large. Generally, sensor nodes utilize batteries to supply power, so the amount of energy is limited. Data transmission is the major part of the energy consumption. When the sensing data item is fixed, the packet size mainly depends on the provenance. Besides, the transmission channels do not have sufficient capacity for transmitting large provenance.

As a result, in large-scale WSNs, the provenances generally cannot be directly and completely transmitted due to both the bandwidth and the energy constraints on wireless sensor nodes. For the same reason, the provenance encoding schemes for wired computer networks, e.g., the works of [4–6], are not applicable for WSNs. Hence, several lightweight

or compact data provenance schemes [7–9] as well as the compression schemes [1, 10, 11] have been proposed. The lightweight schemes drop some information with less significance in the provenance, e.g., the provenance graph's topology, and then shorten the provenances size. The compression schemes decrease the provenances size through arithmetic coding [12], LZ77 [13], and LZ78 [14]. Note that for a given packet path, the provenance compression has a determined entropy upper bond according to Shannon's theory. Even the dictionary-based scheme [10] can achieve the highest provenance compression rate up to date, in a large-scale network the provenance overload problem is inevitable.

To mitigate the average provenance size increases as well as utilize the provenance data efficiently, we propose a CBP (cluster-based provenance) encoding scheme for WSNs. The CBP scheme focuses on encoding and decoding the provenance incrementally (like Google Maps, can be zoomed in and out according to the user's requirement) at the BS. The specific contributions of the paper are as follows:

(i) We proposed a cluster-based lossless provenance arithmetic encoding scheme (CBP) for WSNs. Our approach not only has the ability of encoding and decoding the provenance incrementally, but also achieves a higher average provenance compression rate.

(ii) We derived the optimal cluster size for the CBP scheme as a mathematical function of the number of the nodes in a WSN.

(iii) We provided a detailed performance analysis, simulation, and experimental evaluations of our CBP scheme and a comparison with other related schemes.

The rest of the paper is organized as follows: Section 2 surveys the related works. Section 3 introduces the system model and the related background. Section 4 gives an overview of our method. Section 5 describes our proposed encoding and decoding approaches for simple and aggregated provenances, respectively. The cases study is presented in Section 6. Section 7 theoretically analyzes the performances of our method. Sections 8 and 9 show the simulation and experimental results, respectively. Section 10 concludes the paper.

2. Related Work

Shebaro et al. [9] proposed an in-packet Bloom filter (BF, for short) based lightweight secure provenance scheme, in which every node on packet's path is embedded into an array with fixed size through a set of hash functions. A BF is a simple but space efficient randomized data structure which supports fast membership queries with false positive, where the false positive rate depends on the array's size. Alam et al. [7] proposed an energy-efficient provenance encoding scheme based on the probabilistic incorporation of the nodes' IDs into the provenance, in which the entire provenance is scattered into a series of packets that are transmitted along the same route. Therefore, all the sections carried by the packets have to be retrieved correctly at the BS and then the provenance can be decoded. The major advantage of such a method is that it successfully limits the size of the provenance attached to a single packet, whereas the drawback is that it has a higher decoding error rate compared to the methods that encode the entire provenance into a single packet. The above methods are all lossy provenance encoding schemes because the topology information of the WSN is not included.

Hussain et al. [1] proposed an arithmetic coding based provenance scheme (ACP). The ACP scheme assigns a global cumulative probability for each node according to its occurrence probability in the used packet paths. For a given packet path in a WSN, the ACP scheme utilizes the global cumulative probability of the source node as the initial coding interval, and then the cumulative probabilities acquired from the associated probabilities derived from each connected nodes pair are used to generate the provenance; i.e., all the nodes IDs along the path are sequentially encoded into a half-open interval through arithmetic coding. Unlike most of the known provenance schemes whose average provenance size is directly in proportion to the increases in the packet transmission hops, the ACP scheme's provenance size is decided by the packet path's occurrence probability in a WSN.

Wang et al. [10] proposed a dictionary-based provenance encoding scheme (DP). The DP scheme treats every packet path as a string of symbols consisting of the node IDs along the packet path. Just like building a dictionary for symbol strings in LZ compression [13, 14], the DP scheme builds the dictionary for packet paths with the used packet paths at every node and the BS holds a copy of each node's dictionary. Therefore, the provenance can be encoded through an index or a series of indices of the packet paths at each node along the packet path and be decoded by looking up the dictionaries at the BS. When the topology of the WSN is relatively stable, the provenance size under the DP scheme can be even shorter than the provenance's entropy; on the contrary, the provenance compression rate increases drastically with the quick change of the WSN's topology. Wang et al. [11] proposed a dynamic Bayesian network based provenance scheme (DBNP) for WSNs. The DBNP scheme encodes edges instead of node IDs along the packet path into the provenance by overlapped arithmetic coding. Compared to the known lossless provenance schemes, a higher provenance compression rate can be achieved by the DBNP scheme. Furthermore, such a scheme is not sensitive to the WSN's topology changes. However, applying the overlapped arithmetic coding leads to decoding false positives [15], where extra knowledge is added to eliminate the false positives. Therefore, it is difficult to make a tradeoff between the acceptable false positive rate and the optimum compression rate.

All the approaches mentioned above focus on how to mitigate the provenances size rapid expansion in WSNs. However, none of these approaches supports the provenance incremental encoding and decoding, which can raise both the efficiencies of the provenances decoding and the data trust assessment.

3. Background and System Model

In this section, we provide a brief primer on arithmetic coding. We also introduce the system model applied in this

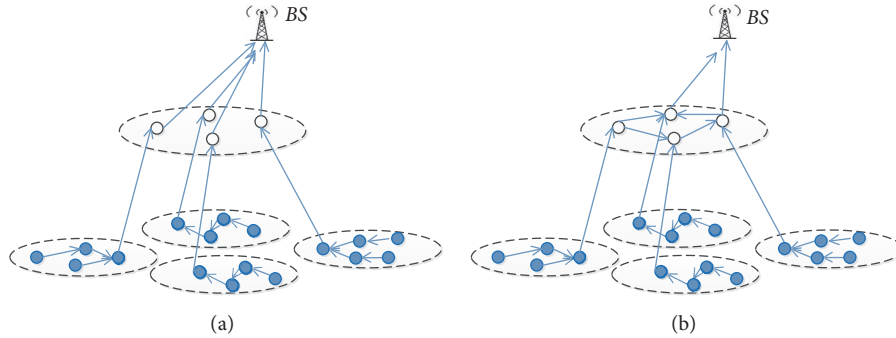


Figure 1: The two data collection modes in cluster-based WSNs.

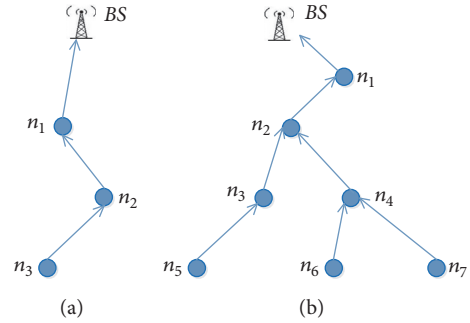


Figure 2: (a) Simple provenance. (b) Aggregated provenance.

paper. Some of these definitions are partly from our previous work [1, 10, 11].

3.1. Clustering Model. In WSNs, clustering management refers to selecting a series of nodes as cluster-heads according to a certain communication protocol, e.g., EEHC (Energy-Efficient Hierarchical Clustering) [16]. A cluster-head aggregates the data generated by the nodes in its cluster and then sends such data to the BS.

Figure 1 shows the two data collection modes in cluster-based WSNs, where the solid circles and the empty circles denote the member nodes of a cluster and the cluster-heads, respectively. Figure 1(a) shows an example of the single-hop communication between the cluster-heads and the BS, which is applied for small-scale WSNs; Figure 1(b) shows an example of multihop communication among the cluster-heads and the BS, which is applied for large-scale WSNs because some cluster-heads cannot reach the BS in one hop.

3.2. Provenance Model. In WSNs, the provenance of a data packet refers to where the data packet is generated and how it is transmitted to the BS [1, 2, 8]. In our provenance model, a data source is a node generating data periodically through the sensors attached to the node; a forwarder is a node that relays the packet toward the BS along the packet path; an aggregator is a node that aggregates two or more received data packets from its upstream nodes as a new one and then sends the new packet toward the BS. The aggregator nodes in our scheme are not selected. While being transmitted toward the BS, only the packets that fulfill the aggregation conditions are aggregated [17]. Note that such a process results in the provenances aggregation accordingly.

Each packet contains the following: (i) a unique packet's sequence number; (ii) a cluster-head sequence number; (iii) data source node ID; (iv) data value; (v) provenance; and (vi) a message authentication code (MAC), which binds the provenance and its data together to prevent any unauthorized modification.

There are two different kinds of provenance in WSNs. Figure 2(a) presents a simple provenance, where data is generated at leaf node n_3 and forwarded by nodes n_2 and n_1 toward the BS; Figure 2(b) shows an aggregated provenance, where data are aggregated at nodes n_4 and n_2 on the way to the BS. The aggregated provenance can be presented as a tree through a recursive expression $\langle((a)(b)), c\rangle$, where c denotes the root and (a) and (b) denote the left and the right subtrees, respectively. Therefore, the aggregated provenance in Figure 2(b) has the form

$$\langle(((n_5), n_3))(((n_6)(n_7), n_4)), n_2), n_1\rangle. \quad (1)$$

Without loss of generality, the formal definition of data provenance in a WSN is as follows [11].

Definition 1 (provenance). For a given data packet d, the provenance p_d is a directed acyclic graph $G(V, E)$, each vertex $v_x \in V$, where $1 \le x \le |V|$ and $|V|$ represents the cardinality of the set V, is attributed to a specific node n_i, and represents the provenance record for that node. One refers to this relation as $HOST(v) = n_i$; i.e., node n_i is the host of v_x. Each

Table 1: Probability model for the alphabet N= {a,b,c}.

Symbol	Occurrence Probability	Assigned Interval
a	0.3	[0, 0.3)
b	0.3	[0.3, 0.6)
c	0.4	[0.6, 1)

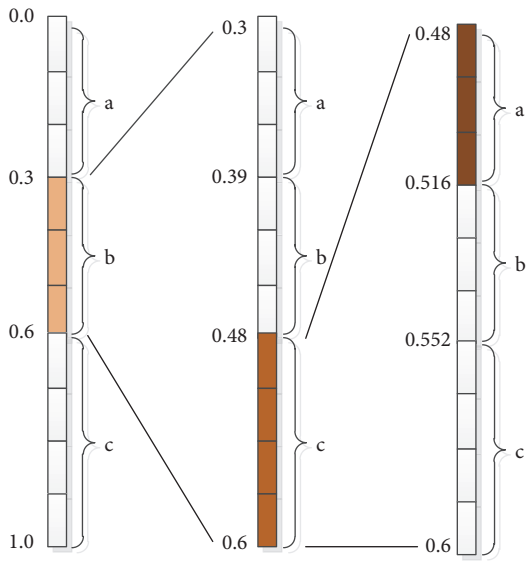


Figure 3: The processes of arithmetic coding with the probability model shown in Table 1.

edge $e_{xy} \in E$ represents a directed edge from vertex v_x to v_y, where $1 \leq y \leq |V|$. Meanwhile $G(V, E)$ satisfies the following properties: (1) p_d is a subgraph of the WSN; (2) for, $v_x, v_y \in V$, v_y is a child of v_x iff $Host(v_y)$ forwards d to $Host(v_x)$; (3) $U \subset V$ is a set of children of $v_x \in V$ iff, for each $v_y \in U$, $Host(v_x)$ receives data packets from $Host(v_y)$.

3.3. Arithmetic Coding. Arithmetic coding [1, 18, 19] is a lossless data compression method that assigns short code words to the symbols with high occurrence probabilities and leaves the longer code words to the symbols with lower occurrence probabilities. The main idea of arithmetic coding is that each symbol of a message is represented by a half-open subinterval of the initial half-open interval [0, 1), and then each subsequent symbol in the message decreases the interval's size by a corresponding subinterval according to the symbol's occurrence probability [19]. Figure 3 shows the process in which a message "bca" is encoded with the probability model in Table 1.

The encoding and decoding operations are as follows:

(1) Encoding: The initial interval is [0, 1). When the first symbol "b" is encoded, it narrows the interval from [0, 1) to [0.3, 0.6), where [0.3, 0.6) is the interval assigned to "b". After the second symbol "c" is encoded, it narrows the [0.3, 0.6) to [0.48, 0.6) according to the interval assigned to "c". Finally, the message "bca" is encoded as the interval [0.48, 0.516).

(2) Decoding: The decoding algorithm utilizes the same probability model in Table 1. With the interval [0.48, 0.516) of being encoded, the first symbol "b" is decoded because the [0.48, 0.516) is a subinterval of the interval [0.3, 0.6) which is the interval assigned to "b". In what follows, the subinterval of "b" is further divided in the same manner to derive the subsequent symbols until the interval of being decoded is equal to the interval of being encoded, namely, [0.48, 0.516) in this example.

The detailed encoding and decoding algorithms of arithmetic coding can be found in [1, 18].

4. Overview of Our Method

In a large-scale WSN, the provenance decoding load at the BS is heavy, which also results in the low efficiency for the data trust evaluation. The layered clustering management provides a good way to manage large-scale WSNs. With a multilayer cluster structure, provenance can be encoded hierarchically, where the final provenance consists of multiple segments. By decoding the provenance segment on the higher layers, the BS obtains the provenance information roughly and then assesses the trustworthiness of the data quickly. Whether to decode the provenance on the next layer depends on the current decoding results; i.e., if we assure that the data have been tempered, the decoding stops immediately.

Compared to the ACP scheme, our scheme has the following characteristics:

(1) Because the layered clustering management is applied, the provenance on each layer can be encoded as an independent segment and the final encoded provenance is composed of a series of segments from different cluster layers. When the BS receives a provenance, it decodes the segment from the highest layer first, and then the BS obtains the provenance information on the most coarse-grained layer, which can be used for a rapid data trust evaluation. Thereafter, the BS continues to decode the provenance step by step. Finally, the BS reconstructs the accurate provenance by combining each segment's decoding result. Therefore, our scheme can encode and decode the provenance in an incremental manner. Such a decoding method raises the efficiencies of the provenance decoding as well as the data trust evaluation.

(2) Compared to the ACP scheme, which uses global probabilities to encode provenance, our scheme encodes the provenance through local probabilities which are only valid in a cluster. By using local probabilities, our scheme not only has a higher compression rate but also can update the probability model partially, which raises the provenance's encoding and decoding efficiencies.

A large WSN can be managed through a multilayer cluster structure, in which the cluster-heads of the same layer are the nodes of an upper layer cluster. As shown in Figure 4, we manage the WSN by different clusters and then form a two-layer cluster managing structure.

In each cluster, the local cumulative probabilities are assigned for each member node. Note that the local cumulative probability is only valid in a cluster, which is quite different from the ACP scheme using global cumulative probabilities [1]. In Figure 4, the highlighted packet path starts from a data source node in a layer-1 cluster, and then

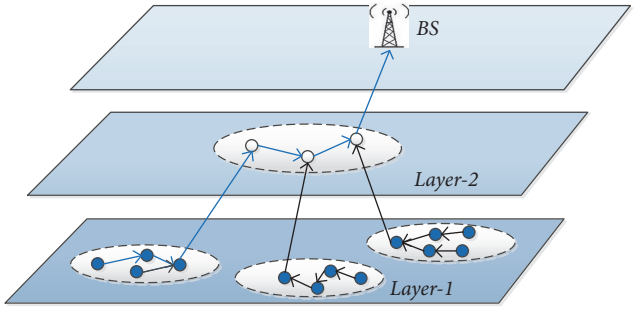


Figure 4: The topology of a two-layer cluster-based WSN.

the packet passes through the nodes in a layer-2 cluster before it reaches the BS. Therefore, the provenance can be encoded into two segments for incremental decoding as shown in Figure 5.

When the BS receives the encoded provenance, it can decode the provenance incrementally and reconstruct the provenance in a way of stepwise refinement. From the first provenance segment decoded results, the BS derives the packet path on layer-2 and the source node of the packet path indicates the cluster from which the packet comes. Then the second segment of the provenance is decoded and yields the packet path on layer-1. Finally, we can recover the entire packet path by combining the two parts decoding. Note that if we know that the packet has passed through some undependable nodes by the layer-2 decoding, the provenance and its packet will be dropped without further decoding, which increases the efficiencies of both provenance decoding and data trust assessment.

Besides, applying a local cumulative probability rather than a global cumulative probability achieves a higher compression rate compared to the ACP scheme. Furthermore, in contrast to the DBNP scheme [11], we use nonoverlapped arithmetic coding which has no false positives.

Because our CBP scheme uses the encoding intervals generated from each node's cumulative probability, the cumulative probabilities are then not changed drastically with respect to the WSN's topology changes. Hence, our scheme is robust to the WSN's topology changes.

5. Cluster-Based Provenance Encoding and Decoding

Before introducing our scheme, we first define the main symbols used in the scheme and algorithms. See Table 2 for details.

At the beginning, the BS trains the network for a certain period to get the number of the times each node n_i appears on the packets' paths as the node's occurrence frequency of_i. During the training process, we let each source node in the network send a certain number of packets to the BS. Upon receiving these packets, the BS computes the occurrence frequencies for each node in the network. Then the local probabilities of each node are computed in their cluster, respectively. How long the training process takes depends on the WSN's scale as well as the accuracy requirement. The more packets used in the training process, the more model accuracy attained, which is also time consuming.

Table 2: The main symbols used in the scheme and algorithms.

Symbols used in scheme/algorithms	Corresponding meanings
$\mathbf{of_i}$	Occurrence frequency of n_i
$\mathbf{op_i}$	Occurrence probability of n_i
$\mathbf{cp_i}$	Cumulative occurrence probability of n_i
$\mathbf{of_{ij}}$	Associated frequency of n_i and n_j
$\mathbf{op_{ij}}$	Associated probability of n_i and n_j
$\mathbf{cp_{ij}}$	Cumulative associated probability of n_i and n_j
$[\mathbf{L_x}, \mathbf{H_x})$	Coding interval
$\mathbf{n_{C_x}}$	Cluster-head ID
buffer	Data cache for the most significant digits

For each node n_i, its occurrence probability op_i can be computed by the following formula: $op_i = of_i/of$, where $of = \sum_{i=1}^{|N|} of_i$. Thereafter, the BS computes the cumulative occurrence probability cp_i for each node n_i by the following equation:

$$\begin{aligned} cp_0 &= 0, \\ cp_i &= cp_{i-1} + op_i. \end{aligned} \tag{2}$$

In what follows, the BS calculates the occurrence frequency with which n_i appears next to n_j as the associated frequency of_{ij}.

Because the number of times a node appears on all the packet paths is equal to the number of the packets that the other nodes receive from it, the total associated frequency of node n_j is then equal to its occurrence frequency of_j. Hence, the associated probability $op_{ij} = of_{ij}/of_j$. At the BS, the cumulative associated probability cp_{ij} for each node n_i is thus derived as

$$\begin{aligned} cp_{0j} &= 0, \\ cp_{ij} &= cp_{(i-1)j} + op_{ij}. \end{aligned} \tag{3}$$

Once the nodes of the WSNs we considered in our scheme are deployed, they stay stationary. In our scheme, we hypothesize that topology changes of the WSNs are slow and infrequent. This kind of slow and infrequent topology changes cannot drastically modify the occurrence frequencies and the probabilities assigned to the nodes. To keep the occurrence frequencies and the probabilities as accurate as possible, we update the probability model periodically or on request by the BS.

5.1. Simple Provenances Encoding and Decoding. Along a packet transmission path, in our simple provenance encoding algorithm (see Algorithm 1), the initial coding interval at the data source is $[L_0, H_0) = [0, 1)$. In what follows the

Input: $[L_{x_1-1}, H_{x_1-1})$
Output: $n_{C_{x_1}}[L_{x_1}, H_{x_1}), buffer$
$low = L_{x_1-1}$, $high = H_{x_1-1}$
$range = high - low$
$buffer = \emptyset$
IF n_i is a source node or a cluster source node **THEN**
  $high = low + cp_i \times range$
  $low = low + (cp_i - op_i) \times range$
ELSE IF n_i is a forwarder node or a cluster forwarder node receiving packet from n_j **THEN**
  $high = low + cp_{ij} \times range$
  $low = low + (cp_{ij} - op_{ij}) \times range$
ELSE IF n_i is a cluster-head **THEN**
  $high = high$
  $low = low$
  $n_{C_{x_1}} = n_i$
END IF
WHILE MSD(low)=MSD($high$) **DO**
    /∗MSD(x) returns the most significant digit of x∗/
  $buffer = buffer \cup MSD(low)$
  $low = ShiftL(low, 1)$
  $high = ShiftL(high, 1)$
    /∗ShiftL(x,y) function returns x shifted by y digits in the left∗/
END WHILE
$L_{x_1} = low$
$H_{x_1} = high$

Algorithm 1: Simple provenance encoding.

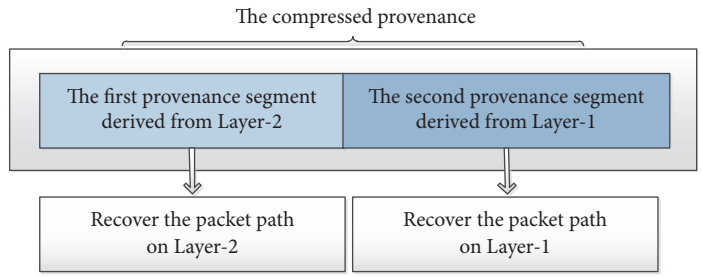


Figure 5: The encoded provenance with two segments.

interval $[L_{x-1}, H_{x-1})$ is used to denote the provenance at the (x-1)th node. When the provenances come from different clusters, the ID of the cluster-head is attached to distinguish the provenances that may share the same coding interval.

A cluster-head of the first layer may play the role of a data source, forwarder, or aggregator node on the second layer. The time complexity of the compression at a cluster-head is the same as that of a data source node, forwarder, or aggregator node, where the space complexity which depends on the layer where the cluster-head is located is doubled or even more.

(1) If node n_i is the data source with the occurrence probability op_i and the cumulative probability cp_i, the interval $[L_1, H_1)$ is then encoded as follows:

$$[L_0, H_0) = [0, 1);$$
$$H_1 = L_0 + cp_i \times (H_0 - L_0);$$
$$L_1 = L_0 + (cp_i - op_i) \times (H_0 - L_0). \quad (4)$$

(2) If node n_i is a forwarder node which receives an interval $[L_{x-1}, H_{x-1})$ from the (x–1)th node n_j, the provenance $[L_x, H_x)$ is then encoded as follows:

$$H_x = L_{x-1} + cp_{ij}(H_{x-1} - L_{x-1});$$
$$L_x = L_{x-1} + (cp_{ij} - op_{ij}) \times (H_{x-1} - L_{x-1}). \quad (5)$$

Although real numbers are used in our algorithms to represent the values of the probabilities and the intervals, at each sensor node the real numbers are replaced by integers to fit for the limited computational ability. Therefore, to meet the demands for the increasing precision as well as avoid transmitting duplicated data, when the most significant digits of the two numbers that define the interval $[low, high)$ are

Input: $n_{C_{x_i}}[L_f, H_f), buffer$
Output: Provenance P
$low = L_f$
$high = H_f$
IF $buffer = \emptyset$**THEN**
$code = \dfrac{(low + high)}{2}$
ELSE
$code = ShiftR(buffer + (low + high)/2, |buffer|)$
/∗ShiftR(x,y) function returns x shifted by y digits in the right∗/
END IF
/∗If code is equals to the one we retrieved. ∗/
/∗Then we retrieve the path which includes $n_{C_{x_i}}$ ∗/
FOR i**=1 to |N| DO**
IF $(cp_i - op_i) \leq code < cp_i$**THEN**
$source = n_i$
BREAK
END IF
END FOR
$dc = source$/∗source decoded∗/
$P = P \cup dc$
$R_{dc} = op_{(dc)}$/∗probability range of decoded∗/
$L_{dc} = cp_{(dc)} - op_{(dc)}$/∗lower end of probability range∗/
WHILE $dc \neq$the node from which BS received packet do
/∗remove effect of decoded node∗/
$code = \dfrac{code - L_{dc}}{R_{dc}}$
FOR i**=1 to|N| DO**
IF $(cp_{i(dc)} - op_{i(dc)} \leq code < cp_{i(dc)})$ **THEN**
$P = P \cup n_i$
$L_{dc} = cp_{i(dc)} - op_{i(dc)}$
$R_{dc} = op_{i(dc)}$
$dc = n_i$
BREAK
END IF
END FOR
END WHILE

Algorithm 2: Simple provenance decoding.

identical, the most significant digit will be shifted out and stored in a buffer. For example, the interval $[0.21, 0.26)$ is represented as $[0.1, 0.6)$ $\{2\}$, where $\{2\}$ is the buffer.

Upon receiving a packet, the BS recovers the provenance through the encoded provenance $n_{C_{x_i}}[L_f, H_f)$; refer to Algorithm 2. The middle point number of the L_f and the H_f is selected using (6) as the flag code to locate the data source node's interval and the data source node is then retrieved. Thereafter, the data source node's effect is removed from the interval $[L_f, H_f)$ and the next node on the packet path will be retrieved through the new flag code by (7).

$$code = \frac{\left(L_f + H_f\right)}{2}, \tag{6}$$

$$code_x = \frac{code_{x-1} - \left(cp_{x-1} - op_{x-1}\right)}{op_{x-1}}, \tag{7}$$

where cp_{x-1} and op_{x-1} denote the cumulative occurrence probability and the occurrence probability of the node being decoded, respectively. Furthermore, the cluster-head's ID $n_{C_{x_i}}$ is used to exclude the nodes that do not belong to such a cluster.

5.2. Aggregated Provenances Encoding and Decoding. Without loss of generality, at an aggregator node assume that there are two provenance intervals $[L_{b_1}, H_{b_1})$, $[L_{b_2}, H_{b_2})$. The aggregator node encodes its ID into both the $[L_{b_1}, H_{b_1})$ and the $[L_{b_2}, H_{b_2})$, which yields the new intervals $[L'_{b_1}, H'_{b_1})$ and $[L'_{b_2}, H'_{b_2})$, respectively. The aggregator node compares the lengths of the two intervals $[L'_{b_1}, H'_{b_1})$ and $[L'_{b_2}, H'_{b_2})$ and then sets the longer one as the first interval which indicates that it is the active interval. Thereafter, the aggregator node randomly chooses a real number r_b in the shorter interval and sends r_b with the active interval to the next hop. Finally the aggregated provenance has the form $n_{c_{b_a}}[L'_{b_a}, H'_{b_a}), n_{c_{b_1}} r_{b_1}, \ldots, n_{c_{b_{\lambda-1}}} r_{b_{\lambda-1}}$; refer to Algorithm 3.

Input: $n_{C_{b_1}}[L_{b_1}, H_{b_1}), n_{C_{b_2}}[L_{b_2}, H_{b_2}), \ldots, n_{C_{b_\lambda}}[L_{b_\lambda}, H_{b_\lambda})$
Output: $n_{C_{b_a}}[L'_{b_a}, H'_{b_a}), n_{C_{b_1}} r_{b_1}, \ldots, n_{C_{b_{\lambda-1}}} r_{b_{\lambda-1}}$
FOR x=1 to λ **DO**
  n_i encodes itself with$[L_{b_x}, H_{b_x})$using Algorithm 1 and results$[L'_{b_x}, H'_{b_x})$
  IF(x=1) **THEN**
    $MaxL = L'_{b_1}$
  $MaxH = H'_{b_1}$
  END IF
  IF(x>1) THEN
    IF $(H'_{b_x} - L'_{b_x} \geq MaxH - MaxL)$**THEN**
      n_i chooses a real number $r_{b_x} \in [MaxL, MaxH)$
      $MaxH = H'_{b_x}$
  $MaxL = L'_{b_x}$
    ELSE
      $n_i$chooses a real number $r_{b_x} \in [L'_{b_x}, H'_{b_x})$
    END IF
  END IF
END FOR
$L'_{b_a} = MaxL$
$H'_{b_a} = MaxH$
prepend $r_{b_1}, r_{b_2}, \ldots, r_{b_{\lambda-1}}$ to $[L'_{b_a}, H'_{b_a})$

Algorithm 3: Aggregated provenance encoding.

Input: $n_{C_{b_a}}[L'_{b_a}, H'_{b_a}), n_{C_{b_1}} r_{b_1}, \ldots, n_{C_{b_{\lambda-1}}} r_{b_{\lambda-1}}$
Output: Aggregated provenance AP
P=decode $[L'_{b_a}, H'_{b_a})$ using Algorithm 2
$AP = AP \cup P$
FOR x=1 to $\lambda - 1$ **DO**
  Let code=r_{b_x}
  P=*decode code* using *A*lgorithm 2 until it reaches a node on the path retrieved from $[L'_{b_a}, H'_{b_a})$
  $AP = AP \cup P$
END FOR

Algorithm 4: Aggregated provenance decoding.

The BS decodes an aggregated provenance as a series of simple provenances which consists of the aggregated provenances (see Algorithm 4), where the simple provenances are packet path sections: (1) from a data source to an aggregator node, (2) from an aggregator node to another aggregator node, (3) from an aggregator node to the BS.

6. Cases Study

In this section, two cases are provided. Without loss of generality, the WSN is organized in two levels in these cases. Our provenance scheme uses hierarchical arithmetic coding to compress provenance within and among the clusters. When the packets pass through the cluster-heads, the provenance will be encoded at a higher level. Figure 6 shows the topology of a WSN composed of four clusters in layer-1. The four cluster-heads, namely, n_5, n_{10}, n_{15}, and n_{20}, collect packets within their clusters and then transmit the packets to the BS.

In Figure 6, the initial probabilities for the member nodes and the cluster-heads were assigned evenly; refer to Tables 3 and 4.

For simplicity, we take a sample provenance and an aggregated provenance that were generated in cluster A as the examples to illustrate how our provenance encoding scheme works.

In Figure 7, the values assigned to the directed edge represent the associated probabilities between the nodes. Based on the data from Tables 3 and 4, the cumulative associated probabilities assigned to the nodes and the cluster-heads were derived by (3).

6.1. Sample Provenance. Assume that the simple provenance to be encoded is $\langle n_8, n_9, n_7, n_{10}, n_{20}, BS\rangle$. As the occurrence probability's range of the data source n_8 is $[0.5, 0.75)$, n_8 then sends the provenance as $\{[0.5, 0.75), \{\emptyset\}\}$ to its next node n_9, where $\{\emptyset\}$ denotes that the buffer is now empty.

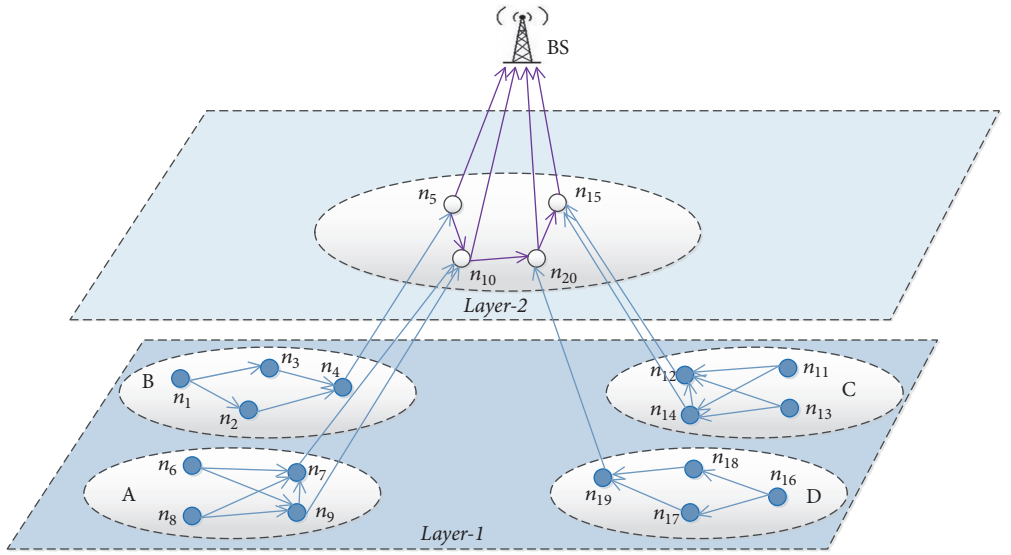


Figure 6: The topology of a WSN which consists of four clusters and a BS.

Table 3: Occurrence probabilities and cumulative occurrence probabilities for the member nodes.

Nodes	op_i	cp_i	Range
n_1, n_6, n_{11}, n_{16}	0. 25	0. 25	[0, 0.25)
n_2, n_7, n_{12}, n_{17}	0. 25	0. 5	[0.25, 0.5)
n_3, n_8, n_{13}, n_{18}	0. 25	0. 75	[0.5, 0.75)
n_4, n_9, n_{14}, n_{19}	0. 25	1	[0.75, 1)

Table 4: Occurrence probabilities and cumulative occurrence probabilities for the cluster-heads.

Nodes	op_i	cp_i	Range
n_5	0. 25	0. 25	[0, 0.25)
n_{10}	0. 25	0. 5	[0.25, 0.5)
n_{15}	0. 25	0. 75	[0.5, 0.75)
n_{20}	0. 25	1	[0.75, 1)

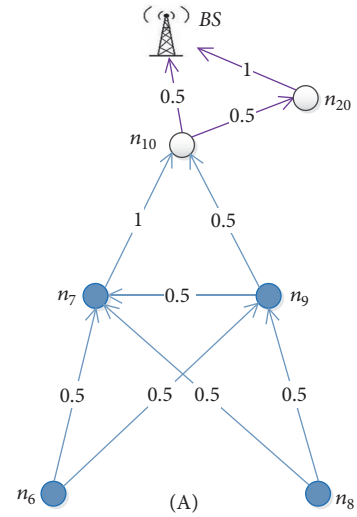


Figure 7: The topology of the cluster A in Figure 6.

Node n_9 then derives the new coding interval through its associated probability and cumulative associated probability with respect to n_8. According to Algorithm 1, the new coding interval at n_9 is equal to $[0.625, 0.75)$. Similarly, the new interval at n_7 is encoded as $[0.625, 0.6875)$ and then the provenance is updated as $\{[0.25, 0.875), \{6\}\}$ when being sent to n_{10}. Because n_{10} is a cluster-head, it simply adds its cluster ID to the provenance and updates the provenance as $\{\{n_{10}\}, [0.25, 0.875), \{6\}\}$.

Because the occurrence probability's range of the data source n_{10} is $[0.25, 0.5)$, n_{10} sends the provenance as $\{\{\emptyset\}, [0.25, 0.5), \{\emptyset\}\}$ with $\{\{n_{10}\}, [0.25, 0.875), \{6\}\}$ to its next node n_{20}. Because n_{20} is a cluster-head, it only updates the provenance at the higher level, i.e., the cluster-head level, and then sends the updated provenance in the form of $\{\{\{\emptyset\}, [0.25, 0.375), \{\emptyset\}\}, \{\{n_{10}\}, [0.25, 0.875), \{6\}\}\}$ to the BS. Table 5 shows the encoding processes of the simple provenance $\langle n_8, n_9, n_7, n_{10}, n_{20}, BS\rangle$ at each node along the packet path.

When the packet arrives, the BS parses the attached provenance. Because there are two cluster levels, the provenance has two sections accordingly. For the first section, namely, $\{\{\{\emptyset\}, [0.25, 0.375), \{\emptyset\}\}$, the BS filtrates the interval $[0.25, 0.375)$ and the buffer and then derives that $code_1 = 0.3125$ which belongs to the cumulative probability of n_{10}. The data source node of the first section is thus n_{10}. Thereafter, the BS removes the effect of n_{10} from the $code_1$ and then derives that $code_2$ is equal to 0.26. Because the associated probability of n_{20} with respect to n_{10} is $[0, 0.5)$, n_{20} is yielded from the decoding. The BS stops decoding the first section because the packet was received from n_{20} and yields the decoding result of layer-2 as $\langle n_{10}, n_{20}, BS\rangle$. With the second section, the BS filtrates that the cluster-head ID is n_{10}, the

Table 5: Simple provenance encoding at each node along the packet path.

Pack Path	Encoded Node	Cluster-head ID	Coding Interval	Buffer	Provenance
n_8	n_8	∅	[0.5, 0.75)	∅	{{∅}, [0.5, 0.75), {∅}}
n_8n_9	n_9	∅	[0.625, 0.75)	∅	{{∅}, [0.625, 0.75) , {∅ }}
$n_8n_9n_7$	n_7	∅	[0.625, 0.6875)	6	{{∅}, [0.25, 0.875), {6}}
$n_8n_9n_7n_{10}$		n_{10}			{{n_{10}}, [0.25, 0.875), {6}}
n_{10}	n_{10}	∅	[0.25, 0.5)	∅	{{{∅}, [0.25, 0.5), {∅}}, {{n_{10}}, [0.25, 0.875) {6}}}
$n_{10}n_{20}$	n_{20}	∅	[0.25, 0.375)	∅	{{{∅}, [0.25, 0.375), {∅}}, {{n_{10}}, [0.25, 0.875), {6}}}

Table 6: Simple provenance decoding at the BS.

Parts of Provenance	Code	Cluster-head ID	Buffer	Decoded Node	Pack Path
First Part: {{∅}, [0.25, 0.5), {∅}}	0.3125	∅	∅	n_{10}	n_{10}
	0.26	∅	∅	n_{20}	$n_{10}n_{20}$
Second Part: {{n_{10}}, [0.25, 0.875), {6}}	0.65625	n_{10}	6	n_8	n_8
	0.625	n_{10}	∅	n_9	n_8n_9
	0.25	n_{10}	∅	n_7	$n_8n_9n_7$

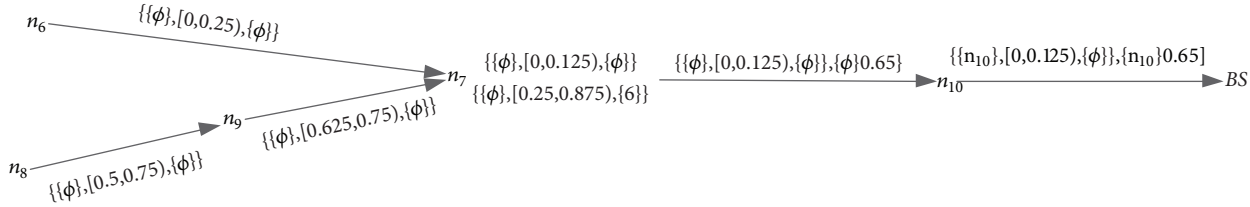


Figure 8: Aggregated provenance encoding at each node along the packet path.

coding interval is [0.25, 0.875), and the buffer is {6}. The BS then derives that $code_1 = 0.65625$ which belongs to the occurrence probabilities of n_3, n_8, n_{13}, and n_{18}. Note that the cluster-head's ID is n_{10}; the data source is thus n_8. Therefore, the BS removes the n_8's effect from the $code_1$ and derives that $code_2$ is equal to 0.625. Because the associated probability's range of n_9 with respect to n_8 is [0.5, 1), n_9 is yielded from the decoding. The BS stops decoding until the last node n_7 is yielded. The decoding result is then $\langle n_8, n_9, n_7, n_{10}\rangle$. Finally, the BS combines the incremental decoding results and then knows the provenance is $\langle n_8, n_9, n_7, n_{10}, n_{20}, BS\rangle$.

In order to explain the decoding process more clearly, Table 6 shows the steps of the simple provenance decoding $\langle n_8, n_9, n_7, n_{10}, n_{20}, BS\rangle$ at the BS. The BS decodes the first part of the provenance and reconstructs the data packet path on layer-2 which is composed of cluster-heads for the clusters on layer-1 (see Figure 6). According to the layer-2 decoding result, the BS knows from which cluster this data packet comes. If the BS finds any undependable information such as tampered nodes or faked packet path in the layer-2 decoding, it will discard the packet and stop decoding, which raises the provenance decoding efficiency as well as the data trust assessment efficiency.

6.2. Aggregated Provenance. In Figure 7, now we consider the encoding and decoding of an aggregated provenance $(((n_6)((n_8), n_9), n_7), n_{10})$. According to Algorithm 3, at the beginning, the data source n_6 sends $[0, 0.25)$ (referred to as $[L_1, H_1)$) with a packet p_1 to its next node n_7. At the moment, the data source n_8 also sends [0.5, 0.75) as the provenance with packet p_2 to its next node n_9, where the new coding interval [0.625, 0.75) (referred to as $[L_2, H_2)$) is derived.

At the first aggregator node n_7, the two packets p_1 and p_2 are met. Because there are two coding intervals, i.e., $[L_1, H_1)$ and $[L_2, H_2)$, n_7 encodes its ID into both of them and makes the new intervals as $[L_1', H_1')$=[0, 0.125) and $[L_2', H_2')$= [0.625, 0.6875), respectively. Thereafter, n_7 compares the lengths of $[L_1', H_1')$ and $[L_2', H_2')$ and chooses a random real number r_b belonging to the shorter interval. Subsequently, n_7 updates the aggregated provenance to the form of {{{∅}, [0, 0.125), {∅}}, {∅}0.65}, where $r_b = 0.65$, and sends it with the new aggregated packet to n_{10}. Because n_{10} is a cluster-head, it simply adds its ID to the provenance and updates the provenance to the form of {{{n_{10}}, [0, 0.125), {∅}}, {n_{10}}0.65}. Finally, the provenance of the aggregated packet is {{{n_{10}}, [0, 0.125), {∅}}, {n_{10}}0.65} at the BS. Figure 8 shows the encoding of the aggregated provenance $(((n_6)((n_8), n_9), n_7), n_{10})$ at each node along the packet path.

When the packet arrives, first, the BS decodes the foremost part {{n_{10}}, [0, 0.125), {∅}} of the provenance, which yields $\langle n_6, n_7, n_{10}, BS\rangle$. Second, the BS decodes {n_{10}} 0.65 in the provenance according to Algorithm 4, because $code = 0.65$ which belongs to the occurrence probability ranges of n_3,

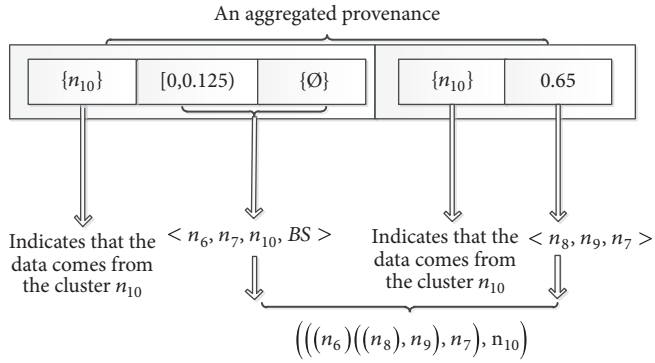


Figure 9: The decoding processes for an aggregated provenance at the BS.

n_8, n_{13}, and n_{18}. Note that the cluster-head's ID is n_{10}; the data source is thus decoded as n_8. In what follows, the BS removes the n_8's effect from the *code* and yields $code_1$ as 0.6. Because the associated probability of n_9 with respect to n_8 is $[0.5, 1)$, n_9 is then yielded from the decoding. The decoding process stops until n_7 is yielded. As n_7 is a node on the path decoded from the foremost part of the provenance, the decoding process ends with the result $(((n_6)((n_8), n_9), n_7), n_{10})$. Figure 9 shows the decoding processes of the aggregated provenance $(((n_6)((n_8), n_9), n_7), n_{10})$ at the BS.

7. Performance Analysis

In this section, we theoretically analyze the performance of our CBP scheme with respect to the compressed provenance length and the optimal clustering size.

7.1. Entropy of Simple Provenance. In our approach, according to Shannon's theory, it takes $H_j = \lceil -\log op_j \rceil$ bits to represent a source node n_j and $H_{ij} = \lceil -\log op_{ij} \rceil$ bits to represent a forwarder node n_i, where n_j is n_i's children node. Hence, we can derive the entropy of the provenance from the number of the bits required to represent an interval r; i.e., $H_r = \lceil -\log r \rceil$. Note that the final interval's size is equal to the product of the occurrence probabilities of the source node and the corresponding associated probabilities of the forwarder nodes; i.e., $r = op_j \times \prod_{l_{ji} \in E} op_{ji}$. H_r is thus calculated as

$$\begin{aligned} H_r &= \left\lceil -\log\left(op_j \times \prod_{l_{ji} \in E} op_{ji} \right) \right\rceil \\ &= \left\lceil -\log op_j - \sum_{l_{ji} \in E} op_{ji} \right\rceil. \end{aligned} \tag{8}$$

7.2. Entropy of Aggregated Provenance. Assume that the number of aggregator nodes is R and the number of the branches is $b_r (b_r > 1)$ at an aggregator node $n_r (1 \le r \le R)$. Therefore, $-\sum_{r=1}^{R} \sum_{q=1}^{b_r - 1} \lceil \log op_{n_r j} \rceil$ bits are needed to represent all the aggregator nodes, where $op_{n_r j}$ is the associated probability of node n_r with respect to its child node n_j.

7.3. The Optimal Cluster Size. Assume that the number of the nodes in a WSN is N and the number of the cluster-heads is M, where $M < N$ and M and N are positive integers. For the given M and N, if the average provenance size has the minimum value compared to other values assigned to M and N, M is then the optimal clustering size for the WSN.

Claim. With our cluster-based arithmetic coding provenance scheme (CBP), the optimal cluster size is $M = \sqrt{N}$.

Justification. Assume that the occurrence probability op_i of each node is evenly distributed. All member nodes' occurrence probabilities are then evenly distributed too; i.e., $op_1 = op_2 = \cdots = op_i = \cdots = op_n$. Therefore, each node's cumulative occurrence probability is $cp_0 = 0, cp_i = cp_{i-1} + op_i$, where $op = \sum_{i=1}^{|N|} op_i = 1$. According to the entropies of the simple provenance and the aggregated provenance, when $M = \sqrt{N}$, the average entropy of the provenances has the minimum value, which indicates that the average provenance size reaches its minimum value.

8. Simulations

We used TinyOS 2.1.2 TOSSIM and PowerTOSSIMz for the implementation of our approach to evaluate the performance of our approach. In the simulations, a WSN consisting of 101 nodes with IDs 0 through 100 is deployed, where the node with ID 0 is set as the BS. The network diameters vary from 2 to 14 hops. The duration of each data collection round is set to 2 seconds. We define the topology of a WSN by offering one-hop neighboring information between nodes. According to the topology of such a WSN, we randomly select some nodes either as the data source or aggregator nodes in TinyOS, and then we take a two-layer cluster to verify our scheme. First, all the 100 nodes are managed into different clusters. The cluster-head collects data from its member nodes and then sends the data toward the BS through multihops. At the beginning, the BS computes the occurrence probability, the cumulative occurrence probability for each node, and the associated

probability and the cumulative associated probability for each node pair in the WSN.

For the purpose of simplicity, real numbers are used in this paper to represent the probabilities and define the coding intervals. In the simulations and the experiments, because of the computational limitations, integer arithmetic coding is applied because of the computational limitations at sensor nodes. By using a 2-byte integers, the initial interval $[0, 1)$ is represented as $[0, 2^{16} - 1)$.

In addition, we directly compare our approach with the ACP scheme [1] which has close relationship with our CBP scheme, where the ACP scheme assigns each node a global cumulative probability and then uses the cumulative conditional probabilities for each connected node pair to generate the provenance through arithmetic coding.

8.1. Performance Metrics. The following performance metrics are used in the paper:

(i) Average provenance sizes (APS): When the BS receives a packet, it filtrates the $C_f[L_f, H_f)$ and *buffer* from the provenance. Assume that S_{C_f}, S_{L_f}, *and* S_{H_f} denote the sizes of C_f, $[L_f, H_f)$, and *buffer*, respectively. The size of the provenance is then equal to $(S_{C_f} + S_{L_f} + S_{H_f} + S_{buffer})$. In the simulations, we use 8 bits to denote C_f and 16 bits to denote L_f and H_f, respectively.

Assume that there are m packets $p_1, p_2, p_3, \ldots, p_m$; APS is then defined as follows:

$$APS = \frac{\sum_{i=1}^{m} PS_i}{m}, \tag{9}$$

where PS_i represents the provenance size of the packet p_i.

(ii) Total energy consumption (TEC): Suppose that there are $n_1, n_2, n_3, \ldots, n_m$ nodes in a WSN. TEC is defined as follows:

$$TEC = \sum_{i=1}^{m} EC_i, \tag{10}$$

where EC_i denotes the energy consumed by node n_i and m represents the total number of the nodes in the WSN.

8.2. Simulation Results. The identical simulation environment is used to simulate the ACP scheme and our CBP scheme. Figure 10 shows the relationship between the number of the clusters and the average provenance size. The curve shows that when the number of clusters is equal to the number of the nodes in each cluster, the average provenance size reaches its minimum value. When all nodes are assigned in one cluster or each node forms an individual cluster, the average provenance size reaches its maximum value. Such simulation results also verified the conclusion in Section 7.3.

From the trends of the two curves in Figure 11(a), it can be concluded that our scheme can achieve a higher compression rate than that of the ACP scheme. With the number of hops increasing, our scheme outperforms the ACP scheme with respect to the average provenance size. Figure 11(b) shows the total energy consumption of the ACP and the CBP schemes when 100 packets are transmitted. Compared to Figures 11(a) and 11(b), we can find that curves share the same trend. The more messages are transmitted through wireless signals, the more energy is consumed. As to the CBP and the ACP schemes, the total energy consumption of the CBP scheme at each node is lower than that of ACP scheme (see Figure 11(b)). From Figures 11(a) and 11(b), it can be concluded that the CBP scheme has better performance with respect to total energy consumption (TEC) and average provenance size (APS).

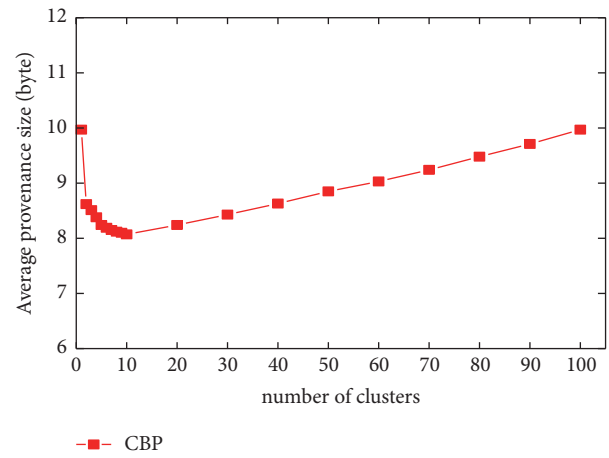


Figure 10: The relationship between the number of the clusters and the average provenance size in a WSN.

Figure 12(a) shows the average provenance size for the ACP and the CBP schemes with respect to the number of packets sent by data source nodes. In the simulations, there are 100 nodes; 30 of them are data source nodes and 5 of them are aggregator nodes. Figure 12(b) shows the average provenance size for the ACP and the CBP schemes with respect to the number of packets sent by data source nodes. In the simulations, there are 100 nodes; 30 of them are data source nodes and 10 of them are aggregator nodes. It can be concluded that as the number of packets sent by data source nodes increases, the average provenance sizes of the ACP and the CBP schemes increase too, but the average provenance size of the CBP scheme is less than that of the ACP scheme.

Furthermore, in [1], it has been shown that the ACP scheme is better than the Bloom filter based provenance scheme (BFP) [9], the Generic secure provenance scheme (SPS) [20], and the MAC based provenance scheme (MP) [8]. The comparisons between the CBP scheme and the ACP scheme indirectly show that the CBP scheme outperforms the BFP, the SPS, and the MP schemes with respect to both the average provenance size and the energy consumption. As to the dictionary-based scheme (DP) [10], it has been compared with the ACP and the DBNP schemes in [11] and the results show that the DP scheme is sensitive to the WSN's topology change, whereas the CBP scheme keeps stable when the WSN's topology changes.

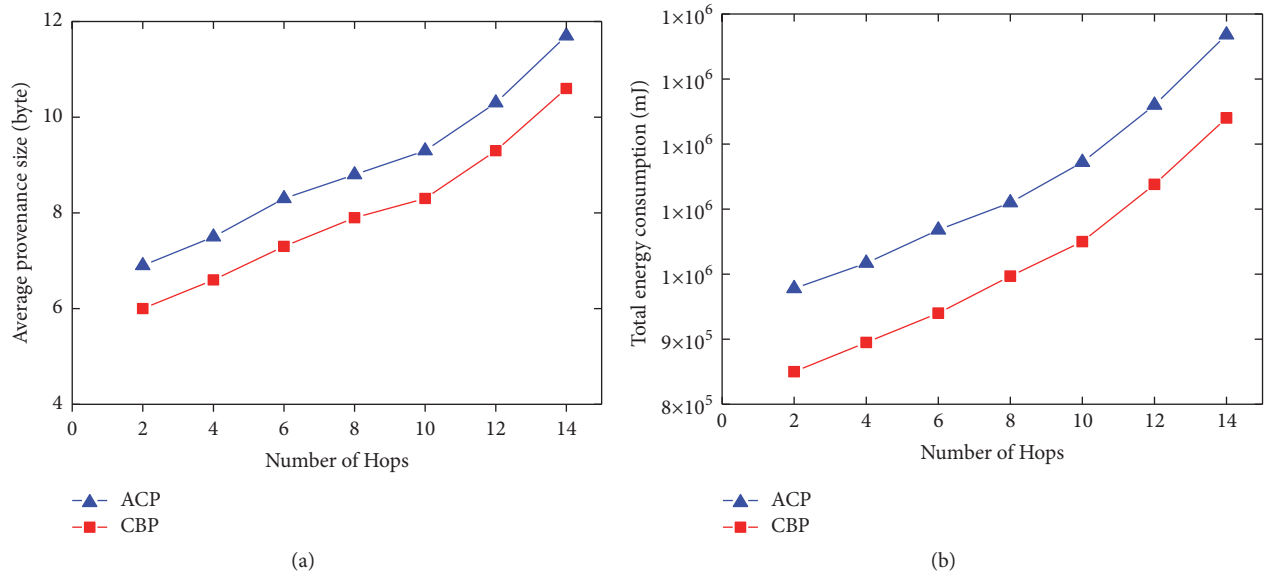


FIGURE 11: (a) The relationship between the number of packet transmission hops and the average provenance size for the ACP and the CBP schemes. (b) The total energy consumption of the CBP and the ACP schemes with respect to the number of packet transmission hops.

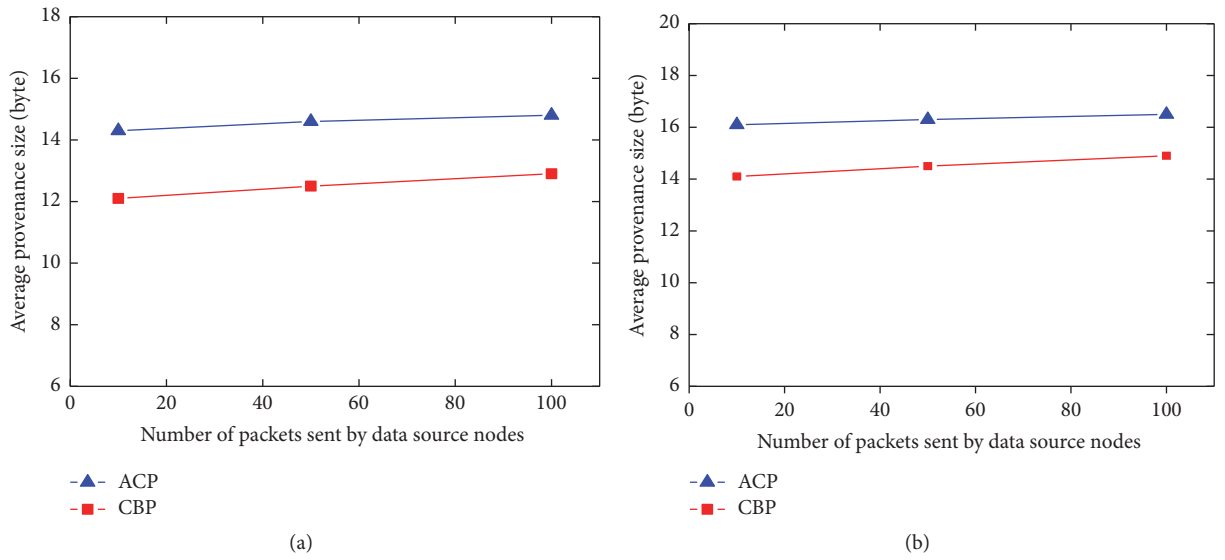
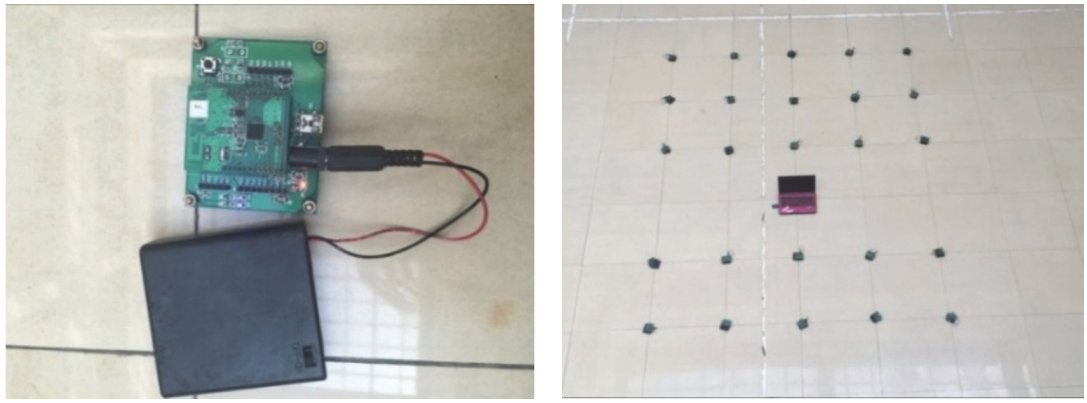


FIGURE 12: (a) With 30 data source nodes and 5 aggregator nodes, the average provenance size for the ACP and the CBP schemes with respect to the number of sent packets. (b) With 30 data source nodes and 10 aggregator nodes, the average provenance size for the ACP and the CBP schemes with respect to the number of sent packets.

9. Experiments

To further evaluate the performance of our scheme, we deployed the ACP and the CBP schemes in a test-bed which included 26 sensor motes. In the experiments, the performances were evaluated by the average provenance size only.

9.1. Experimental Setup. We used ZigBee sensor motes to port the implementation. The ZigBee mote used by us has a CC2530 microcontroller, 2.4GHz radio, 8KB RAM, and 256KB external flash for data logging (see Figure 13(a)). ZigBee motes are placed in an indoor environment in a grid topology with network areas of $20\times10m^2$ (see Figure 13(b)). In addition, the equipment also includes the IAR compiler environment, the serial port view tool. The mote connecting to the laptop computer through a USB port is set as the BS. The I/O functions of the TinyOS simulation codes are modified and then ported to the ZigBee mote. The experiments are performed with more than 100 packets' transmission in the WSN.



(a) (b)

Figure 13: (a) Zigbee sensor motes. (b) The test-bed.

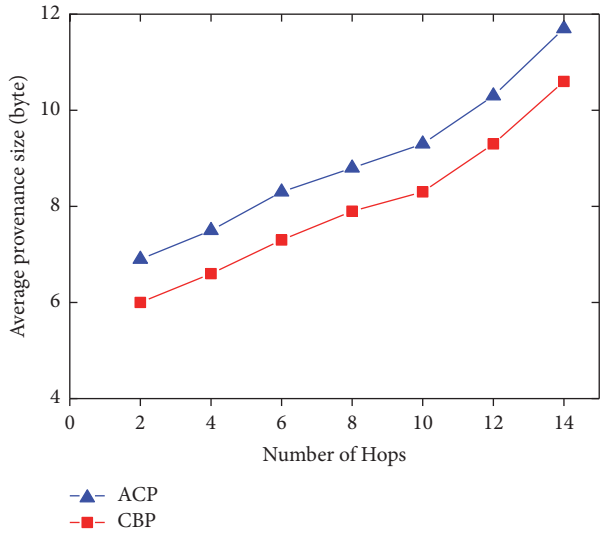


Figure 14: The average provenance size for the ACP and the CBP schemes with respect to the number of packet transmission hops.

9.2. Experimental Results. In the test-bed experiments, Figure 14 shows the average provenance sizes for the ACP and the CBP schemes with respect to the number of packet transmission hops. The data show that the curves have similar trends compared with the simulation results in Figure 11(a), which also shows that the CBP scheme under the optimal clustering size can achieve a much higher average provenance compression rate than that of the ACP scheme.

10. Conclusions

In a large-scale WSN, in order to mitigate the data provenances size's rapid expansion, we propose a cluster-based arithmetic coding provenance scheme (CBP). Compared to the known provenance schemes based on arithmetic coding, the CBP scheme not only yields a higher provenance compression rate, but also can encode and decode the provenance incrementally at the BS, which increases the efficiencies of both the provenance decoding and the data trust assessment. Furthermore, the optimal cluster size for the CBP scheme is formally derived. Both the simulation and the experimental results show the effectiveness and the efficiency of our CBP scheme in the paper.

Data Availability

All data generated and analyzed during this study are from our simulations and experiments, rather than public repositories online. The simulator we used is TinyOS 2.1.2 TOSSIM. The energy consumption is measured through PowerTOSSIMz. We submitted our simulation code of the ACP scheme and our scheme when we submitted our article. The experiment results are generated by porting the simulation code to the ZigBee sensor nodes. The major changes in the simulation code were related to the I/O functions.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61672269 and the Jiangsu Provincial Science & Technology Project under Grant BA2015161.

Supplementary Materials

The supplementary material file we submitted includes the simulation code of the ACP scheme and our scheme. Before running the code, TinyOS 2.1.2 should be installed in the Linux environment (the version of the Linux we use is Ubuntu) and some related configurations should be made. Almost all the configurations can be found on the Internet. We just mention some main code execution

instructions. After entering the specified directory, execute "make micaz" and "make micaz sim" instructions. After that, execute "python test.py". The simulation results will be stored in "log.txt" file. Execute "python postprocessZ.py --powercurses Energy.txt > EnergyPowerCurses.txt" when the PowerTOSSIMz is successfully configurated. The energy consumption results will be stored in "EnergyPowerCurses.txt" file. Before porting the simulation code to the ZigBee sensor nodes, the I/O functions should be changed according to the specific node type. *(Supplementary Materials)*

References

[1] S. R. Hussain, C. Wang, S. Sultana, and E. Bertino, "Secure data provenance compression using arithmetic coding in wireless sensor networks," in *Proceedings of the 2014 IEEE International Performance Computing and Communications Conference (IPCCC)*, pp. 1–10, Austin, TX, USA, December 2014.

[2] H. S. Lim, Y. S. Moon, and E. Bertino, "Provenance-based trustworthiness assessment in sensor networks," in *Proceedings of the 7th International Workshop on Data Management for Sensor Networks*, pp. 2–7, ACM, 2010.

[3] F. Zafar, A. Khan, S. Suhail et al., "Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes," *Journal of Network and Computer Applications*, vol. 94, pp. 50–68, 2017.

[4] W. Zhuo, Q. Fei, A. Narayan, A. Haeberlen, B. Loo, and M. Sherr, "Secure Network Provenance," in *Proceedings of the 23rd ACM Symposium on Operating Systems Principles*, 2011.

[5] W. Zhou, M. Sherr, T. Tao et al., "Efficient querying and maintenance of network provenance at internet-scale," in *Proceedings of the ACM SIGMOD International Conference on Management of data*, pp. 615–626, ACM, 2010.

[6] M. T. Goodrich, "Probabilistic packet marking for large-scale IP traceback," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 15–24, 2008.

[7] S. M. I. Alam and S. Fahmy, "Energy-efficient provenance transmission in large-scale wireless sensor networks," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM '11*, pp. 1–6, IEEE, 2011.

[8] S. Sultana, G. Ghinita, E. Bertino, and M. Shehab, "A lightweight secure scheme for detecting provenance forgery and packet dropattacks in wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 3, pp. 256–269, 2015.

[9] B. Shebaro, S. Sultana, S. R. Gopavaram et al., "Demonstrating a lightweight data provenance for sensor networks," in *Proceedings of the ACM conference on Computer and communications security*, pp. 1022–1024, ACM, 2012.

[10] C. Wang, S. R. Hussain, and E. Bertino, "Dictionary Based Secure Provenance Compression for Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 405–418, 2016.

[11] C. Wang and E. Bertino, "Sensor network provenance compression using dynamic bayesian networks," *ACM Transactions on Sensor Networks*, vol. 13, no. 1, p. 5, 2017.

[12] M. Guazzo, "A general minimum-redundancy source-coding algorithm," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 26, no. 1, pp. 15–25, 1980.

[13] J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[14] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, 1978.

[15] X. Artigas, S. Malinowski, C. Guillemot, and L. Torres, "Overlapped quasi-arithmetic codes for distributed video coding," in *Proceedings of the 14th IEEE International Conference on Image Processing, ICIP 2007*, pp. II9–II12, usa, September 2007.

[16] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the IEEE Societies 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM '03)*, vol. 3, pp. 1713–1723, March-April 2003.

[17] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 131–146, 2002.

[18] P. G. Howard and J. S. Vitter, "Arithmetic coding for data compression," *Proceedings of the IEEE*, vol. 82, no. 6, pp. 857–865, 1994.

[19] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, 1987.

[20] R. Hasan, R. Sion, and M. Winslett, "The case of the fake picasso: preventing history forgery with secure provenance," *FAST*, vol. 9, pp. 1–14, 2009.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5787930, 7 pages
https://doi.org/10.1155/2018/5787930

Research Article

Privacy Protection of IoT Based on Fully Homomorphic Encryption

Wei-Tao Song, Bin Hu, and Xiu-Feng Zhao

Information Science and Technology Institute, Zhengzhou 450001, China

Correspondence should be addressed to Wei-Tao Song; weitaosong@163.com

Received 4 March 2018; Revised 27 April 2018; Accepted 15 May 2018; Published 20 June 2018

Academic Editor: Ximeng Liu

With the rapid development of Internet of Things (IoT), grave questions of privacy protection are raised. This greatly impacts the large-scale applications of IoT. Fully homomorphic encryption (FHE) can provide privacy protection for IoT. But, its efficiency needs to be greatly improved. Nowadays, Gentry's bootstrapping technique is still the only known method of obtaining a "pure" FHE scheme. And it is also the key for the low efficiency of FHE scheme due to the complexity homomorphic decryption. In this paper, the bootstrapping technique of Halevi and Shoup (EUROCRYPT 15) is improved. Firstly, by introducing a definition of "load capacity", we optimize the parameter range for which their bootstrapping technique works. Next we generalize their ciphertext modulus from closing to a power of two to more general situations. This enables the method to be applied in a larger number of situations. Moreover, this paper also shows how to introduce SIMD homomorphic computation techniques into the new method, to improve the efficiency of recryption.

1. Introduction

Nowadays, the IoT is becoming an attractive system paradigm to drive a substantive leap on goods and services through physical, cyber, and social spaces. It covers from traditional equipment to general household equipment, which bring more efficiency and convenience to the users and change current ways of life greatly [1]. See Figure 1.

However, the application of IoT involves mass private information about users, such as healthcare, location, etc. For the users, they want service providers to process the data accurately and efficiently and extract the contained valuable information with keeping user data unknown by others (including themselves). All these problems are difficult to achieve by traditionally encryption schemes. Homomorphic encryption technology is a good choice to solve all these problems [2, 3].

FHE permits a worker to perform arbitrarily complex programs on encrypted data without knowing the secret key [4]. And FHE has been the focus of extensive study [5–13], since the first candidate scheme was introduced by Gentry [14]. But its efficiency needs to be greatly improved.

Since bootstrapping technology is the essential technology to obtain a "pure" FHE at present. Meanwhile, it is also the main bottleneck in any practical implementation due to the complexity homomorphic decryption. It is very meaningful to improve the efficiency of bootstrapping, which mainly refers to fast low-circuit implementation of decryption function. Without loss of generality, the decryption function for LWE- (Learning with Errors-) based FHE can be computed by evaluating some linear operation between ciphertext and secret key, then reducing the result modulo a big odd modulus q and then reducing the result modulo a small modulus p, to get the plaintext m, namely, $m = [[L_c(s)]_q]_p$. For the decryption function, the modular-reduction operation of $[[z]_q]_p$ ($z \in \mathbb{Z}$) affects the depth of decryption circuit most.

The past few years have seen an intensive study of bootstrapping technique. In the original bootstrapping technique of Gentry [14], he put forward an idea of "squash the decryption circuit" to transform modular-reduction operation into summing operation. This got a moderate polynomial $O(\lambda^4)$ runtime. By proposed an amortized bootstrapping method, Brakerski, Gentry, and Vaikuntanathan (BGV) reduced the

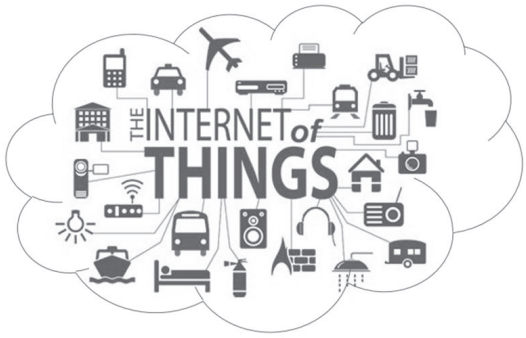


Figure 1: IoT architecture.

runtime to $O(\lambda^2)$ [10]. However, these results applied only to "non-packed" ciphertexts (i.e., ones that encrypt just one bit each).

Gentry, Halevi, and Smart (GHS, PKC2012) reached a major milestone of a bootstrapping algorithm concentrating on the BGV ring-LWE-based scheme (ideal lattice-based FHE) [15]. They proposed a simpler decryption formula. This is done by choosing a prime plaintext modulus p and a ciphertext modulus q close to a power of p. Besides, they utilized packed ciphertexts and Fourier Transform to aid efficiency. To improve the Fourier Transform step of [15], [16] presented a ring/field switching technique. This obtained an asymptotically efficient bootstrapping method for BGV style SHE scheme. Orsini, Pol, and Smart (PKC15) proposed a bootstrapping BGV ciphertexts with a wider choice of p and q, but their decryption formula was not simple as GHS's work. Halevi and Shoup (HS, EUROCRYPT 15) generalized the plaintext modulus p in [17] to more general situations and got a better efficiency by improving the bit-extraction way. This is asymptotically optimal space and time so far.

In another line of work, [18–22] present a bootstrapping technique for the GSW-FHE [13] scheme. They get significant progress in improving the bootstrapping technique on standard lattice-based FHE. And their progress mainly relies on the characteristic that noise in ciphertexts of GSW-FHE grows asymmetrically. Since compared with standard LWE-based FHE schemes, ring-LWE-based FHE schemes always have more efficient homomorphic operations. And among all the ring-LWE-based FHE schemes, BGV ring-LWE-based FHE scheme is optimal (note that GSW-FHE scheme is better than BGV-FHE scheme only in standard LWE-based FHE schemes). Note that, in this paper, the aim is to provide practical FHE scheme for the privacy protection of IoT. Thus, it concentrates on improving the bootstrapping technique of BGV ring-LWE-based FHE schemes in this paper.

The starting point of this paper is the HS's work [17], where decryption procedure consists of a linear algebra step $< c, s >$ and a modular reduction step $[[z]_q]_p$. When $|z| \leq q^2/4 - q$ and $|[z]_q| \leq q/4$, modular reduction step $[[z]_q]_p$ can be converted to simple bit operations. This greatly reduces the circuit depth of modular reduction. When homomorphically performed above simple decryption formula, the deepest part is homomorphic bit-extraction procedure, and its complexity (both time and depth) increases with the most-significant extracted bit. In [17], by adding to ciphertext multiples of q and also multiples of p, they proposed a lower-degree homomorphic bit-extraction procedure. And the bigger the parameter range of z for the simple formula of modular reduction, the better the performance for the improved homomorphic bit-extraction procedure. See [17] for further details.

Contributions. In this paper, we optimize the parameters of bootstrapping algorithm proposed in EUROCRYPT 2015 by Halevi and Shoup. Firstly, by introducing a definition of load capacity, we optimize the parameters range for which their bootstrapping technique works for the first time. Next we generalize their ciphertext modulus q to more general situations. This makes our method applicable to more cases. Moreover, we also show how to introduce SIMD technique into our new method, to improve the efficiency of bootstrapping technique.

Organization. Section 2 presents the notations and some background on the BGV cryptosystem. Section 3 optimizes the parameter range for which bootstrapping technique of Halevi and Shoup works. Next, the ciphertext modulus is generalized from closing to a power of two to more general situations in Section 4. Moreover, it also shows how to introduce SIMD homomorphic computation techniques into the new method to get an efficient bootstrapping method. And in Section 5, an implementation is made of BGV ring-LWE-based scheme based on our efficient bootstrapping method. Finally, Section 6 concludes.

2. Preliminaries

Basic Notations. Set $\mathbb{Z}_q \in (-q/2, q/2] \bigcap \mathbb{Z}$, and the notation $[a]_q$ is referred to as a mod q, with coefficients being reduced into the range $(-q/2, q/2]$. For an integer z (positive or negative), we consider the base-p representation of z and denote its digits by $z\langle 0\rangle_p, z\langle 1\rangle_p, \cdots$.

2.1. Homomorphic Encryption Schemes. Let $\mathbb{M}$ be the message space and $\mathbb{C}$ be the ciphertext space. A homomorphic encryption scheme $HE = \{KeyGen, Enc, Dec, Eval\}$ is as follows:

(i) $KeyGen(1^\lambda)$: output public key pk, secret key sk, and evaluation key evk.

(ii) $Enc_{pk}(\mu)$: output ciphertext $c \in \mathbb{C}$ encrypted by plaintext $\mu \in \mathbb{M}$ with public key pk.

(iii) $Dec_{sk}(c)$: recover the message encrypted in the ciphertext c by secret key sk.

(iv) $Eval_{evk}(f, c_1, \ldots, c_l)$: output ciphertext $c_f \in \mathbb{C}$ which is obtained by applying evaluation key evk and the function $f : \mathbb{M}^l \to \mathbb{M}$ to $c_1, \ldots, c_l$.

Suppose that (sk_1, pk_1) and (sk_2, pk_2) are two key-pairs of scheme HE. Let c be a ciphertext of plaintext μ under pk. Let $\overline{sk_{1i}}$ be a ciphertext of the i-th bit of the first secret key sk_1

under the second public key pk_2. D is a decryption circuit. See Algorithm 1 for the "Bootstrapping" algorithm.

It can be found that $HE.Dec(sk_2, c') = HE.Dec(sk_1, c) = \mu$ only when scheme HE can compactly evaluate its decryption circuit. However, most of the existing schemes do not satisfy this condition naturally. It needs some extra operations, such as "squashing the decryption circuit", which cause the low efficiency of FHE. Thus, it is very meaningful for lower-depth circuit implementation of decryption function.

3. Analysis of HS Recryption Procedure

We start by introducing the HS recryption procedure [17] on that how to homomorphically compute the modular-reduction operation in a lower-depth circuit. The specifics are in Lemma 1.

Lemma 1 (see [17]). *Let $p > 1$, $r \geq 1$, $e \geq r + 2$ and $q = p^e + 1$ be integers, and also let z be an integer such that $|z| \leq q^2/4 - q$ and $|[z]_q| \leq q/4$.*

(i) *If p is odd then* $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > (\mathrm{mod}\, p^r)$.

(ii) *If $p = 2$ then* $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > -z < e-1 > (\mathrm{mod} 2^r)$.

Lemma 1 transforms complex modular operations into simple bit operation, to get a lower-depth circuit of decryption function. But it is still not easy to execute a homomorphic bit-extraction operation. Next, [17] proposed a fast bit-extraction procedure. As stated in the former introduction, the performance of fast bit-extraction procedure is dependent on the parameter range of z in Lemma 1. That is, the bigger parameter range of z, the better performance of fast bit-extraction procedure. Thus, next we analyse whether the parameter range of z in Lemma 1 is optimal. In order to do so, we introduce a new concept called "load capacity".

Definition 2 (load capacity). Let $q \in \mathbb{Z}^+$, $z \in \mathbb{Z}$. Suppose the formula of modular reduction converted to simple bit operations works when $-q/2 < a \leq [z]_q \leq b \leq q/2$, and $c \leq z < d$. Then the load capacity is defined by the product of two span lengths of z and $[z]_q$, namely, $(b-a) \times (d-c)$.

Next Theorem 3 presents the general relationship between the value z and $[z]_q$ for the formula of modular reduction converted to simple bit operations.

Theorem 3. *Let $p > 1$, $r \geq 1$, $e \geq r + 2$ and $q = p^e + 1$ be integers, and also let z be an integer such that $[z]_q \in [a, b]$, and let $a \cdot (1-q) \leq z < (q-1) \times (q-b)$. Then*

(i) *if p is odd then* $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > (\mathrm{mod}\, p^r)$;

(ii) *if $p = 2$ then* $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > -z < e-1 > (\mathrm{mod} 2^r)$.

Input: pk_2, D, $\langle \overline{sk_{1i}} \rangle$, c
Output: c'
Step 1. $\overline{c_i} \xleftarrow{R} HE.Enc(pk_2, c_i)$ where c_i is the i-bit of c
Step 2. $c' \leftarrow HE.Eval(pk_2, D, \langle \langle \overline{sk_{1i}} \rangle, \overline{c_i} \rangle)$

Algorithm 1: "Bootstrapping" algorithm.

Proof. It starts with the odd-p case. Let $z_0 = [z]_q \in [a, b]$ and $z = z_0 + k \cdot q$ with $k \in \mathbb{Z}$. Then

$$z = z_0 + k \cdot (p^e + 1) = (z_0 + k) + k \cdot p^e. \tag{1}$$

Since $e \geq r + 2$, we can get that

$$z = z_0 + k \,(\mathrm{mod}\, p^r). \tag{2}$$

Besides, since

$$\begin{aligned} a \cdot (1-q) \leq z < (q-1) \times (q-b), \\ z_0 + k = z_0 + \frac{z - z_0}{q} = \frac{z + z_0 (q-1)}{q}, \end{aligned} \tag{3}$$

then

$$\begin{aligned} z_0 + k \in \Bigg[& \frac{a \cdot (1-q) + z_0 (q-1)}{q}, \\ & \frac{(q-1) \times (q-b) + z_0 (q-1)}{q} \Bigg). \end{aligned} \tag{4}$$

And since $-q/2 \leq a \leq z_0 \leq b \leq q/2$, then

$$\begin{aligned} z_0 + k \in \Bigg[& \frac{a \cdot (1-q) + a (q-1)}{q}, \\ & \frac{(q-1) \times (q-b) + b (q-1)}{q} \Bigg) = [0, p^e). \end{aligned} \tag{5}$$

Thus, combined with formula (2), we can get that

$$k \langle r-1, \cdots, 0 \rangle = z \langle e+r-1, \cdots, e \rangle, \tag{6}$$

where $k < r-1, \cdots, 0 >$ and $z < e+r-1, \cdots, e >$ are mod-p representation. Then it follows that

$$\begin{aligned} z_0 \langle r-1, \cdots, 0 \rangle &= z \langle r-1, \cdots, 0 \rangle - k \langle r-1, \cdots, 0 \rangle \\ &= z \langle r-1, \cdots, 0 \rangle \\ &\quad - z \langle e+r-1, \cdots, e \rangle \,(\mathrm{mod}\, p^r). \end{aligned} \tag{7}$$

The proof for the $p = 2$ case is similar. The details can be referred to in the proof of [17]. It is omitted here. □

Next we discuss how to choose the value of a, b in order to obtain the maximum "load capacity". Load capacity is denoted by t, then

$$\begin{aligned} t &= ((q-1) \cdot (q-b) - a (1-q)) \cdot (b-a) \\ &= (1-q) \cdot (b-a)^2 + q (q-1) \cdot (b-a). \end{aligned} \tag{8}$$

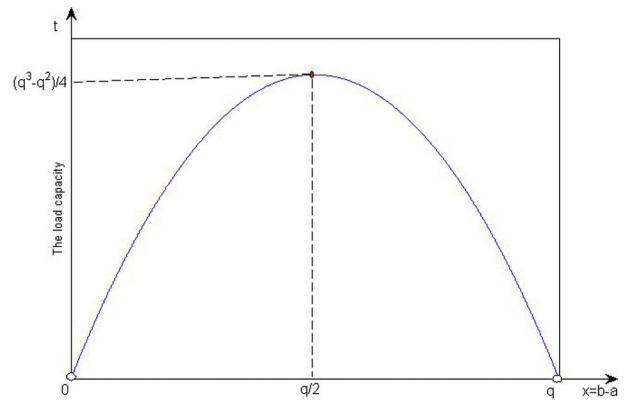


Figure 2: The load capacity on the span length of a and b.

Let $x = b - a$, then $t = (1-q)\cdot x^2 + q(q-1)\cdot x$. The concrete relations are as shown in Figure 2

It can be easily seen from Figure 2 that the load capacity takes the maximum value when $b - a = q/2$. That is, the load capacity for HS work is only related to the span length, not to the value of a and b. Then Corollary 4 presents the optimal choice of $[z]_q$ and z for the formula of modular reduction converted to simple bit operations.

Corollary 4. *Let $p > 1$, $r \geq 1$, $e \geq r+2$, $q = p^e + 1$ and $a \in (-q/2, 0]$ be integers, and also let z be an integer such that $z \in \mathbb{Z}$, $[z]_q \in [a, a+q/2]$, and $a\cdot(1-q) \leq z < (q-1)\times(q/2-a)$.*

(i) *If p is odd then* $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > (\mathrm{mod}\, p^r)$.

(ii) *If $p = 2$ then* $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > -z < e-1 > (\mathrm{mod} 2^r)$.

The conclusion is obvious; the proof is omitted here.

Note that, when $a = -q/4$, namely, $|[z]_q| \leq q/4$, it is the same as HS's work. But, the load capacity of this paper is bigger than that of HS's work, since z of ours has a bigger span length, namely, $|z| \leq q^2/4 - q/4$. The details are present in Table 1.

As seen from Table 1, compared to HS' work, it can be seen that our scheme has a better load capacity. Note that, while on the surface, it appears to obtain a tiny improvement in a nondominant term, i.e., where the load capacity of the choice in HS is $q^3/4 - q^2$, this is improved to $q^3/4 - q^2/4$, it is actually a meaningful job when you carefully analyse the principle of the trick of the fast bit-extraction procedure in [17]. That is, add to the coefficients of ct multiples of q and p^r, making them divisible by $p^{e'}$ for some $r \leq e' < e$ without increasing them too much and also without increasing the noise too much. This means that bit-extraction can be implemented using only polynomials of degree at most $e-e'$, smaller than e. Since the load capacity of this paper is $3q^2\backslash 4$ bigger than that of HS's work, it means our work allows adding more multiples of q and p^r to the coefficients of ct. That is, bit-extraction can be implemented using polynomials of lower degree to get a faster implementation. Besides, our variant of HS is more flexible and general on parameters.

4. Generalize Modulus to More General Situations

In this section, it extends HS recryption procedure to have a wider choice of ciphertext modulus. The specifics are in Theorem 5.

Theorem 5. *Let $p > 1$, $r \geq 1$, $e \geq r+2$, and $q = u\cdot p^e + v$ with $u, v \in \mathbb{Z}$ and $u \in [1, p^r - 1]$, $v \in [1, p^e - 1]$, $p \nmid u, v$, also let z be an integer such that $[z]_q \in [a, b]$, and*

$$\frac{a\cdot(v-q)}{v} \leq z < \frac{b\cdot(v-q) + p^e q}{v}. \tag{9}$$

Then,

(i) *if p is odd then* $[z]_q = z < r-1, \cdots, 0 > -(((v < r-1, \cdots, 0 >)\times(z < e+r-1, \cdots, e >))/u < r-1, \cdots, 0 >)(\mathrm{mod}\, p^r)$,

(ii) *if $p = 2$ then* $[z]_q = z < r-1, \cdots, 0 > -((v < r-1, \cdots, 0 >)\times(z < e+r-1, \cdots, e >))/u < r-1, \cdots, 0 > -z < e-1 > (\mathrm{mod}\, p^r)$,

where "$\times$"refers to scalar multiplication.

Proof. We begin with the odd-p case. Let $z_0 = [z]_q \in [a, b]$ and $z = z_0 + k\cdot q$ with $k \in \mathbb{Z}$. Then

$$z = z_0 + k\cdot(u\cdot p^e + v) = (z_0 + kv) + ku\cdot p^e. \tag{10}$$

Since $e \geq r+2$, we can get that

$$z = z_0 + kv\,(\mathrm{mod}\, p^r). \tag{11}$$

Besides, since

$$\begin{aligned} &\frac{a\cdot(v-q)}{v} \leq z < \frac{b\cdot(v-q) + p^e q}{v}, \\ &z_0 + kv = z_0 + \frac{z - z_0}{q}v = \frac{zv + z_0(q-v)}{q}, \end{aligned} \tag{12}$$

then

$$\begin{aligned} z_0 + kv \in \Bigg[&\frac{((a\cdot(v-q))/v)\,v + z_0(q-v)}{q}, \\ &\frac{((b\cdot(v-q) + p^e q)/v)\,v + z_0(q-v)}{q}\Bigg). \end{aligned} \tag{13}$$

And since $-q/2 \leq a \leq z_0 \leq b \leq q/2$, then

$$\begin{aligned} z_0 + kv \in \Bigg[&\frac{a\cdot(v-q) + a(q-v)}{q}, \\ &\frac{b\cdot(v-q) + p^e q + b(q-v)}{q}\Bigg) = [0, p^e). \end{aligned} \tag{14}$$

Table 1: The parameter of low-circuit implementation of modular reduction on HS and our work.

Scheme	z	$[z]_q$	load capacity
HS' work [17]	$[q-q^2\backslash 4, q^2\backslash 4-q]$	$[-q\backslash 4, q\backslash 4]$	$q^3/4-q^2$
Our work ($a \in (-q/2, 0]$)	$[a\cdot(1-q), (q-1)\times(q/2-a))$	$[a, a+q/2]$	$q^3/4-q^2/4$

Input: Modulus $q = u \cdot p^e + v$, and a ciphertext c encrypting a constant $b \in (\mathbb{Z}/p^{e+1}\mathbb{Z})$ relative to secret key s and modulus p^{e+1}
Output: A ciphertext c' encrypting the bit operation relative to secret key s and modulus p^e
1. Set $c_0 \leftarrow c$ //c encrypt z w.r.t. s
2. For $i = 1$ to r
3. Set $acc \leftarrow c$ // acc is an accumulator
4. For $j = 0$ to $i = 1$ //Compute $z - \sum_j p^j w_j^{i-1}$
5. Set $tmp \leftarrow HomExp(c_j, p^{i-j})$ //Homomorphic exponentiation to the power p^{i-j}
6. Set $acc \leftarrow acc - p^j \cdot tmp \bmod p^{r+1}$
7. Set $c_i \leftarrow acc \cdot ((q_0+1)/p)^i \bmod p^{r+1}$ // encrypts $z < i >_p$
8. Output $c_0 - (v\langle 0\rangle_p / u\langle 0\rangle_p)c_r \bmod p^{r+1}$

Algorithm 2: Bit-extraction (c, r, e, p).

Thus, combined with formula (11), we can get that

$$\begin{aligned} z\langle r-1,\cdots,0\rangle &= z_0\langle r-1,\cdots,0\rangle \\ &\quad + (k\langle r-1,\cdots,0\rangle) \\ &\quad \times (v\langle r-1,\cdots,0\rangle) \left(\bmod p^r\right) \\ z\langle e+r-1,\cdots,e\rangle &= (k\langle r-1,\cdots,0\rangle) \\ &\quad \times (u\langle r-1,\cdots,0\rangle) \left(\bmod p^r\right) \\ p &\nmid u, v \end{aligned} \tag{15}$$

Thus,

$$\begin{aligned} [z]_q &= z\langle r-1,\cdots,0\rangle \\ &\quad - \frac{(v\langle r-1,\cdots,0\rangle)\times(z\langle e+r-1,\cdots,e\rangle)}{u\langle r-1,\cdots,0\rangle} \left(\bmod p^r\right). \end{aligned} \tag{16}$$

The proof for the $p = 2$ case is similar. We omit it here. □

Next we discuss how to choose the value of a, b in order to obtain the maximum "load capacity". Load capacity is denoted by t, then

$$\begin{aligned} t &= \left(\frac{b\cdot(v-q)+p^e q}{v} - \frac{a\cdot(v-q)}{v}\right)\cdot(b-a) \\ &= \frac{(v-q)}{v}\cdot(b-a)^2 + \frac{(q-v)q}{uv}\cdot(b-a). \end{aligned} \tag{17}$$

Let $x = b - a$, then

$$t = \frac{(v-q)}{v}\cdot x^2 + \frac{(q-v)q}{uv}\cdot x. \tag{18}$$

It is easy to get that t takes the maximum value when

$$x = \frac{(q-v)q/uv}{-2((v-q)/v)} = \frac{q}{2u}. \tag{19}$$

That is, the load capacity is also only related to the span length, not to the value of a and b.

Then Corollary 6 presents the optimal choice of $[z]_q$ and z for the formula of modular reduction converted to simple bit operations.

Corollary 6. *Let $p > 1$, $r \geq 1$, $e \geq r+2$ and $q = u\cdot p^e + v$ with $u, v \in \mathbb{Z}$ and $u \in [1, p^r - 1]$, $v \in [1, p^e - 1]$, $p \nmid u, v$, also let z be an integer such that $[z]_q \in [a, a+q/2u]$, and*

$$\frac{a\cdot(v-q)}{v} \leq z < \frac{q^2-(v+2au)q+2auv}{2uv}. \tag{20}$$

Then

(i) *if p is odd then $[z]_q = z < r-1,\cdots,0 > -(((v < r-1,\cdots,0>)\times(z < e+r-1,\cdots,e>))/u < r-1,\cdots,0 >)(\bmod p^r)$;*

(ii) *if $p = 2$ then $[z]_q = z < r-1,\cdots,0 > -((v < r-1,\cdots,0>)\times(z < e+r-1,\cdots,e>))/u < r-1,\cdots,0 > -z < e-1 > (\bmod p^r)$.*

The conclusion is obvious; the proof is omitted here.

To get a homomorphic implementation of the simple decryption formula from above, firstly a homomorphic bit-extraction procedure (Algorithm 2) is presented, which is slightly varied from the bit-extraction procedure of [17].

$HomExp(c, n)$ uses native homomorphic multiplication to multiply *operatornamec* by itself n times. To aid exposition, this code assumes that the modulus and secret key remain fixed; otherwise modulus-switching and key-switching should be added (and the level should be increased correspondingly to some $i > 0$).

Then Algorithm 3 shows how to combine our optimized "bootstrapping" techniques with the SIMD homomorphic computation techniques of Smart-Vercauteren [23], to get a bootstrapping method that works in time quasilinear in the security parameter.

Step 1. The user first post-processes the q_L-secret-key by encrypting s as a q_0-ciphertext $c' = (c_0', c_1')$ with respect to the q_0-secret-key $s' = (1, s)$, namely the user has $[< c', s' > \bmod \Phi_m]_{q_0} = [c_0', c_1' \cdot s \bmod \Phi_m]_{q_0} = p^{r+1} \cdot k + s$ where $k \in \mathbb{Z}[X]/\Phi_m(X)$ with small coefficients.

Step 2. The server computes z homomorphically. Specifically, the server compute the mod-p^{r+1} inner product homomorphically by setting

$$\tilde{z} = \left([c_0 + c_1 \cdot c_0' \bmod \Phi_m]_{q_0}, [c_1 \cdot c_1' \bmod \Phi_m]_{q_0}\right).$$

Step 3. Apply a homomorphic inverse-DFT transformation to convert to CRT-based "packed" ciphertexts that hold the coefficients of z in their plaintext slots.

Step 4. Apply the bit extraction procedure to all these slots in parallel. The result is encryption of polynomials that have the coefficients of z in their plaintext slots.

Step 5. Apply a homomorphic DFT transformation to get back a ciphertext that encrypts the polynomial z itself.

Algorithm 3: Batched bootstrapping implementation of our scheme.

Table 2: Experimental results for our batched bootstrapping and HS.

cyclotomic ring m	plaintext space	number of slots	security level	total recrypt (sec)		space usage (GB)	
				Our work	HS' work	Our work	HS' work
$21845 = 257 \cdot 5 \cdot 17$	2^{16}	1024	76	97	172	2.3	3.0
$18631 = 601 \cdot 31$	2^{25}	720	110	168	235	2.6	3.2
$45551 = 41 \cdot 11 \cdot 101$	17^{40}	1000	106	1475	2037	11.2	13.8
$51319 = 19 \cdot 73 \cdot 37$	127^{36}	1296	161	984	1461	31.7	36.4

5. Implementation and Performance

In this section, an implementation of BGV ring-LWE-based scheme is made, since it offers nearly the most efficient homomorphic operations. This scheme is defined over a ring $R \overset{def}{=} \mathbb{Z}[X]/(\Phi_m(X))$, where $\Phi_m(X)$ is the m th cyclotomic polynomial. Let p be a prime or a prime power, and $\mathbb{A}_p := \mathbb{Z}_p[X]/\Phi_m(X)$. Specifically, assume $\Phi_m(X) \equiv F_1(X) \cdots F_\ell(X) (\mathrm{mod} p)$, where each F_i has the same degree d, which is equal to the order of p modulo m. Then, by the Chinese Remainder Theorem, it has the isomorphism $R_p \cong \bigoplus_{i=1}^{k} (\mathbb{Z}[X]/(p, F_i(X)))$. Besides, suppose $\mathrm{sk} = (1, s)$ is the q_L-secret-key, where $s \in \mathbb{Z}[X]/\Phi_m(X)$ is an integer polynomial with small coefficients. $\mathrm{sk}' = (1, s')$ is the q_0-secret-key. $c = (c_0, c_1)$ is the q_L-ciphertext.

First, several groups (m, p, r) are chosen which satisfy $\Phi_m(X) \equiv F_1(X) \cdots F_\ell(X) (\mathrm{mod}\, p)$. For each triple (m, p, q), a test is run separately based on our work and HS' work. These tests were run on a four-year-old IBM System x3850 server, with two 64-bit 4-core Intel Xeon E5450 processors, and 35MB L2 cache and 32GB of RAM at 3.0 GHz. And the implementation was mainly based on Shoup's NTL library [23] version 9.10.0 and GNU's GMP library [24]. The former is used for high-level numeric algorithms, and the latter is used for the underlying integer arithmetic operations. Besides, the code was compiled using the gcc compiler (version 4.9.1). Table 2 summarizes the results from our experiments based on our work and HS'.

The first column gives cyclotomic ring m and its factorization into prime powers. The second column gives the plaintext space, i.e., the field/ring that is embedded in each slot. The third column gives the number of slots packed into a single ciphertext. The fourth column gives the effective security level, computed using the formula that is used in HElib taken from [15, Eqn. (8)]. The total recrypt gives the total time for a single recryption, while the previous two rows give a breakdown of that time (note that the time for the linear transforms includes some trivial preprocessing time, as well as the less trivial unpacking/repacking time). The last two rows give the memory used (in gigabytes).

As seen from Table 2, compared to HS' work, it can be easy seen that the variant of HS has advantages both in efficiency and in storage space. Besides the variant of HS is more flexible and general on parameters. This enables our method to be applied in a larger number of situations.

6. Conclusions

Up to now, Gentry's bootstrapping technique is still the only known method of obtaining a "pure" FHE scheme. Meanwhile it is also the key for the low efficiency of FHE scheme. It is very meaningful to improve the efficiency of bootstrapping, which mainly refers to lower-depth circuit implementation of decryption function. In this paper, it improves the "load capacity" of HS's work with a better efficiency for bootstrapping and to generalize q to more general situations in a similar simple way. This enables our method to be applied in a larger number of situations, such as privacy protection of IoT.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was sponsored in part by the National Natural Science Foundation of China (Grants nos. 61272041, 61202491, 61272488, and 61601515) and was also supported by the Foundation of Science and Technology on Information Assurance Laboratory (no. KJ-15-006).

References

[1] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[2] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Information Sciences*, vol. 379, pp. 42–61, 2017.

[3] W.-T. Song, B. Hu, and X.-F. Zhao, "Optimizing LWE-based FHE for better security and privacy protection of smart city," *Journal of Information Science and Engineering*, vol. 33, no. 4, pp. 939–952, 2017.

[4] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and pri-vacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 169–180, 1978.

[5] M. van Dijk, C. Gentry, S. Halevi et al., "Fully homomorphic encryption over the integers," in *Proceedings of the 29th International Conference on The-ory and Application of Cryptographic Techniques*, pp. 24–43, Springer, Berlin, Germany, 2010.

[6] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Proceedings of the 13th International Con-ference on Pronctice and Theory in Public Key Cryptography*, pp. 420–443, Springer, Berlin: Germany, 2010.

[7] J.-S. Coron, T. Lepoint, and M. Tibouchi, "Scale-invariant fully homomorphic encryption over the integers," in *Proceedings of the 17th Interna-tional Conference on Practice and Theory in Public Key Cryptography*, vol. 8383 of *Lecture Notes in Computer Science*, pp. 311–328, Springer, Berlin, Germany, 2014.

[8] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic en-cryption from (standard) LWE," in *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science*, pp. 97–106, IEEE Computer Society, Washington, Wash, USA, 2011.

[9] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Proceedings of the 31st Annual Conference on Advances in Cryptology*, pp. 505–524, Springer, Berlin, Germany, 2011.

[10] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully ho-momorphic encryption without bootstrapping," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, Optimizing GSW-FHE and Private Information Retrieval 21, pp. 309–325, ACM Press, New York, NY, USA.

[11] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Proceedings of the 32nd Cryptology Conference*, pp. 868–886, Springer, Berlin, Germany, 2012.

[12] C. Gentry, S. Halevi, C. Peikert et al., "Ring switching in BGV-style homomorphic encryption," in *Proceedings of the 8th International Security and Cryptography for Networks*, pp. 19–37, Springer, Berlin, Germany, 2012.

[13] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attrib-ute-based," in *Proceedings of the 33rd Annual Cryptology Conference*, pp. 75–92, Springer, Berlin, Germany, 2013.

[14] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pp. 169–178, ACM Press, New York, NY, USA, 2009.

[15] C. Gentry, S. Halevi, and N. P. Smart, "Better bootstrapping in fully ho-momorphic encryption," in *Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography*, pp. 1–16, Springer, Berlin, Germany, 2012.

[16] J. Alperin-Sheriff and C. Peikert, "Practical bootstrapping in quasilinear time," in *Proceedings of the 33rd Annual Cryptology Conference*, pp. 1–20, 2013.

[17] S. Halevi and V. Shoup, "Bootstrapping for helib," in *Eurocrypt*, pp. 641–670, 2015.

[18] Z. Brakerski and V. Vaikuntanathan, "Lattice-based FHE as secure as PKE," in *ITCS*, pp. 1–8, 2014.

[19] J. Alperin-Sheriff and C. Peikert, *Faster Bootstrapping with Polyno-mial Error*, vol. 8616 of *Lecture Notes in Computer Science*, 2014.

[20] R. Hiromasa, M. Abe, and T. Okamoto, "Packing Messages and Optimiz-ing Boot-strapping in GSW-FHE," in *Public-Key Cryptography-PKC*, pp. 699–715, Springer, Heidelberg, Berlin, Germany, 2015.

[21] L. Ducas and D. Micciancio, "FHEW: Bootstrapping homomorphic encryption in less than a second," in *Eurocrypt*, pp. 617–640, 2015.

[22] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 seconds.

[23] V. Shoup, *NTL: A Library for doing Number Theory*, 2016, http://shoup.net/ntl/.

[24] "The GNU Multiple Precision Arithmetic Library," 2016, http://gmplib.org/.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7218061, 10 pages
https://doi.org/10.1155/2018/7218061

Research Article

Multitask Allocation to Heterogeneous Participants in Mobile Crowd Sensing

Weiping Zhu,[1] **Wenzhong Guo**,[1,2,3] **Zhiyong Yu**,[1,2] **and Haoyi Xiong**[4]

[1] *College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China*
[2] *Fujian Provincial Key Laboratory of Networking Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China*
[3] *Key Laboratory of Spatial Data Mining and Information Sharing, Ministry of Education, Fuzhou 350003, China*
[4] *Department of Computer Science, Missouri University of Science and Technology, MO 65409, USA*

Correspondence should be addressed to Wenzhong Guo; guowenzhong@fzu.edu.cn

Received 10 March 2018; Revised 21 April 2018; Accepted 9 May 2018; Published 14 June 2018

Academic Editor: Huaqun Wang

Task allocation is a key problem in Mobile Crowd Sensing (MCS). Prior works have mainly assumed that participants can complete tasks once they arrive at the location of tasks. However, this assumption may lead to poor reliability in sensing data because the heterogeneity among participants is disregarded. In this study, we investigate a multitask allocation problem that considers the heterogeneity of participants (i.e., different participants carry various devices and accomplish different tasks). A greedy discrete particle swarm optimization with genetic algorithm operation is proposed in this study to address the abovementioned problem. This study is aimed at maximizing the number of completed tasks while satisfying certain constraints. Simulations over a real-life mobile dataset verify that the proposed algorithm outperforms baseline methods under different settings.

1. Introduction

An era of "Internet of Things" has been reached given the development of wireless communication, sensor technology, smartphones, and wearable devices. A new sensing paradigm called Mobile Crowd Sensing (MCS), where mobile devices play an important role in large-scale sensing and information sharing, has become research issue in academia and industry. Due to its advantages such as low cost and wide spatiotemporal coverage, MCS applications have been studied like intelligent transportation [1, 2], environment monitoring [3], target identification [4], and so on.

In contrast to traditional wireless sensor networks (WSN) [5], MCS is a human-centered sensing model, in which MCS applications must recruit participants to complete the sensing tasks. A straightforward way for obtaining a highly reliable sensing data is to recruit as many participants as possible to complete tasks. However, this strategy results in a high sensing cost. A key issue is allocating the tasks to proper participants, while accounting for the various initial locations of different participants, sensing data reliability, and sensing cost. Therefore, task allocation is an important issue in linebreak MCS.

Due to the importance, several studies have been conducted to select participants to complete tasks [6–20]. In these schemes, participants are selected to achieve a specific goal (e.g., maximize the regional coverage) under certain constraints, such as the budget cost. These studies have assumed that every user is indiscriminate; that is, everyone can complete any task provided that this individual arrives at the location of tasks. This assumption is reasonable in the initial stage of crowd sensing, where sensing tasks are simple and everyone can complete any type of task.

However, sensing tasks have become complicated with the increase in demands of crowd sensing. The abovementioned assumption may be invalid. On one hand, several tasks can be rather complex and not easily completed by participants. The platform requires that participants must preassemble special sensors or possess certain skills to accomplish tasks. On the other hand, different devices integrated into various sensors, which can accomplish different tasks. To complete complex tasks, it is imperative for platform to

select participants whose devices preassemble the required sensors.

In this study, we consider the multitask allocation to heterogeneous participants (MTHP). The tasks may require the participants, whose devices preassemble the required sensors, to move to the location of tasks at a given time and complete these tasks. From the perspective of participants, everyone differs in sensing capability, including sensor type and the number of tasks that they can completed. There are two challenges that must be considered to overcome this problem. First, we must consider the tasks that the participants can complete during a specific duration in accordance with the sensors that participants carry and the location of participants and tasks, but only a few participants exist for all tasks. Second, from perspective of platform, it expects as many tasks as possible to be allocated, although several constraints may prevent the completion of the tasks. These constraints include the maximum number of tasks that participants can complete and the distance between the location of tasks and their current location. So the other challenge is on coordinating with participants to complete different tasks. Thus, we explore the multitask allocation problem by considering the heterogeneity of participants. The main contributions of this study are as follows:

(1) We formulate the problem of MTHP, which considers the heterogeneity among participants. Specifically, participants are different in sensor type, maximum workload, and moving speed. The object of MTHP is to maximize the number of accomplished tasks under the sensing capacity and time constraints.

(2) A greedy discrete particle swarm optimization with genetic algorithm (GDPSOGA) operation is specifically designed to tackle the multitask allocation problem. It first selects participants using heuristic strategies to reduce the search space of discrete particle swarm optimization (DPSO). Then, the random two-point mutation and random two-point crossover operations in genetic algorithm (GA) are incorporated to coordinate the participant resource among tasks and further maximize the number of completed tasks.

(3) Experiments on a real-world mobile usage dataset indicate that the proposed algorithm outperforms baseline methods under different settings.

The remainder of this paper is organized as follows: in Section 2, the related works are presented. In Section 3, the problem of MTHP is described in detail, and then our proposed algorithm and strategy are introduced in Section 4. In Section 5, we compare our algorithm with baseline algorithms and evaluate the performance of proposed algorithm. Finally, conclusions drawn from this study are presented in Section 6.

2. Related Work

Task allocation is a key research issue in MCS and has drawn considerable attention from researchers. In [6, 7], Reddy et al. considered the location, time constraints, and habits of users and proposed a coverage-based framework to select proper participants to maximize spatial coverage. Xiong et al. [8] defined a temporal-spatial coverage called k-depth coverage and then discussed selection of participants to maximize coverage under the constraints of budget and minimize the budget while satisfying the predefined coverage goal. Zhang et al. [9] investigated coverage quality and selected a subset of mobile users to maximize coverage quality under constrained budget. Several researchers considered selecting participants to minimize cost while guaranteeing data quality. Karaliopoulos et al. [10] studied the manner for selecting mobile users to minimize cost while ensuring the coverage of points of interest. Zhang et al. [11] predicted the mobility of participants and then selected minimum number of participants to meet the predefined temporal-spatial coverage. Wang et al. [12] proposed a framework that considers the spatial and temporal correlations among different subareas to reduce the number of participants required. Another work [13] defined a new coverage metric, namely, "t-sweep k-coverage", and proposed two methods for selecting smallest set of candidate participants to satisfy predefined requirements. However, these works studied the task allocation problem for single task and have disregarded the competition of participants for sensing tasks.

Recently, several works have been proposed for the multitask allocation. Li et al. [14] proposed a greedy-based participant selection algorithm for heterogeneous tasks to minimize the number of participants while guaranteeing a certain level of coverage. Liu et al. [15] studied the multitask allocation problem under the two conditions. The first is few participants and many tasks. The second is many participants and few tasks. Wang et al. [16] considered the maximum workload of participants and proposed a two-phase offline multitask allocation approach. In [17], the goal is to select a subset of participants to maximize the quality of information under budget constraints. Guo et al. [18] proposed a worker selection framework for time-sensitive and delay-tolerant tasks. Time-sensitive tasks are aimed at minimizing the distance traveled by workers. For delay-tolerant tasks, the goal is to minimize the total number of workers. He et al. [19] studied the optimal allocation algorithms for location-dependent tasks to maximize the reward of platform. In [20], a novel framework was proposed to improve the data accuracy of task allocation with the aid of fog-nodes.

The related works presented above have mainly assumed that participants can accomplish tasks as long as they arrive at the location of tasks and ignored the heterogeneity among participants. For several complicated tasks, platform must select participants who already preassembled specific sensors to get reliable sensing data. In our study, we propose a task allocation framework that considers the heterogeneity among participants. In particular, we study the problem of multitask allocation toward heterogeneous participants, which assumes that everyone carries a mobile device integrated with different sensors and can accomplish different types of task. We present the modified PSO algorithm to address it.

3. Problem Formulation

In this section, we present the formal definition of the MTHP. Assume that the union of sensors is denoted as $S = \{s_1, s_2, s_3, \ldots, s_k\}$. Each user possesses one or multiple sensors

in S. There are m users on the platform, which are denoted by the set $U = \{u_1, u_2, u_3, \ldots, u_m\}$. Each user can be depicted by a tuple: $(\mathrm{id}_i, \mathrm{loc}_i, S_i, v_i, q_i)$. It indicates that user u_i has a set S_i ($\subseteq S$) of sensors, is located at position loc_i, and can move with velocity v_i anywhere. Due to the limited sensing capability, we assume that u_i can complete at most q_i tasks. Giving n different tasks on the platform are denoted by the set $T = \{t_1, t_2, t_3, \ldots, t_n\}$. Each task can be depicted by a tuple of five elements: $(\mathrm{loc}_j, s_j, p_j, st_j, et_j)$. To ensure the quality of sensing data, p_j users who preload sensor s_j are required to move to the location loc_j to complete the task during the time interval $[st_j, et_j]$. We use $T_{u_i} = \{t_1, t_2, t_3, \ldots\}$ to denote the task set that is allocated to u_i and $U_{t_j} = \{u_1, u_2, \ldots\}$ to denote the user set that completes the task t_j. The MTHP problem aims to allocate the tasks to the subset of the user set so that the number of tasks that can be accomplished is maximized.

To complete tasks, users must travel from their current location to the location of the tasks. We use Manhattan distance [21] to measure the distance between the location of the tasks and the users.

Based on the abovementioned definition, the MTHP problem can be formulated as

$$\max \quad \sum_{i=1}^{m} \left|T_{u_i}\right|$$

$$\text{s.t.} \quad \begin{cases} \left|T_{u_i}\right| \le q_i, & 1 \le i \le m \\ \left|U_{t_j}\right| = p_j, & 1 \le j \le n \\ S_{t_j} \in S_{u_i}, & 1 \le i \le m,\ 1 \le j \le n \\ st_j \le \dfrac{\mathrm{dis}\left(\mathrm{loc}_{u_i}, \mathrm{loc}_{t_j}\right)}{v_i} \le et_j, & u_i \in U_{t_j}, \end{cases} \tag{1}$$

where $\mathrm{dis}(\mathrm{loc}_{u_i}, \mathrm{loc}_{t_j})$ is the Manhattan distance between the task j and the user i.

This is a combinatorial optimization problem, and the solution space is quite large. For example, given n tasks and m users that satisfy all the constraints on the platform, if every task requires p participants, then there are $(C_m^p)^n$ combination schemes, and the value sharply increases with the increase of n, m, and p. Thus a heuristic task allocation algorithm must be designed to reduce the search space of the problem. Furthermore, considering the limited sensing capacity of participants, MTHP not only allocates tasks to users under the various constraints, but also coordinates users among tasks to further maximize the number of completed tasks, which makes the task allocation more complex. Several efficient coordination strategies must be introduced to achieve optimal utilization of participants and maximize the number of completed tasks. According to the analysis above, we adopt the greedy DPSO algorithm with GA (GDPSOGA) operation to solve this problem.

4. Algorithm

4.1. Basic Particle Swarm Optimization. Particle swarm optimization (PSO) is a population-based intelligence algorithm that simulates the movement of a flock of birds to seek food. It is a popular optimization method in many fields due to its simplicity and fast convergence. For PSO, a particle is defined as a potential solution to a problem in the D-dimensional space. Each particle adjusts its position and velocity according to its own experience and that of neighboring particles during the iteration. The particles are manipulated according to the following formula:

$$v_i^{t+1} = w \times v_i^t + c_1 r_1 \left(p_i - x_i^t\right) + c_2 r_2 \left(p_g - x_i^t\right) \tag{2}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{3}$$

where t is the iteration index; v_i^t and x_i^t represent the velocity and position of particle i in the iteration t, respectively. p_i and p_g represent the previous optimal position of particle i and optimal position of all particles after $t-1$ iterations, respectively. w is the inertia weight, which indicates the impact of the last iteration to the current iteration. c_1 and c_2 are the acceleration factors, which reflect the simulative ability to the previous optimal solution and global optimal solution. r_1 and r_2 are random numbers distributed uniformly on the interval from 0 to 1.

Problem (1) is a discrete problem. Thus, the standard PSO is not appropriate for this problem due to its continuous nature. Inspired by our previous work [22], a DPSO is designed here to solve the problem.

4.2. Particle Representation. In the scenario of task allocation, each particle represents a potential allocation scheme for all tasks. A favorable particle representation considers not only the redundancy of the search space, but also the efficiency of the algorithm. According to [22], the particle encoding scheme should follow three main principles, that is, nonredundancy, completeness, and soundness.

Definition 1 (nonredundancy). A one-to-one relationship exists between the encoding scheme and the potential solution in the problem space.

Definition 2 (completeness). All feasible solutions can be represented by the particle according the encoding scheme.

Definition 3 (soundness). Each particle in the encoding space must correspond to the potential solution in the problem space.

Satisfying all three principles simultaneously is difficult for an encoding scheme. The task allocation problem is selecting users to complete tasks, so we adopt the task-user encoding scheme to represent a particle. Each point of particle indicates the user that is selected to complete corresponding task. For example, in Figure 1, four tasks are presented, and every task requires three participants. t_1 is allocated to u_1, u_2, and u_4, and other tasks are allocated similar to t_1.

Properties. The task-user encoding scheme satisfies the principles of completeness and soundness but does not meet the principle of nonredundancy.

Input: Task set T, User set U, the number of users that each task required p
Output: The <task, users> tuple
(1) For each task t in T
(2) Find the users with the required sensor and maximum workload > 0, denotes as U_t
(3) Sort the U_t according to the moving time, maximum workload and number of sensors
(4) Greedily select users from U_t whose moving time is shortest
(5) If moving time is equal, select non-competitive users firstly
(6) If all users are competitive users, select users whose devices with fewer sensors
(7) Delete task t if there are p users to complete it
(8) End for
(9) Output the <task, users> tuple

Algorithm 1: Greedy strategy algorithm.

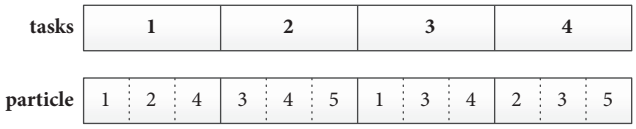


Figure 1: Particle encoding.

Proof. Each point of a particle represents a potential task allocation of corresponding task, so all feasible solutions of tasks can be represented by the particle according to the task-user encoding scheme; thus the task-user encoding scheme satisfies the principle of completeness. The problem optimizes the number of completed tasks, and the result can be computed through particle. So every particle corresponds to a potential solution in the problem space; i.e., the task-user encoding scheme satisfies the principle of soundness. However, the problem only counts the number of completed tasks, and different particles may correspond to the same result, although the solutions are different. Thus, the task-user encoding scheme does not satisfy the principle of nonredundancy. □

4.3. Fitness Function. Fitness function is introduced to evaluate the accuracy of each individual in achieving the goal of a problem. As discussed in Section 3, from perspective of platform, the optimization goal of the MTHP problem is to allocate tasks as many as possible. So we define the fitness function to count the number of completed tasks, as shown in formula (4).

$$\text{fitness} = \left|T_{u_i}\right| \tag{4}$$

4.4. Particle Initialization. For the PSO, the initial state of particle significantly influences the result because the final solution is derived from the initial solution. In general, the particle should be randomly initialed from the potential solution space. However, a random initialization particle may be far away from the optimal solution due to the huge solution space. Thus, several heuristic strategies are required for initialization processing.

For the MTHP problem, we attempt to maximize the number of completed tasks within the tasks during time. Inspired by work [18], we design a greedy strategy to select users. We firstly select the users that preload the sensor that the task required and move to the location of tasks within the shortest possible time. We give the priority to moving time to ensure that tasks can be completed while users have more extra time to complete other tasks.

When selecting candidate users according to the moving time, some of these users may be also required by other tasks simultaneously. In this case, we should consider the situation of maximum workload. Inspired by work [16], we define a user as competitive if the number of tasks that tend to select the user is greater than the maximum workload of the user. Otherwise we define the user as noncompetitive. We firstly consider the noncompetitive user and then leave the competitive user to other tasks to maximize the utilization of users.

Furthermore, the number of sensors that users carry must be considered. If all candidates are competitive users, we select the user with fewer sensors and leave users with more sensors to other types of tasks. The pseudocode is presented in Algorithm 1.

For example, as Tables 1 and 2 show, there are four tasks on the platform. The sensor they need is depicted as A, B, C, and D, respectively. Each task requires two different users to complete and only users carry the corresponding type of sensor can complete the task. There are three users u_1, u_2, u_3 on the platform. The circumstance of sensors and maximum workload are shown in Table 2. For simplicity, we assume that the moving time of users is equal and the task allocation depends on the maximum workload and sensors of users. Firstly, we consider the allocation of task t_1, u_1 and u_3 carry the sensor A, and thus we allocate the task A to u_1 and u_3, and then the maximum workload of the u_1 and u_3 declines to 1 and 2, respectively. For task t_2, all users can complete it, but all users are competitive. In this case, we firstly select the users who carry fewer sensors to complete task; i.e., we select u_1 and u_2 to complete t_2. After that, the maximum workload of u_1 and u_2 declines to 0 and there are no more tasks that can be completed because of the limited capacity of all users. Thus, the number of completed tasks is 2 after particle initialization. The allocation result is summarized in Table 3.

4.5. Particle Updating. In general, the greedy strategy in the particle initialization cannot guarantee obtaining global

Table 1: Task set.

Tasks	Sensor Required	p
t_1	A	2
t_2	B	2
t_3	C	2
t_4	D	2

Table 2: User set.

Users	Sensors Preloaded	Maximum Workload
u_1	A, B, C	2
u_2	B, D	1
u_3	A, B, C, D	3

Table 3: Task allocation.

Tasks	Users Selected
t_1	u_1, u_3
t_2	u_1, u_2
t_3	u_3
t_4	u_3

optimal solution because this strategy selects the best local participants for every task. Thus, several refined strategies must be designed to improve this situation. Inspired by our previous work [22], we incorporate the mutation operation and crossover operation in GA to update the particles.

Formula (2) defines three parts for velocity updating. We incorporate the mutation operation in the first part. Tasks coordinate participants through the mutation operation. The updating formula is defined as follows:

$$A_i^t = w \otimes M_u\left(X_i^{t-1}\right) = \begin{cases} M_u\left(X_i^{t-1}\right) & r_1 < w \\ X_i^{t-1} & \text{esle,} \end{cases} \tag{5}$$

where M_u is the mutation operation with the probability of w. r_1 is random number between 0 and 1. The mutation operation randomly selects two points of a particle and then changes the value of all points between the two points. Note that the mutation operation must ensure satisfying all constraints in formula (1). The operation is illustrated in Figure 2.

The inertia weight influences the convergence and search ability. When we set a smaller w, PSO performs with high ability in local search; otherwise it performs with high ability in global search. In the early period of the algorithm, we must pay more attention to the diversity and global search ability of particles. To guarantee the convergence of algorithm, we must focus on the local search ability in the late period. Therefore, w should be reduced with the iterations. The updated function is expressed as follows:

$$w = w_{\max} - \text{iters}_{\text{cur}} \times \frac{w_{\max} - w_{\min}}{\text{iters}_{\max}}, \tag{6}$$

where $w_{\max}$ and $w_{\min}$ represent the maximum and minimum value of w, respectively. $\text{iters}_{\text{cur}}$ is the current iteration and $\text{iters}_{\max}$ is the maximum iterations.

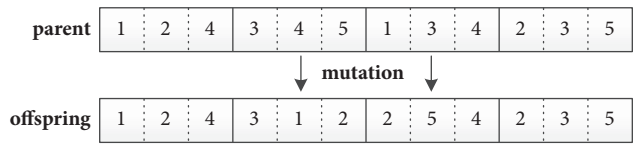


Figure 2: Mutation operation.

The second and third parts of formula (2) are the cognition of particle, indicating a particle learns from its previous optimal position and global optimal position. For these parts, we adopt the notion of crossover operation in the GA to update the particles, and the updating formulas are defined as follows:

$$\begin{aligned} B_i^t &= c_1 \oplus C_p\left(A_i^t, pBest^{t-1}\right) \\ &= \begin{cases} C_p\left(A_i^t, pBest^{t-1}\right) & r_2 < c_1 \\ A_i^t & \text{else} \end{cases} \\ X_i^t &= c_2 \oplus C_g\left(B_i^t, gBest^{t-1}\right) \\ &= \begin{cases} C_g\left(B_i^t, gBest^{t-1}\right) & r_3 < c_2 \\ B_i^t & \text{else,} \end{cases} \end{aligned} \tag{7}$$

where c_1 and c_2 are the crossover probabilities. Particles learn from previous optimal and current global optimal positions through crossover operation and finally converge to the optimal solution. The crossover operation must also ensure satisfying all constraints in formula (1). The operation is depicted in Figure 3.

In PSO, c_1 and c_2 are important parameters that reflect information exchange among particles. If we set greater c_1, particles intend to learn more from their personal scheme and hover within a local range. If we set greater c_2, particle may get into local optimality early. To keep the diversity of population, particles must learn more from their personal optimal scheme with great probability during the early stage of iteration. However, to guarantee the convergence of the algorithm, participates must learn more from global best scheme with great probability in the late stage of iterations. Thus c_1 should decrease while c_2 increase with the iterations. We update two acceleration factors with the linear strategy. The updated formulas are expressed as follows:

$$\begin{aligned} c_1 &= c_1\_\text{start} - \frac{c_1\_\text{start} - c_1\_\text{end}}{\text{iters}_{\max}} \times \text{iters}_{\text{cur}} \\ c_2 &= c_2\_\text{start} - \frac{c_2\_\text{start} - c_2\_\text{end}}{\text{iters}_{\max}} \times \text{iters}_{\text{cur}}, \end{aligned} \tag{8}$$

where c_1_start and c_2_start are the initial values of c_1 and c_2, respectively. c_1_end and c_2_end are the final values of c_1 and c_2, respectively.

In summary, the position of the particle i at iteration t can be updated according to following formula:

Input: The result of Algorithm 1
Output: Number of completed tasks
(1) Initialize the relative parameters;
(2) Iteration = 1;
(3) Get the particle i by operation of mutation and crossover;
(4) Compute the number of completed tasks and update the $pbest_i$ and $gbest$ if necessary;
(5) Iteration++;
(6) If the value of $gbest$ keeps the same for twenty times, terminate the iteration;
(7) If iteration < Maximum iteration, go to (3);
(8) Output the number of completed tasks.

Algorithm 2: GDPSOGA.

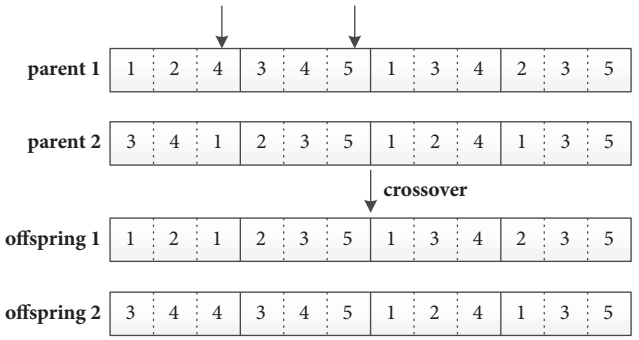


Figure 3: Crossover operation.

$$X_i^t = c_2 \oplus C_g\left(c_1 \oplus C_p\left(w \otimes M_u\left(X_i^{t-1}\right), pBest_i^{t-1}\right), gBest^{t-1}\right) \quad (9)$$

4.6. Algorithm Overview. To address the MTHP problem, we propose the GDPSOGA. The pseudocode is presented in Algorithm 2. According to Algorithm 2, we initiate the particles with the result of Algorithm 1. To improve the result of greedy scheme, we incorporate the notions of mutation and crossover operations in GA into the DPSO. The integration can maintain not only the diversity of population, but also preferable characteristics of offspring population.

Taking Tables 1 and 2 as an example, there are two tasks that can be completed after particle initialization. If a particle changes the allocation of t_2 to u_1 and u_3, then, u_2 has spare capacity to complete t_4. So t_4 can be completed by u_2 and u_3. The mutation operation is presented in Figure 4. Thus, the number of completed tasks can be improved to 3. Note that because of the randomness of mutation, we only present a possible case in the mutation operation.

4.7. Time Complexity

Theorem 4. *The time complexity of GDPSOGA is $O(n \times k \log k + R \times S \times n \times p)$, where n is the number of tasks. p indicates the number of candidate participants that every task requires. k is the number of participants that carry the required sensor. R denotes the maximum number of iterations. S represents the population size of the GDPSOGA.*

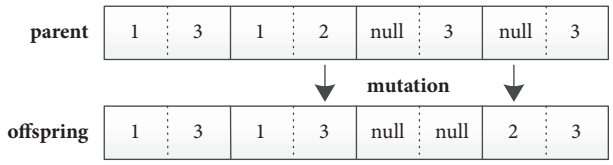


Figure 4: Mutation operation of tasks in Table 1.

Proof. In the first step of Algorithm 1, we sort the participants according to their moving time and sensors; the complexity is $O(n \times k \log k)$.

For the second step of Algorithm 1, we greedily select participants to initiate the particle. The length of particles is $n \times p$. Thus the time complexity of particle initialization is $O(S \times n \times p)$.

During each round of the iteration, the GDPSOGA updates particles by incorporating the two-point mutation and two-point crossover operations in the GA. In the worst case, the whole particle is updated. Thus the time complexity of updating is $O(S \times n \times p)$. Furthermore, the complexity of updating R iterations is $O(R \times S \times n \times p)$.

In summary, the time complexity of the GDPSOGA in the worst case is $O(n \times k \log k + R \times S \times n \times p)$. □

5. Evaluation and Discussion

In this section, we conduct experiments to examine the performance of our proposed algorithm under different settings. First, we present the dataset and experiment settings. Then, we introduce some baseline algorithms for evaluation. Finally, the detailed results of the proposed and baseline algorithms are presented and analyzed.

5.1. Dataset. We evaluate the performance of the proposed algorithm using the D4D dataset [23]. It is a large-scale real-world dataset that contains two types of data. One type is the location information of 1231 cell towers. The other type comprises individual call records for over 50,000 users of the Orange Group during two weeks in Ivory Coast. Each record includes user id, connection time, and cell tower. We design experiments based on the location information of cell towers and users.

(a) Compact (b) Hybrid (c) Scattered

Figure 5: Three types of tasks distribution.

Table 4: Relative parameters of the GDPSOGA.

Parameter	Value or Range
population size	100
inertia value w	0.9 to 0.4
acceleration factor c_1	0.9 to 0.2
acceleration factor c_2	0.4 to 0.9
maximum iteration	1000

5.2. Experiment Settings. For candidate participants, we select the candidate participants who made a phone call at the scope of cell towers between 18:00 and 19:00 on November 11, 2011. The maximum workload of users is set randomly between 5 and 10 and the walking speed of users is randomly set between 65 and 70 meters per minute. For tasks, we set the location of cell towers as the location of tasks. The sensor type that tasks require is randomly set between "A" and "E". The duration of tasks is randomly set to 1 to 3 minutes. Furthermore, we select different task distributions to measure the performance of our proposed method.

(i) Compact Distribution. Tasks are distributed compactly in a special region, and the distance among the tasks is relatively close.

(ii) Scattered Distribution. Tasks are scattered over the target region, and the distance among the tasks is relatively far.

(iii) Hybrid Distribution. Tasks are randomly distributed in the target area.

An illustration of three types of distribution is presented in Figure 5.

For the GDPSOGA, the relative parameters are listed in Table 4.

5.3. Baseline Algorithms. To compare the effectiveness of the proposed algorithm, we design three baseline task allocation methods as follows.

(i) Random Allocation (RA). This method selects participants randomly provided that the participants satisfy all constraints in formula (1) and without considering any heuristic strategy in this scheme. We repeat the random selection for 30 times and compute the average value of completed tasks.

(ii) Greedy Strategy Based Allocation (GSA). This method selects participants as presented in Algorithm 1, which considers the moving time, maximum workload of participants, and sensor number of devices, and does not implement the process of PSO.

(iii) Discrete PSO with GA (DPSOGA). This method selects participants using the PSO with mutation and crossover operations, without any heuristic strategy. Similar to the GDPSOGA, the DPSOGA terminates the iteration after the global optimal solution is kept unchanged for 20 times.

5.4. Number of Completed Tasks Comparison. We complete experiments in different situations to evaluate the performance of GDPSOGA. In the first set of simulations, we evaluate the influence of the number of participants required for every task. We fix the number of tasks at 300 and total number of candidate participants at 200. We vary p from 4 to 12. Figure 6 demonstrates that the number of completed tasks decreases when p increases. This is reasonable because finding a sufficient number of participants to complete task is difficult given the limited candidate participants. GSA and GDPSOGA can complete most of tasks when p is set to 4. With the increment of p, GSA only selects the optimal participants for every task under limited participant resource. GDPSOGA coordinates the user resource after GSA and thus obtains better results than GSA.

In the next set of simulations, we evaluate the influence of number of tasks. Here we set number of candidate participants at 160 and the required number of participants by every task at 5. The experiments are completed with varying number of tasks from 200 to 300. Figure 7 displays the results. In general, the number of completed tasks increases when the task number increases. However, RA and GSA cannot improve the result when the number of tasks reaches a certain extent because, on one hand, the number of candidate participants is limited. On the other hand, the two schemes merely select participants for every task and do not coordinate all candidate participants for all tasks.

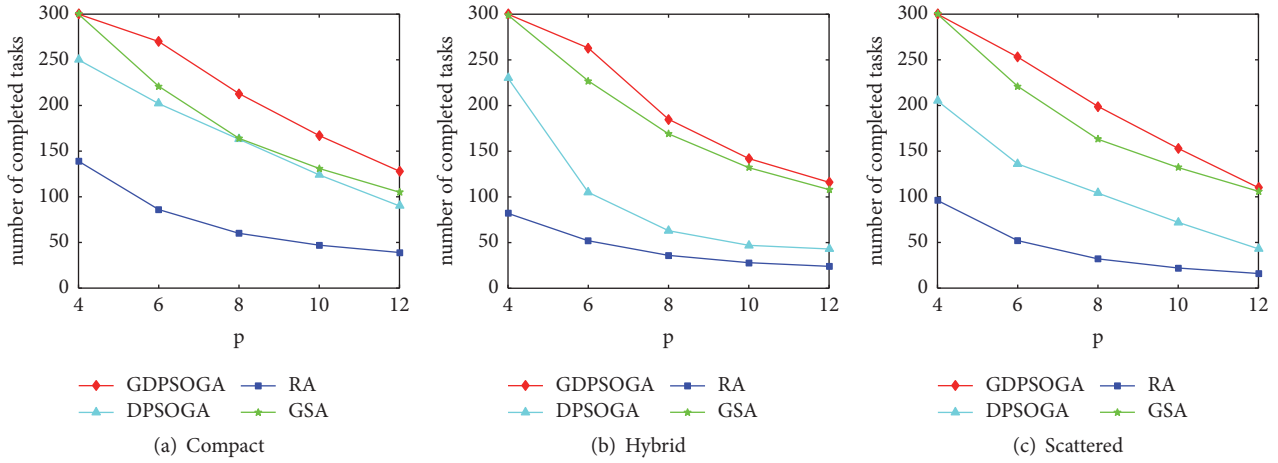


Figure 6: Performance comparison under different p.

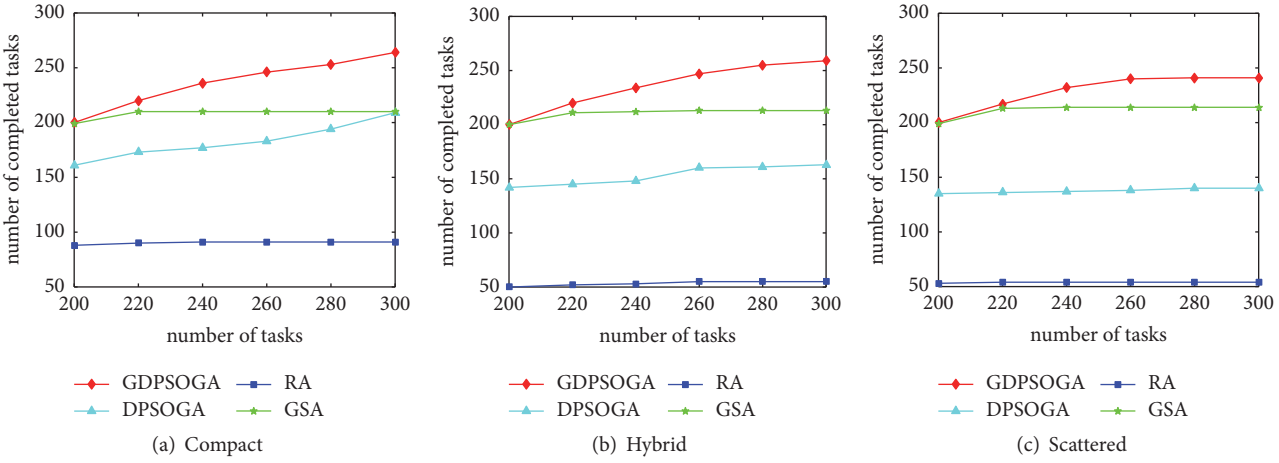


Figure 7: Performance comparison under different number of tasks.

In the last set of simulations, we fix the number of tasks at 300 and the required number of participants by every task at 5. Here, we vary the number of candidate participants from 100 to 200. Figure 8 illustrates the results under different task distributions. The number of completed tasks increases with candidate participants because additional candidate participants may offer increased possible optimized allocation schemes.

In general, the completion ratio of tasks is higher in compact distribution than in the two other distributions. For example, the average completion ratio of the three types of distribution in the second simulation is 95.2%, 94.9%, and 92.3%. Participants may not need to travel long distance to complete tasks because the locations among tasks are nearest. Thus, more tasks can be completed within the tasks during time.

Among the different task assignment schemes, the GDPSOGA outperforms other schemes because the GDPSOGA combines the advantage of the DPSO and GSA. Specifically, a heuristic strategy, which can effectively reduce the solution space of our problem, is adopted in the early stage of the GDPSOGA to obtain the optimal or suboptimal solution. However, the heuristic strategy selects the local optimal solution for every task. It cannot guarantee the optimal solution for all tasks. Thus several strategies must be adopted to coordinate the participants to complete as many tasks as possible. The GDPSOGA adopts the crossover and mutation operations to match the participants among tasks after a greedy strategy participant selection; thereby it improves the results.

5.5. Computing Time. In this section, we compare the computing time of all algorithms. We employ Java language to implement the algorithm. All the experiments are conducted on a PC with 3.10 GHz CPU and 16 GB memory.

Table 5 shows a comparison of computing time of the first simulation in the compact distribution. From Table 5, we can see that RA takes the least time because it just selects participants as long as participants satisfy all the constraints and does not consider any heuristic strategy. GSA firstly sorts the participants according to the moving time, maximum workload, and the sensors of participants and then greedily selects participants according to the sort results. So it takes more time than the RA. Our algorithm takes longer time than

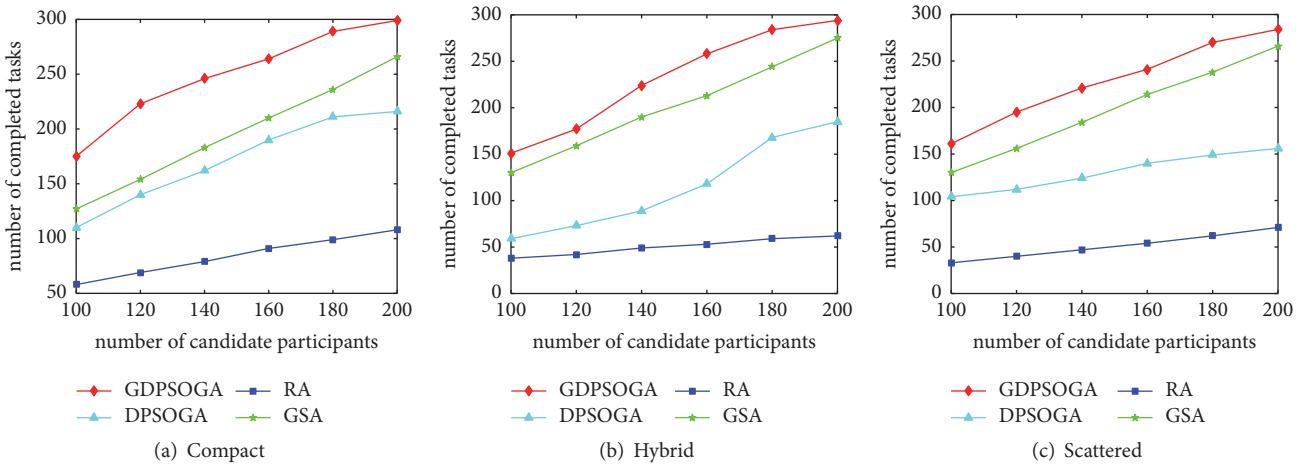


Figure 8: Performance comparison under different number of candidate participants.

Table 5: Comparison of CPU time (s) of second simulation in compact distribution.

Algorithms	p				
	4	6	8	10	12
GDPSOGA	1.294	1.816	2.238	2.452	2.76
DPSOGA	1.124	1.135	1.654	2.229	2.578
GSA	0.102	0.105	0.11	0.115	0.124
RA	0.02	0.023	0.027	0.028	0.032

GSA due to the iteration of PSO to coordinate the participant resource among tasks. But it is worthy, because it improves the number of completed tasks and obtains the results within reasonable time. Note that without any heuristic strategy DPSOGA may fall into local optimum, so it takes less time than GDPSOGA.

6. Conclusion

In this study, we focus on the problem of MTHP in MCS to maximize the completed tasks while satisfying certain constraints. In contrast to other existing works, this study considers the heterogeneity among participants. We propose the GDPSOGA to solve the problem, which incorporates the random two-point crossover and random two-point mutation operations in GA into DPSO. Extensive simulations conducted over a real-life dataset confirm the efficiency of the proposed algorithm. However, in this study, we assume that the tasks are static and no new sensing tasks emerge after the task allocation starts. In our future work, we will explore the allocation schemes for dynamic tasks.

Data Availability

Dataset used in this article is released by D4D Challenge. Participants in this challenge signed an agreement on data confidentiality. So it is available in limited way. One of the coauthors participated in this challenge. So access to the dataset can be got. Some introduction about the dataset can be obtained from https://arxiv.org/abs/1210.0137.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (nos. U1705262, 61672159, and 61772136), the Outstanding Youth Rolling Project of Fujian Province under Grant no. 2018J07005, the Technology Innovation Platform Project of Fujian Province under Grant nos. 2014H2005 and 2009J1007, the Fujian Collaborative Innovation Center for Big Data Application in Governments, and the Fujian Engineering Research Center of Big Data Analysis and Processing.

References

[1] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, "Cooperative transit tracking using smart-phones," in *Proceedings of the 8th ACM International Conference on Embedded Networked Sensor Systems, SenSys 2010*, pp. 85–98, November 2010.

[2] N. Pham, R. K. Ganti, Y. S. Uddin, S. Nath, and T. Abdelzaher, "Privacy-preserving reconstruction of multidimensional data maps in vehicular participatory sensing," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5970, pp. 114–130, 2010.

[3] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: when urban air quality inference meets big data," in *Proceedings of the 19th ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13), pp. 1436–1444, Chicago, Ill, USA, August 2013.

[4] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. S. Yilmaz, and H. Ferhatosmanoglu, "Crowd-sourced sensing and collaboration using twitter," in *Proceedings of the 2010 IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks", WoWMoM 2010*, can, June 2010.

[5] W. Guo, J. Li, G. Chen, Y. Niu, and C. Chen, "A PSO-Optimized Real-Time Fault-Tolerant Task Allocation Algorithm in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3236–3249, 2015.

[6] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *Pervasive Computing*, vol. 6030 of *Lecture Notes in Computer Science*, pp. 138–155, Springer, Berlin, Germany, 2010.

[7] S. Reddy, K. Shilton, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using context annotated mobility profiles to recruit data collectors in participatory sensing," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5561, pp. 52–69, 2009.

[8] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, "ICrowd: Near-Optimal Task Allocation for Piggyback Crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2010–2022, 2016.

[9] M. Zhang, P. Yang, C. Tian et al., "Quality-aware sensing coverage in budget-constrained mobile crowdsensing networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7698–7707, 2016.

[10] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks, IEEE INFOCOM 2015*, pp. 2254–2262, May 2015.

[11] D. Zhang, H. Xiong, L. Wang, and G. Chen, "CrowdRecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*, pp. 703–714, 2014.

[12] L. Wang, D. Zhang, A. Pathak et al., "CCS-TA: quality-guaranteed online task allocation in compressive crowdsensing," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*, pp. 683–694, ACM, Osaka, Japan, September 2015.

[13] Z. Yu, J. Zhou, W. Guo, L. Guo, and Z. Yu, "Participant selection for t-sweep k-coverage crowd sensing tasks," *World Wide Web*, pp. 1–18, 2017.

[14] H. Li, T. Li, and Y. Wang, "Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks," in *Proceedings of the 12th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2015*, pp. 136–144, Dallas, TX, USA, October 2015.

[15] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "TaskMe: Multi-task allocation in Mobile Crowd Sensing," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp 2016*, pp. 403–414, deu, September 2016.

[16] J. T. Wang, Y. S. Wang, D. Q. Zhang, F. Wang, Y. D. He, and L. T. Ma, "PSAllocator: multi-task allocation for participatory sensing with sensing capability constraints," in *Proceedings of the International Conference on Computer-Supported Cooperative Work and Social Computing (CSCW)*, pp. 1139–1151, 2015.

[17] Z. Song, C. H. Liu, J. Wu, J. Ma, and W. Wang, "QoI-aware multitask-oriented dynamic participant selection with budget constraints," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4618–4632, 2014.

[18] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han, "ActiveCrowd: A Framework for Optimized Multitask Allocation in Mobile Crowdsensing Systems," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 392–403, 2017.

[19] S. He, D. Shin, J. Zhang, and J. Chen, "Near-optimal allocation algorithms for location-dependent tasks in crowdsensing," *IEEE Transactions on Vehicular Technology*, pp. 1-1, 2017.

[20] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. S. Shen, "Providing Task Allocation and Secure Deduplication for Mobile Crowdsensing via Fog Computing," *IEEE Transactions on Dependable and Secure Computing*, pp. 1-1.

[21] D.-J. Chang, A. H. Desoky, M. Ouyang, and E. C. Rouchka, "Compute pairwise Manhattan distance and Pearson correlation coefficient of data points with GPU," in *Proceedings of the 10th ACIS Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2009, In conjunction with IWEA 2009 and WEACR 2009*, pp. 501–506, Daegu, South Korea, May 2009.

[22] W. Guo, W. Hong, B. Zhang, Y. Chen, and N. Xiong, "Reliable adaptive data aggregation route strategy for a trade-off between energy and lifetime in WSNs," *Sensors*, vol. 14, no. 9, pp. 16972–16993, 2014.

[23] V. D. Blondel, M. Esch, C. Chan et al., "Data for Development: the D4D Challenge on Mobile Phone Data," 2012, https://arxiv.org/abs/1210.0137.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 1701675, 12 pages
https://doi.org/10.1155/2018/1701675

Research Article

Traceable Ciphertext-Policy Attribute-Based Encryption with Verifiable Outsourced Decryption in eHealth Cloud

Qi Li,[1,2,3] **Hongbo Zhu**,[3] **Zuobin Ying**,[4] **and Tao Zhang**[5]

[1]*School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*
[2]*Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*
[3]*Jiangsu Innovative Coordination Center of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[4]*School of Computer Science and Technology, Anhui University, Hefei 230601, China*
[5]*School of Computer Science and Technology, Xidian University, Xian 710071, China*

Correspondence should be addressed to Qi Li; liqics@njupt.edu.cn

Received 8 March 2018; Revised 19 April 2018; Accepted 7 May 2018; Published 6 June 2018

Academic Editor: Kim-Kwang Raymond Choo

In cloud-assisted electronic health care (eHealth) systems, a patient can enforce access control on his/her personal health information (PHI) in a cryptographic way by employing ciphertext-policy attribute-based encryption (CP-ABE) mechanism. There are two features worthy of consideration in real eHealth applications. On the one hand, although the outsourced decryption technique can significantly reduce the decryption cost of a physician, the correctness of the returned result should be guaranteed. On the other hand, the malicious physician who leaks the private key intentionally should be caught. Existing systems mostly aim to provide only one of the above properties. In this work, we present a verifiable and traceable CP-ABE scheme (VTCP-ABE) in eHealth cloud, which simultaneously supports the properties of verifiable outsourced decryption and white-box traceability without compromising the physician's identity privacy. An authorized physician can obtain an ElGamal-type partial decrypted ciphertext (PDC) element of original ciphertext from the eHealth cloud decryption server (CDS) and then verify the correctness of returned PDC. Moreover, the illegal behaviour of malicious physician can be precisely (white-box) traced. We further exploit a delegation method to help the resource-limited physician authorize someone else to interact with the CDS. The formal security proof and extensive simulations illustrate that our VTCP-ABE scheme is secure, efficient, and practical.

1. Introduction

Electronic health care (eHealth) system is regarded as an outstanding approach to provide well health care service through various emerging technologies, including Internet of Things, cloud computing, mobile computing, and wireless sensor networks. In cloud-assisted eHealth systems, an individual patient integrates his/her personal health information (PHI) collected via various wearable and embedded sensors, stores the PHI in the cloud, and receives real-time and high-quality medical treatment. Unfortunately, when the patient enjoys convenient storage services provided by cloud server, the risk of privacy exposure also raises. The sensitive PHI may be exposed to the cloud server which can not be fully trusted. Even worse, the PHI may be widely propagated to unauthorized parties for commerce benefit or other purposes. Thus, the PHI must be encrypted before hosted to the eHealth cloud. Meanwhile, an access policy must be specified to point out who are authorized to access the PHI.

Aiming to realize access control on encrypted message, attribute-based encryption (ABE) [1] was presented to provide an efficient solution to this kind of applications. According to the place where the access policy is embedded, the ABE schemes are divided into two forms, key-policy ABE (KP-ABE) [2] and another type of ABE named ciphertext-policy ABE (CP-ABE) [3]. In the former framework, every user's key is labeled with an access policy while the ciphertexts are annotated with chosen sets of attributes. On the contrary, the

user's key in CP-ABE is issued according to his/her attributes while the ciphertext is encrypted under an access policy. Since that ABE is a feasible mechanism which preserves the security and privacy of patients' PHI, a series of attribute-based access control systems [4–8] have been proposed, aiming at expressive policies, security, or efficiency. In particular, there remain two significant features to be considered in utilizing ABE technique in eHealth systems.

The first feature is verifiability of outsourced decryption. In most ABE systems [1–3, 9–12], the decryption overhead is linear to the scale of involved attributes and expensive for energy-constrained terminals. The decryption outsourcing technique [13] is proposed to reduce the number of exponential operations and bilinear pairing operations on user side by offloading the heavy decryption computation to a third-party server, e.g., the cloud server. The user then recovers the plaintext by executing only one exponential operation over ElGamal-style partial decrypted ciphertext element generated by the third-party server. However, such outsourced scheme can not guarantee the correctness of returned ElGamal-style element. Lai *et al.* [14] presented the verifiable approach in ABE to check whether the third-party server has honestly executed the decryption service. They also bring redundant overhead in both encryption computation and ciphertext size. Qin *et al.* [15] provided an efficient verifiable ABE scheme which significantly reduces the computation cost in encryption and the decryption overhead for users.

Another considerable feature is traceability. We take CP-ABE as an instance; the private key is generated from some descriptive attributes rather than from a unique identity. Each attribute may be possessed by multiple users. It could be impossible to distinguish who is the original owner of a given private key. Imagine two physicians in eHealth systems, Tomas and Jack. They have the attribute set '{orthopedics department, chief physician}' which is not possessed by any other users. By the key delegate technique [3], both Tomas and Jack can regenerate a private key responding to the set '{orthopedics department, chief physician}', if there is a third user who can decrypt the ciphertext labeled by access policy '{'orthopedics department' *AND* 'chief physician'}'. Where did the key come from? Tomas or Jack? To solve the problem above, Liu *et al.* [16] extended an adaptively secure CP-ABE scheme [9] to support 'white-box' traceability, where the malicious user directly leaks his/her private key. Subsequently, Ning *et al.* [17] constructed a large attribute universe and traceable CP-ABE scheme. On the contrary to the 'small universe' in [3, 10, 14–16], the 'large universe' means that the scale of attribute universe is unbounded [18].

However, existing works mostly aimed to support the property of verifiable outsourced decryption or traceability separately. There is no CP-ABE scheme with both verifiable outsourced decryption and white-box traceability in practice: (1) the CP-ABE schemes [16, 17] support the traceability well, but the user's decryption cost grow with the attribute number; (2) these CP-ABE schemes [14, 15, 19, 20] provide decryption assistance for users, and the correctness of returned PDC element is guaranteed; however, the traceability property is not addressed.

In this work, we propose a novel verifiable and traceable CP-ABE scheme named VTCP-ABE for eHealth cloud applications. The VTCP-ABE scheme is the first scheme which simultaneously achieves white-box traceability and verifiable outsourced decryption without exposing the physician's identity information. Since we take the 'large universe' scheme [18] as the basis, the attribute universe in our scheme is inherently unbounded. We further extend the VTCP-ABE to support another delegation property. We also provide the formal proof of the selective CPA security, verifiability, and traceability for VTCP-ABE. The comparison and simulation results show that our VTCP-ABE is applicable for practical eHealth cloud applications. In particularly, we make the following contributions:

(1) We propose a new VTCP-ABE scheme which simultaneously achieves the properties of verifiable outsourced decryption, white-box traceability, and large universe. An authorized physician can check the correctness of partial decrypted ciphertext (PDC) which is requested from the eHealth CDS. Given a private key, the original owner can be precisely tracked. The attribute universe can be exponentially large and the number of public parameter elements is constant no matter how many attributes are chosen.

(2) We present an efficient approach to prevent the CDS from knowing the fixed identification information of physician during offering decryption service. The original ciphertext and the transmission private key will be preprocessed before being sent to the CDS. This method is acceptable since only two additional exponential operations for each decryption request are added.

(3) We exploit an additional property of delegation for our VTCP-ABE, with which a resource-constrained physician can delegate someone to obtain a PDC element without compromising the privacy of PHI.

1.1. Related Works. ABE was first introduced in [1]. The first KP-ABE scheme with threshold tree access structures was presented in [2]. The first CP-ABE scheme with the same structures was presented in [3]. Waters [21] presented several CP-ABE schemes to support the access policy defined as Linear Secret Sharing Schemes (LSSS). Yu *et al.* [22] demonstrated the deployment of ABE technique in cloud computing. In [4], Li *et al.* presented a personal health record (PHR) secure sharing scheme in cloud computing. Subsequently, various constructions of ABE schemes were presented in [9, 23–29].

Green *et al.* [13] constructed the first decryption outsourcing ABE, where the most decryption overhead is hosted to a third party. With the returned partial decrypted ciphertext, a user could recover the plaintext message by executing only one exponential operation. Based on the outsourced method [13], Li *et al.* [7] presented a PHR data sharing scheme for cloud storage applications in the multi-authority settings. In both [7, 13], the correctness of returned PDC is not guaranteed. Lai *et al.* [14] presented an approach to check whether the partial decrypted ciphertext element (transformed ciphertext element) is correctly calculated. Their technique incurred noticeable overhead in both decryption and encryption. Based on key encapsulation mechanism, Lin

Table 1: Comparison between ours and some related works.

Systems	CP/KP	AU[1]	OD[2]	Verifiability	Traceability	Delegation
Rouselakis *et al.* [18]	CP,KP	Large	No	No	No	No
Green *et al.* [13]	CP,KP	Small	Yes	No	No	No
Lai *et al.* [14]	CP	Small	Yes	Yes	No	No
Qin *et al.* [15]	CP	Small	Yes	Yes	No	No
Liu *et al.* [16]	CP	Small	No	No	Yes	No
Ning *et al.* [17]	CP	Large	No	No	Yes	No
Our VTCP-ABE	CP	Large	Yes	Yes	Yes	Yes

[1]AU is the abbreviation of attribute universe.
[2]OD is the abbreviation of outsourced decryption.

et al. [19] and Qin *et al.* [15] separately proposed a fascinating method to support verifiable outsourced decryption in ABE. The difference between [19] and [15] is that, in [19], the hash value of a random group element R is set as the symmetric key to encrypt the original data, then R is encrypted by a ABE scheme to obtain a ABE-type ciphertext, which will be used to generate the verification key. In [15], the original data M is encrypted along with a randomly chosen bit string r, while the verification key is set by executing exponential operations in the group by taking the hash values of M and r as exponents.

Liu *et al.* presented the first adaptively secure and white-box traceable CP-ABE scheme in [16], where any monotonic LSSS access structure is supported. They further constructed another CP-ABE scheme with black-box traceability in [30]. Based on the scheme [31], Ning *et al.* [17] exploited the white-box traceability for CP-ABE in large universe settings. From then on, many traceable ABE constructions are proposed in [6, 32, 33]. However, in these traceable schemes [6, 16, 17, 30, 32, 33], the decryption overhead grows with the scale of attribute set adopted in decryption.

Table 1 compares the characteristics between some related works and our VTCP-ABE. From Table 1, our VTCP-ABE scheme is the only practical scheme to simultaneously support the properties of large universe, verifiable outsourced decryption, white-box traceability, and delegation in CP-ABE.

2. Preliminaries

2.1. Bilinear Maps. Denote $\mathbb{G}$ and $\mathbb{G}_1$ as two multiplicative cyclic groups with prime order p. g is a generator of group $\mathbb{G}$. The bilinear map $e : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_1$ has the following properties:

(1) Bilinearity: $\forall \varsigma, \eta \in \mathbb{G}$ and $\iota, \upsilon \in \mathbb{Z}_p$, $e(\varsigma^\iota, \eta^\upsilon) = e(\varsigma, \eta)^{\iota\upsilon}$.

(2) Non-degeneracy: $e(g, g) \neq 1$.

(3) Computability: for all $\varsigma, \eta \in \mathbb{G}$, $e(\varsigma, \eta)$ are efficiently computable.

Since that $e(g^\iota, g^\upsilon) = e(g, g)^{\iota\upsilon} = e(g^\upsilon, g^\iota)$, e is symmetric.

2.2. Linear Secret Sharing Schemes (LSSS)

Definition 1. Linear Secret Sharing Schemes [21, 34]: let $\mathbb{P}$ denote a set of attributes, and let p be a chosen prime. Let $T \in \mathbb{Z}_p^{m\times n}$ be a matrix. For all $i = 1, \ldots, m$, a function ρ labels the i-th row of T with an attribute (*i.e.,* $\rho \in \mathcal{F}([m] \to \mathbb{P})$). A secret sharing scheme Π over the attribute universe $\mathbb{P}$ is linear if one has the following:

(1) The shares for each attribute make a vector over $\mathbb{Z}_p$.

(2) In order to generate the shares of a secret $s \in \mathbb{Z}_p$, we select the column vector $\overrightarrow{\varrho} = (s, \varrho_2, \ldots, \varrho_n)^\top$, where $\varrho_2, \ldots, \varrho_n$ are randomly selected from $\mathbb{Z}_p$, then $T\overrightarrow{\varrho}$ is the shares of s according to Π. The share $(T\overrightarrow{\varrho})_i$ belongs to the attribute $\rho(i)$.

As demonstrated in [34], the linear reconstruction property of LSSS is defined as follows: Suppose (T, ρ) is the access structure $\mathbb{T}$ and S is an authorized set. Let $I = \{i : \rho(i) \in S\}$ be the index set of rows which are linked with the attributes in S. There exist constants $\{\psi_i \in \mathbb{Z}_p\}_{i\in I}$ which satisfy that if $\{\lambda_i = (T\overrightarrow{\varrho})_i\}$ are valid, then we have $\sum_{i\in I} \psi_i\lambda_i = s$.

2.3. φ-Type Assumption. The security of VTCP-ABE is reduced to a φ-type assumption [18].

Suppose $\mathbb{G}$ is a cyclic group and prime p is the group order. Randomly pick $g \in \mathbb{G}$ and choose $\iota, s, \upsilon_1, \upsilon_2, \ldots, \upsilon_\varphi \in \mathbb{Z}_p$. If an adversary $\mathcal{A}$ is given the group description $(p, \mathbb{G}, \mathbb{G}_1, e)$ and Ξ including all of the following terms:

$$\Xi =$$

$$g, g^s$$

$$g^{\iota^i}, g^{\upsilon_j}, g^{s\upsilon_j}, g^{\iota^i\upsilon_j}, g^{\iota^i/\upsilon_j^2}, \forall (i, j) \in [\varphi, \varphi]$$

$$g^{\iota^i/\upsilon_j}, \forall (i, j) \in [2\varphi, \varphi] \text{ with } i \neq \varphi + 1$$

$$g^{\iota^i\upsilon_j/\upsilon_{j'}^2}, \forall (i, j, j') \in [2\varphi, \varphi, \varphi] \text{ with } j \neq j'$$

$$g^{s\iota^i\upsilon_j/\upsilon_{j'}}, g^{s\iota^i\upsilon_j/\upsilon_{j'}^2} \ \forall (i, j, j') \in [\varphi, \varphi, \varphi] \text{ with } j \neq j'$$

It must be hard for $\mathcal{A}$ to distinguish the element $e(g, g)^{\iota^{\varphi+1}s} \in \mathbb{G}_1$ from a randomly chosen element $F \in \mathbb{G}_1$.

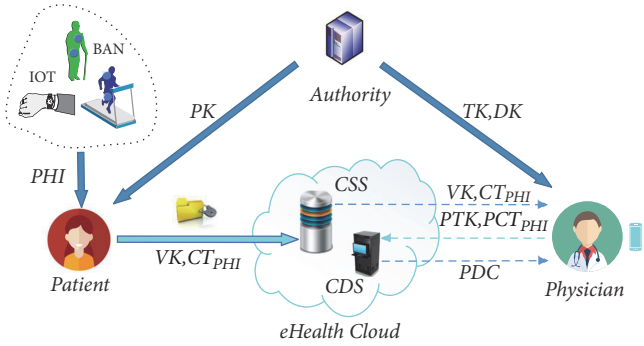


Figure 1: Architecture of VTCP-ABE in eHealth cloud.

The advantage of an algorithm $\mathscr{A}$ which solves the above q-type problem is

$$Adv_{\mathscr{A}(\lambda)} = \left|\Pr\left[\mathscr{A}\left(\Xi, W = e\left(g,g\right)^{t^{\varphi+1}s}\right) = 0\right] - \Pr\left[\mathscr{A}\left(\Xi, W = F\right) = 0\right]\right| \quad (1)$$

Definition 2. We claim that the φ-type assumption holds if the advantage of all polynomial time adversaries is negligible in the above φ-type game.

2.4. ϑ-Strong Diffie-Hellman Assumption (ϑ-SDH). The ϑ-SDH problem [35, 36]: suppose $\mathbb{G}$ is a cyclic group. Let prime p be the group order. g is randomly selected from $\mathbb{G}$. Given a $(\vartheta+1)$-tuple $(g, g^{\omega}, g^{\omega^2}, \ldots, g^{\omega^{\vartheta}})$, output a pair $(\delta, g^{1/(\omega+\delta)}) \in \mathbb{Z}_p \times \mathbb{G}$. An algorithm $\mathscr{A}$ has advantage ϵ in solving ϑ-SDH problem if $\Pr[\mathscr{A}(g, g^{\omega}, g^{\omega^2}, \ldots, g^{\omega^{\vartheta}}) = (\delta, g^{1/(\omega+\delta)})] \geq \epsilon$, where the probability is over the random bits consumed by $\mathscr{A}$ and the randomness of $\omega \in \mathbb{Z}_p$.

Definition 3. We claim that the (ϑ, t, ϵ)-SDH assumption holds if the advantage of all t-time adversaries is at most ϵ in solving the above ϑ-SDH problem.

3. System Architecture and Security Model

3.1. System Description. As shown in Figure 1, our VTCP-ABE framework in the eHealth cloud mainly consists of the following components.

The authority: the authority produces the system parameters and generates private keys for the legal physicians depending on their attributes. It is also in charge of tracing the malicious physicians.

The patient: with the help of IOT techniques, the patient integrates and then encrypts his/her PHI under appropriate access policy and further uploads the ciphertext to the eHealth cloud storage server.

The eHealth cloud storage server (CSS): the eHealth CSS provides storage service for the patient. If necessary, the patient can call CSS to delete his/her PHI data.

The eHealth cloud decryption server (CDS): the eHealth CDS provides pre-decryption service of the encrypted PHI and returns the partial decrypted ciphertext to the authorized physician.

The physician: the physician takes responsibility of medical treatment for the patient whose access policy accepts his/her attributes. The physician is also enabled to check the correctness of returned pre-decryption results from the CDS. The malicious physician may leak his private key for economic benefit or some malignant purpose.

We note that the eHealth CSS and CDS are assumed to be semi-trusted as in [22]. That is, the CSS and CDS honestly execute the pre-set algorithms. But they attempts to get useful information of the encrypted PHI as much as possible. In addition, the eHealth CDS may want to obtain the identification information of physician.

As one of the important applications in IOT environments, the eHealth cloud system enables the patient to collect his PHI via wearable devices, physiologic sensor nodes and body area networks, etc. Before uploading the PHI to the cloud sever to get real-time health care services, the patient can define expressive access policy of his PHI over descriptive attributes by VTCP-ABE. According to the assigned attributes, the individual physicians have differential flexible access rights. They can provide various (free or paid) health care services by smart devices on condition that their attributes match the access policy of patient's PHI. Our VTCP-ABE also offers the traceability to prevent the key abuse problem and the verifiable outsourced decryption technique to offload most decryption cost to the cloud server and guarantee the returned results.

3.2. Definition of VTCP-ABE. Our VTCP-ABE is comprised by the following seven algorithms.

Setup$(\kappa, U) \rightarrow (PK, MSK)$: this algorithm takes in a security parameter κ and the system attribute universe U. It then outputs the system public parameters PK and the master secret key MSK. Besides, it initializes an identity table $IT = \varnothing$.

Encrypt $(M, PK, \mathbb{T}) \rightarrow (CT, VK)$. This algorithm takes in a message M, PK, and an access structure $\mathbb{T}$. It then outputs a ciphertext CT and a verification key VK.

KeyGen $(PK, MSK, id, S) \rightarrow (TK, DK)$. This algorithm takes in PK, MSK, an identity information id and an attribute set S. It then outputs a transmission private key TK and a user decryption key DK.

Pre-Process $(PK, CT, TK) \rightarrow (PCT, PTK)$. This algorithm takes in CT and TK. It then outputs a pre-processed ciphertext PCT and a pre-processed private key PTK.

Pre-Decrypt $(PCT, PTK) \rightarrow (PDC)$. This algorithm takes in PCT and PTK. If S matches $\mathbb{T}$, the algorithm outputs a partial decrypted ciphertext PDC. Otherwise, it outputs $\perp$.

Decrypt $(PDC, DK, VK) \rightarrow (M)$. This algorithm takes in PDC, DK, and VK. If PDC is not valid, it outputs $\perp$. Otherwise, it outputs a message M.

Trace $(IT, PK, TK, DK) \rightarrow id$ or $\top$. This algorithm takes in IT, PK, TK, and DK. It first verifies whether TK and DK are well-formed. If so, this algorithm outputs the id annotated with TK and DK. Otherwise, it outputs $\top$ implying that TK and DK are not required to be traced. If TK and DK can pass

a "key sanity check" which means that they can be used in the normal decryption phase, they are called well-formed [16].

3.3. CPA Security Model. Similar to [17, 18], the definition of selective security model of VTCP-ABE against chosen plaintext attack (CPA) is given as follows:

Init. The adversary $\mathcal{A}$ gives the simulator $\mathcal{B}$ the challenge access policy $\mathbb{T}^*$.

Setup. $\mathcal{B}$ runs **Setup** to produce (PK, MSK) and passes PK to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can ask $\mathcal{B}$ to produce the private keys for $((id_1, S_1), (id_2, S_2), \ldots, (id_{Q_1}, S_{Q_1}))$. For each (id_i, S_i), $\mathcal{B}$ returns by the corresponding private key pairs (TK_i, DK_i). Note that, for each $i \in [Q_1]$, S_i can not match $\mathbb{T}^*$.

Challenge Phase. $\mathcal{A}$ submits two messages M_0 and M_1 of equal length. $\mathcal{B}$ encrypts M_μ under $\mathbb{T}^*$ to obtain CT^* and VK^*, where μ is randomly chosen from $\{0, 1\}$. It then gives CT^* and VK^* to $\mathcal{A}$.

Phase 2. As in **Phase 1**, $\mathcal{B}$ is asked to produce the private keys of $((id_{Q_1+1}, S_{Q_1+1}), \ldots, (id_Q, S_Q))$.

Guess. $\mathcal{A}$ guesses μ' for μ. $\mathcal{A}$'s advantage is defined as $\Pr[\mu' = \mu] - 1/2$.

Definition 4. We claim that a VTCP-ABE scheme is selectively CPA secure if the advantage is negligible for all PPT adversaries in the above selective security game.

3.4. Security Game for Verifiability. Based on the replayable chosen ciphertext attack (RCCA) security model [13, 15], we briefly introduce the verifiability game as follows.

Setup. The challenger $\mathcal{B}$ generates (PK, MSK) and sends PK to the attacker $\mathcal{A}$.

Phase 1. $\mathcal{B}$ queries the results from the *Create*, *Corrupt*, and *Decrypt* oracles as in [15].

Challenge Phase. The attacker $\mathcal{A}$ submits an access policy $\mathbb{T}^*$ and a message M^*. $\mathcal{B}$ encrypts M_* under $\mathbb{T}^*$ to obtain (CT^*, VK^*) and sends them to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ repeats the key queries as in **Phase 1**.

Output. $\mathcal{A}$ gives $\mathcal{B}$ PDC^* and an attribute set S^* which satisfies $\mathbb{T}^*$.

The attacker $\mathcal{A}$ wins the above game if **Decrypt** $(PDC^*, DK^*, VK^*) \notin \{M, \perp\}$. $\mathcal{A}$'s advantage in this game is defined as $ADV_{VTCP-ABE}\mathcal{A}$.

Definition 5. We claim that a VTCP-ABE scheme is verifiable if $ADV_{VTCP-ABE}\mathcal{A}$ is negligible for all PPT attackers in the above game.

3.5. Security Game for Traceability. The traceability game of our VTCP-ABE is defined as follows.

Setup. The challenger $\mathcal{B}$ generates (PK, MSK) and sends PK to the attacker $\mathcal{A}$. It keeps MSK as a secret key.

Key Query. $\mathcal{A}$ submits the tuples $(id_1, S_1), (id_2, S_2), \ldots, (id_q, S_q)$ to $\mathcal{B}$, where q refers to the query number that $\mathcal{A}$ can make.

Key Forgery. $\mathcal{A}$ outputs $DK_\star$ and $TK_\star$. $\mathcal{A}$ wins if **Trace** $(IT, PK, DK_\star, TK_\star) \neq \top$ and **Trace**$(IT, PK, DK_\star, TK_\star) \notin \{id_1, id_2, \ldots, id_q\}$. $\mathcal{A}$'s advantage is defined as $\Pr[\textbf{Trace}(IT, PK, DK_\star, TK_\star) \neq \{\top\} \cup \{id_1, id_2, \ldots, id_q\}]$.

Definition 6. We claim that a VTCP-ABE scheme is fully traceable if the advantage is negligible for all PPT attackers in the above game.

4. The Proposed VTCP-ABE

In this section, we first briefly introduce the techniques of constructing a verifiable and traceable CP-ABE scheme and then give the details of VTCP-ABE construction.

4.1. Technical Overview. To achieve the traceability in [17], each private key is associated with a unique fixed number δ so that the key owner cannot re-randomize his own private key to get a completely new key. In the verifiable CP-ABE scheme with outsourced decryption [15], the private key is composed of a transmission key and a user decryption key. The transmission key is sent to a third party to get the partial decryption result and the user decryption key is used to decrypt the partial decryption result and check its correctness.

Our goal is to achieve the efficient user decryption and traceability without compromising the security and privacy. However, if we combine the traceable CP-ABE [17] and the verifiable outsourced decryption approach [15] in a naive way, the fixed identifier number δ will be exposed to the eHealth CDS. Even worse, the CDS may use δ and the transmission private key to fabricate a key which could pass the check in the traceable algorithm of [17]. That is, a legal physician may be framed to be malicious and further revoked from the system. To prevent the CDS from knowing δ, we process the transmission private key and original ciphertext before submitting them to the eHealth CDS. Meanwhile, we add the user decryption key as input of the traceable algorithm. Finally, we add the property of verifiable outsourced decryption into the traceable CP-ABE scheme [17] at a very low cost on the physician side (one additional element in private key, two additional exponential operations in pre-processing)

4.2. Detailed Construction. We now give the detailed construction of the VTCP-ABE.

Setup. Given a group description $G = (p, \mathbb{G}, \mathbb{G}_1, e)$, where prime order p is the order of groups $(\mathbb{G}, \mathbb{G}_1)$ and e denotes a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$. The system attribute universe is set as $U = \mathbb{Z}_p$. Then randomly pick $g, u, \hbar, w, \nu \in \mathbb{G}$ and $\alpha, a \in \mathbb{Z}_p$.

Select two collision-resistant hash functions $HA_1 : \mathbb{G}_1 \to \{0,1\}^{\ell_{HA_1}}$ and $HA_2 : \{0,1\}^* \to \{0,1\}^{\ell_{HA_2}}$. $SE =$ (**SE-Keygen**, **SE-Encrypt**, **SE-Decrypt**) refers to a one-time symmetric encryption scheme and the key space is defined as $\{0,1\}^{\ell_{SE}}$. Select $HA_3 : \mathbb{G}_1 \to \{0,1\}^{\ell_{SE}}$ from $\mathcal{H}$, which denotes a party of pairwise independent hash functions.

It sets $(G, g, u, \hbar, w, \nu, e(g,g)^\alpha, g^a, SE, HA_1, HA_2, HA_3)$ as PK and (α, a) as MSK. It also initializes $IT = \varnothing$.

Encrypt. Given the PHI data $M \in \aleph$ and a LSSS policy $\mathbb{T} = (T, \rho) \in (\mathbb{Z}_p^{m \times n}, \rho : [m] \to \mathbb{Z}_p)$, the encryption algorithm acts as follows.

Randomly select $\chi \in \mathbb{G}_1$ and $\overrightarrow{\varrho} = (s, \varrho_2, \ldots, \varrho_n)^\top \in \mathbb{Z}_p^{n \times 1}$. Calculate $\overrightarrow{\lambda} = T \cdot \overrightarrow{\varrho} = (\lambda_1, \lambda_2, \ldots, \lambda_n)^\top$. Choose $t_1, t_2, \ldots, t_m$ randomly from $\mathbb{Z}_p$ and compute

$$C = \chi \cdot e(g,g)^{\alpha s},$$

$$C_1 = g^s,$$

$$C_2 = g^{as}.$$

For each $i \in [m]$, compute $C_{i,1} = w^{\lambda_i} v^{t_i}$, $C_{i,2} = (u^{\rho(i)}\hbar)^{-t_i}$ and $C_{i,3} = g^{t_i}$.

The ciphertext of χ is $CT_\chi = (\mathbb{T}, C, C_1, C_2, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i\in[m]})$.

After that, this algorithm sets $TAG_1 = HA_1(\chi)$ and computes a symmetric key $SEK = HA_3(\chi)$. Then it calls SE to create a ciphertext $CT_M = \textbf{SE-Encrypt}(M, SEK)$ and the verification key $VK = TAG_2 = HA_2(TAG_1 \parallel CT_M)$.

Finally, the ciphertext of PHI data $CT = (CT_\chi, CT_M)$ is uploaded to the eHealth CSS as well as VK.

KeyGen. Given a tuple $(id, S = \{AT_1, AT_2, \ldots, AT_k\} \subseteq \mathbb{Z}_p)$, this algorithm randomly selects $b, \delta, \beta, \beta_1, \ldots, \beta_k \in \mathbb{Z}_p$ and then calculates

$$KK = g^{\alpha/b(a+\delta)} w^\beta,$$

$$KK_1 = \delta,$$

$$L_1 = g^\beta,$$

$$L_2 = g^{a\beta}.$$

For each $\tau \in [k]$, it computes $KK_{\tau,1} = g^{\beta_\tau}$ and $KK_{\tau,2} = (u^{AT_\tau}\hbar)^{\beta_\tau} v^{-(a+\delta)\beta}$.

Finally, it outputs the private key for (id, S) as $TK = (S, KK, KK_1, L_1, L_2, \{KK_{\tau,1}, KK_{\tau,2}\}_{\tau\in[k]})$ and $DK = b$. Simultaneously, the tuple (id, δ) is added to IT.

Pre-Process. The physician can request the PHI ciphertext $CT = (CT_\chi, CT_M)$ and VK from the eHealth CSS, which will response by the elements C_1, C_2, CT_M, and VK while the other elements will be sent to the eHealth CDS.

Before calling the pre-decryption service, he/she processes the C_1, C_2, and TK by calculating $C_3 = C_1^{KK_1} C_2 = g^{(a+\delta)s}$ and $L_3 = L_1^{KK_1} L_2 = g^{(a+\delta)\beta}$.

Then $PCT_\chi = C_3$ and $PTK = (KK, L_3, \{KK_{\tau,1}, KK_{\tau,2}\}_{\tau\in[k]})$ are sent to the eHealth CDS.

Pre-Decrypt. Once receiving PCT_χ and PTK, this algorithm works as follows.

If S does not match $\mathbb{T}$, this algorithm aborts. Otherwise, it sets $I = \{i : \rho(i) \in S\}$ and calculates constants $\{\psi_i \in \mathbb{Z}_p\}_{i\in I}$ such that $\sum_{i\in I} \psi_i T_i = (1, 0, \ldots, 0)$, where T_i refers to the i-th row of T. Then it calculates

$$\begin{aligned} C' &= \frac{e(KK, C_3)}{\prod_{i\in I} \left(e(L_3, C_{i,1})\, e(KK_{\tau,1}, C_{i,2})\, e(KK_{\tau,2}, C_{i,3})\right)^{\psi_i}} \\ &= \frac{e(g,g)^{\alpha s/b}\, e(w,g)^{(a+\delta)\beta s}}{\prod_{i\in I} \left(e(g,w)^{(a+\delta)\beta\lambda_i}\, e(g,v)^{(a+\delta)\beta t_i}\, e(g,u)^{-t_i\rho(i)\beta_\tau}\, e(g,\hbar)^{-t_i\beta_\tau}\, e(u,g)^{t_i AT_\tau \beta_\tau}\, e(\hbar,g)^{t_i\beta_\tau}\, e(v,g)^{-(a+\delta)\beta t_i}\right)^{\psi_i}} \\ &= \frac{e(g,g)^{\alpha s/b}\, e(w,g)^{(a+\delta)\beta s}}{\prod_{i\in I} e(g,w)^{(a+\delta)\beta\lambda_i\psi_i}} = \frac{e(g,g)^{\alpha s/b}\, e(w,g)^{(a+\delta)\beta s}}{e(g,w)^{(a+\delta)\beta s}} = e(g,g)^{\alpha s/b} \end{aligned} \tag{2}$$

Finally, it outputs $PDC = C'$.

Decrypt. This algorithm first computes $\chi = C/(C')^b$. Then it calculates $TAG_1 = HA_1(\chi)$. If $HA_2(TAG_1 \parallel CT_M) \neq VK$, it aborts immediately. Otherwise, it calculates $SEK = HA_3(\chi)$ and recovers $M := \textbf{SE-Decrypt}(CT_M, SEK)$.

Trace. This algorithm first verifies whether TK and DK are well-formed by the following checks:

(1) TK is expressed as $(S, KK, KK_1, L_1, L_2, \{KK_{\tau,1}, KK_{\tau,2}\}_{\tau\in[k]})$, where $KK_1 \in \mathbb{Z}_p$ and $KK, L_1, L_2, KK_{\tau,1}, KK_{\tau,2} \in \mathbb{G}$.

(2) $DK = b \in \mathbb{Z}_p$.

(3) $e(g, L_2) = e(g^a, L_1)$.

(4) $e(KK, g^a \cdot g^{KK_1}) = e(g,g)^{\alpha/b} e(L_1^{KK_1} L_2, w)$.

(5) $\exists\tau \in [k]$, *s.t.* $e(KK_{\tau,2}, g) e(L_1^{KK_1} L_2, v) = e(KK_{\tau,1}, \hbar) e(KK_{\tau,1}, u)^{AT_\tau}$.

If TK and DK fail to pass the above five checks, it outputs $\top$. Otherwise, it searches KK_1 in IT: if KK_1 exists, it outputs the corresponding id. If KK_1 does not exist, it aborts.

5. Security Proof

5.1. CPA Security. For simplicity, the security of the presented VTCP-ABE scheme is reduced to that of the traceable scheme [17] which is proved under the φ-type assumption. We let $\sum_{TCP-ABE}$ and $\sum_{VTCP-ABE}$ denote the traceable scheme [17] and our VTCP-ABE scheme, respectively.

Theorem 7. *Suppose that $\sum_{TCP-ABE}$ is selectively secure, the one-time symmetric encryption scheme SE is semantically secure, HA_3 is chosen from a party of pairwise independent hash functions, and the parameters satisfy $0 < \ell_{SE} \leq (\log|\aleph| - \ell_{HA_1}) - 2\log(1/\epsilon_{HA_2})$. Then, the proposed $\sum_{VTCP-ABE}$ is selectively secure.*

Proof. Similar to the proof in [15], we define a series of hybrid argument of games as in [37].

Game$_0$. Identical to the original security game as defined in Section 3.3.

Game$_1$. Identical to **Game**$_0$, except that TAG_1^* and SEK^* are computed by selecting another random key R^* rather than χ^* in CT_χ.

Game$_2$. Identical to **Game**$_1$, except that we replace SEK^* by a randomly selected string $SEK_R^* \in \{0,1\}^{\ell_{SE}}$.

Let SPB_i be the success probability of the attacker in **Game**$_i$. □

Lemma 8. *If $\sum_{TCP-ABE}$ is selectively secure, then the attacker can not distinguish **Game**$_0$ from **Game**$_1$ with a non-negligible advantage.*

Proof. Suppose that an attacker $\mathscr{A}$ can distinguish **Game**$_0$ from **Game**$_1$, then we can build a PPT algorithm $\mathscr{B}$ to break $\sum_{TCP-ABE}$.

Init. The attacker $\mathscr{A}$ submits the challenge access policy $\mathbb{T}^*$ to $\mathscr{B}$. $\mathscr{B}$ then sends $\mathbb{T}^*$ to $\sum_{TCP-ABE}$.

Setup. Based on $\mathbb{T}^*$, $\sum_{TCP-ABE}$ gives $\mathscr{B}$ the parameter $PK_{TCP-ABE} = (G, g, u, \hbar, w, \nu, e(g,g)^\alpha, g^a)$ as in [17]. After that, $\mathscr{B}$ chooses SE^* and sets HA_1^*, HA_2^* and HA_3^* as random oracles. It also sets IT^*. Finally, it sends $PK^* = (PK_{TCP-ABE}, SE^*, HA_1^*, HA_2^*, HA_3^*)$ to $\mathscr{A}$.

Phase 1. To reply the key query of (id, S) from $\mathscr{A}$, $\mathscr{B}$ transmits (id, S) to $\sum_{TCP-ABE}$ and obtains $SK_{TCP-ABE} = (S, \widehat{KK}, \widehat{KK_1}, \widehat{L_1}, \widehat{L_2}, \{\widehat{KK_{\tau,1}}, \widehat{KK_{\tau,2}}\}_{\tau\in[k]})$, where

$$\widehat{KK} = g^{\alpha/(a+\delta)} w^{\hat{\beta}},$$
$$\widehat{KK_1} = \delta,$$
$$\widehat{L_1} = g^{\hat{\beta}},$$
$$\widehat{L_2} = g^{a\hat{\beta}}.$$
$$\forall \tau \in [k], \widehat{KK_{\tau,1}} = g^{\widehat{\beta_\tau}} \text{ and } \widehat{KK_{\tau,2}} = (u^{AT_\tau}\hbar)^{\widehat{\beta_\tau}} \nu^{-(a+\delta)\hat{\beta}}.$$

$\mathscr{B}$ randomly picks $b \in \mathbb{Z}_p$ and sets

$$KK = (\widehat{KK})^{1/b} = g^{\alpha/b(a+\delta)} w^{\beta},$$
$$KK_1 = \widehat{KK_1} = \delta,$$
$$L_1 = (\widehat{L_1})^{1/b} = g^{\beta},$$
$$L_2 = (\widehat{L_2})^{1/b} = g^{a\beta}.$$

□

For each $\tau \in [k]$, it computes $KK_{\tau,1} = (\widehat{KK_{\tau,1}})^{1/b} = g^{\beta_\tau}$ and $KK_{\tau,2} = (\widehat{KK_{\tau,2}})^{1/b} = (u^{AT_\tau}\hbar)^{\beta_\tau} \nu^{-(a+\delta)\beta}$.

$\mathscr{B}$ implicitly sets $\beta = (\hat{\beta})^{1/b}$ and $\beta_\tau = (\widehat{\beta_\tau})^{1/b}$.

Finally, $\mathscr{B}$ sends $TK = (S, KK, KK_1, L_1, L_2, \{KK_{\tau,1}, KK_{\tau,2}\}_{\tau\in[k]})$ and $DK = b$ to $\mathscr{A}$. Simultaneously, it adds (id, δ) to IT^*.

Challenge. $\mathscr{A}$ submits two equal-length messages M_0 and M_1, and $\mathscr{B}$ first picks two independent random keys χ^* and R^* from $\mathbb{G}_1$. It sends $((M_0^* = \chi^*, M_1^* = R^*), \mathbb{T}^*)$ to $\sum_{TCP-ABE}$. $\sum_{TCP-ABE}$ responds by a challenge ciphertext $CT_{M_\mu^*} = (\mathbb{T}^*, C, C_1, C_2, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i\in[m]})$. Then, $\mathscr{B}$ computes $SEK^* = HA_3(R^*)$ and $TAG_1^* = HA_1^*(R^*)$. It randomly picks $\mu \in \{0,1\}$ and calculates $CT_{M_\mu} =$ **SE-Encrypt**(M_μ, SEK^*). It also computes $VK^* = TAG_2^* = HA_2^*(TAG_1^* \parallel CT_{M_\mu})$.

Finally, it sends $CT^* = (CT_{M_\mu^*}, CT_{M_\mu})$ and VK^* to the attacker.

Note that, if the key encrypted under $\mathbb{T}^*$ in $CT_{M_\mu^*}$ is R^*, CT^* is regarded as a challenge ciphertext in **Game**$_0$. Otherwise, CT^* can be regarded as a challenge ciphertext in **Game**$_1$.

Phase 2. Similar to **Phase 1**.

Finally, $\mathscr{A}$ gives $\mathscr{B}$ a μ'. $\mathscr{B}$ then sends μ' to $\sum_{TCP-ABE}$. From the above game, we have $|\Pr[SPB_0] - \Pr[SPB_1]| \le 2ADV_{\mathscr{B}} \sum_{TCP-ABE}$.

Lemma 9. *Suppose that $\mathscr{H}$ is a family of pairwise independent hash functions, then **Game**$_1$ can not be distinguished from **Game**$_2$ with a non-negligible advantage.*

Proof. The key R^* is completely independent of PK, $CT_{M_\mu^*}$, and HA_3^* in both **Game**$_1$ and **Game**$_2$. Moreover, the number of possible values of $TAG_1^* = HA_1^*(R^*)$ is at most $2^{\ell_{HA_1}}$. According to the analysis in [15] and $0 < \ell_{SE} \le (\log|\aleph| - \ell_{HA_1}) - 2\log(1/\epsilon_{HA_2})$, the $SEK^* = HA_3(R^*)$ is ϵ_{HA_2}-statistically indistinguishable from the randomly selected $R_{SE}^* \in \{0,1\}^{\ell_{SE}}$. Hence, we have $|\Pr[SPB_1] - \Pr[SPB_2]| \le \epsilon_{HA_2}$. □

Lemma 10. *Suppose that SE is a semantically secure symmetric encryption scheme, then the attacker can not win **Game**$_2$ with a non-negligible advantage.*

Proof. In **Game**$_2$, $R_{SE}^* \in \{0,1\}^{\ell_{SE}}$ is a truly random symmetric key. An algorithm $\mathscr{B}$ can be directly constructed from $\mathscr{A}$ to break the semantic security of SE^*. Therefore, we have $|\Pr[SPB_1] - 1/2| \le ADV_{\mathscr{B}} SE^*$. □

Remark that **Game**$_0$ is identical to the selective security game for our proposed VTCP-ABE scheme. The advantage is $|\Pr[SPB_0] - 1/2|$. Thus, the security of our $\sum_{VTCP-ABE}$ follows.

5.2. Verifiability

Theorem 11. *Suppose that these two hash functions HA_1 and HA_2 are collision-resistant, our proposed VTCP-ABE scheme is privately verifiable.*

Proof. Suppose that an attacker $\mathscr{A}$ can win the verifiability game, we can employ $\mathscr{A}$ to build an algorithm $\mathscr{B}$ to break the collision-resistance of HA_1 and HA_2.

Given the challenge hash functions HA_1^* and HA_2^*, $\mathscr{B}$ processes as follows.

$\mathscr{B}$ runs **Setup** to generate PK and MSK, except for HA_1^* and HA_2^*. To answer the key queries, $\mathscr{B}$ acts as in **Phase 1** and **Phase 2**.

In the **Challenge** phase, $\mathscr{B}$ invokes the **Encrypt** to obtain the CT_{χ^*}. Then, it computes $TAG_1^* = HA_1^*(\chi^*)$ and $SEK^* = HA_3^*(\chi^*)$. It also calculates $CT_{M^*} =$ **SE-Encrypt**(M^*, SEK^*) and $VK^* = TAG_2^* = HA_2^*(TAG_1^* \parallel CT_{M^*})$. It sends $CT^* = (CT_{\chi^*}, CT_{M^*})$ and VK^* to $\mathscr{A}$.

$\mathcal{A}$ outputs an attribute set S^* which satisfies $\mathbb{T}^*$ and a partially decrypted ciphertext $PDC^* = C'$ and CT_M.

If $\mathcal{A}$ wins the verifiability game, $\mathcal{B}$ will get a message $M \notin \{M^*, \perp\}$. Note that the **Decrypt** algorithm outputs $\perp$ if $HA_2^*(TAG_1 \parallel CT_M) \neq TAG_2^*$, where $TAG_1 = HA_1^*(\chi)$ and χ is recovered from PDC^* and CT_M.

We now analyze the success probability of $\mathcal{A}$ by considering the following cases:

(1) $(TAG_1, CT_M) \neq (TAG_1^*, CT_{M^*})$. If this case happens, $\mathcal{B}$ gets a collision of HA_2^* immediately.

(2) $(TAG_1, CT_M) = (TAG_1^*, CT_{M^*})$, but $\chi \neq \chi^*$. Note that $HA_1^*(\chi^*) = TAG_1^* = TAG_1 = HA_1^*(\chi)$. Thus, $\mathcal{B}$ gets a collision of HA_1^*. □

5.3. Traceability

Theorem 12. *If the ϑ-SDH assumption holds, then our proposed VTCP-ABE scheme is fully traceable on condition that $q < \vartheta$, where q is the number of key queries made by the attacker $\mathcal{A}$.*

Proof. We here briefly introduce the traceability proof. Given $(G = (p, \mathbb{G}, \mathbb{G}_1, e), \overline{g}, \overline{g}^{\omega}, \overline{g}^{\omega^2}, \ldots, \overline{g}^{\omega^{\vartheta}})$, the simulator $\mathcal{B}$ has to generate a pair $(\delta, \overline{g}^{1/(\omega+\delta)}) \in \mathbb{Z}_p \times \mathbb{G}$ to solve the ϑ-SDH problem.

Setup. Assuming $\vartheta = q + 1$, $\mathcal{B}$ sets $D_i = \overline{g}^{\omega^i}$ for each $i \in [\vartheta]$ and randomly selects q distinct numbers $\delta_1, \ldots, \delta_q$ from $\mathbb{Z}_p^*$. It then sets $f(z) = \prod_{i=1}^{q}(z + \delta_i) = \sum_{i=0}^{q} a_i z^i$, where $\{a_i\}$ are the coefficients of $f(z)$. The simulator computes $g \leftarrow \prod_{i=0}^{q}(D_i)^{a_i} = \overline{g}^{f(\omega)}$ and $g^{\omega} \leftarrow \prod_{i=1}^{q+1}(D_i)^{\omega_{i-1}} = \overline{g}^{\omega \cdot f(\omega)}$. It then randomly picks $u, \hbar \in \mathbb{G}$ and $\alpha, \theta_1, \theta_2 \in \mathbb{Z}_p$. Finally, $\mathcal{B}$ sets $(G, g, u, \hbar, w = g^{\theta_1}, v = g^{\theta_2}, e(g,g)^{\alpha}, g^{\omega}, SE, HA_1, HA_2, HA_3, IT)$ as PK, where SE, HA_1, HA_2, HA_3, and IT are set as in the CPA game. It gives PK to $\mathcal{A}$.

Key Query. $\mathcal{B}$ answers the i-th query of (id_i, S_i) as follows.

$\mathcal{B}$ sets $f_i(z) = f(z)/(z+\delta_i) = \prod_{j=1, j\neq i}^{q}(z+\delta_j) = \sum_{j=0}^{q-1} b_j z^j$ and computes $\zeta_i \leftarrow \prod_{j=0}^{q-1}(D_j)^{b_j} = \overline{g}^{f_i(\omega)} = \overline{g}^{f(\omega)/(\omega+\delta_i)} = g^{1/(\omega+\delta_i)}$. Then $\mathcal{B}$ randomly selects $b, \beta, \beta_1, \ldots, \beta_k \in \mathbb{Z}_p$ and calculates TK by computing:

$$KK = (\zeta_i)^{\alpha/b} = g^{\alpha/b(\omega+\delta_i)} w^{\beta},$$
$$KK_1 = \delta_i,$$
$$L_1 = g^{\beta},$$
$$L_2 = g^{\omega\beta}.$$

For each $\tau \in [k]$, it computes $KK_{\tau,1} = g^{\beta_\tau}$ and $KK_{\tau,2} = (u^{AT_\tau}\hbar)^{\beta_\tau}((g^{\omega})^{\theta_2} \cdot v^{\delta_i})^{-\beta} = (u^{AT_\tau}\hbar)^{\beta_\tau} v^{-(\omega+\delta_i)\beta}$.

It gives TK and $DK = b$ to $\mathcal{A}$ and add (id_i, δ_i) to IT.

Key Forgery. $\mathcal{A}$ submits $TK_\star = (S, KK, KK_1, L_1, L_2, \{KK_{\tau,1}, KK_{\tau,2}\}_{\tau\in[k]})$ and $DK_\star = b$ to $\mathcal{B}$. $\Psi_{\mathcal{A}}$ refers to the event that $\mathcal{A}$ wins, i.e., $TK_\star$ and $DK_\star = b$ are well-formed and $KK_1 \notin \{\delta_1, \delta_2, \ldots, \delta_q\}$.

If $\Psi_{\mathcal{A}}$ happens, $\mathcal{B}$ writes $f(z)$ as $f(z) = \phi(z)(z + KK_1) + \phi_{-1}$ for some polynomial $\phi(z) = \sum_{i=0}^{q-1} \phi_i z^i$ and some $\phi_{-1} \neq 0 \in \mathbb{Z}_p$. Note that β in $TK_\star$ is unknown to $\mathcal{B}$. $\mathcal{B}$ then computes

$$\sigma \leftarrow (KK/L_1^{\theta_1})^{b/\alpha} = \overline{g}^{\phi(\omega)} \overline{g}^{\phi_{-1}/(\omega+KK_1)},$$
$$w_s \leftarrow (\sigma \cdot \textstyle\prod_{i=0}^{q-1} D_i^{-\phi_i})^{1/\phi_{-1}} = \overline{g}^{1/(\omega+KK_1)},$$
$$\delta_s \leftarrow KK_1 \bmod p \in \mathbb{Z}_p.$$

Since $e(\overline{g}^{\omega} \cdot \overline{g}^{\delta_s}, w_s) = e(\overline{g}, \overline{g})$, (δ_s, w_s) is the solution for the ϑ-SDH problem.

If $\Psi_{\mathcal{A}}$ does not happen, $\mathcal{B}$ randomly picks $(\delta_s, w_s) \in \mathbb{Z}_p \times \mathbb{G}$ as the solution.

As analyzed in [17], $\mathcal{B}$'s advantage is non-negligible in solving the ϑ-SDH problem. □

6. Performance Comparison

We here compare the performance of the VTCP-ABE scheme with the TCP-ABE scheme [17] and the VCP-ABE scheme [15] in the setting of key encapsulation, where the PHI data is encrypted by a symmetric encryption key χ which will be encrypted under an access policy in ABE.

6.1. Numeric Result. Tables 2 and 3 show the numeric comparison between our scheme and other two schemes [15, 17]. Let P, E, and E_1 be the overhead in executing a bilinear pairing, an exponential operation in $\mathbb{G}$ and $\mathbb{G}_1$, respectively. U denotes the system attribute universe. S_C, S_K, and I refer to the set of attributes used in encryption, key generation, and decryption, respectively. Let ℓ_{H_2} be the output length of H_2.

In Table 2, we calculate the computation cost incurred in the following phases: encryption, key generation, predecryption, and user decryption. The user in VCP-ABE and our VTCP-ABE expends constant size computation cost of exponential operation in $\mathbb{G}_1$. Note that our VTCP-ABE requires two additional exponential operations in the user side since that the ciphertext and transmission key need to be processed before being transmitted to the eHealth CDS.

In Table 3, the length of system public parameter, private key, and ciphertext is calculated by the number of group elements. The VCP-ABE scheme requires more public parameters which are linear with the scale of system attribute universe due to the fact that all the possible attributes need to be listed during the system initialization phase. Compared with the non-outsourced TCP-ABE scheme, our VTCP-ABE requires an additional element as the user decryption key and an output of H_2 as the verification key.

6.2. Implementation. We implement VCP-ABE scheme [15], TCP-ABE scheme [17], and the proposed VTCP-ABE on a windows 7 platform of an Intel(R) Core(TM) i5-3450 CPU at 3.10 GHz with 8.00 GB Memory. A Type A elliptic curve group is chosen from the JPBC library [38] and the order is a 512-bit prime. We mainly count the computation cost incurred by ABE relevant operations. The computation time of each algorithm is the average of 20 trials.

Figure 2 illustrates the computation cost comparison among VCP-ABE scheme, TCP-ABE scheme, and our proposed VTCP-ABE scheme.

Table 2: Computation cost comparison.

	Encryption	KeyGen	Pre-Decrypt	User Decrypt
VCP-ABE [15]	$(3\|S_C\|+1)E+1E_1$	$(\|S_K\|+3)E$	$\|I\|E_1+(2\|I\|+1)P$	$1E_1$
TCP-ABE [17]	$(5\|S_C\|+2)E+1E_1$	$(4\|S_K\|+4)E$	\	$2E+\|I\|E_1$
VTCP-ABE	$(5\|S_C\|+2)E+1E_1$	$(4\|S_K\|+4)E$	$\|I\|E_1+(3\|I\|+1)P$	$+(3\|I\|+1)P$ $3E_1$

Table 3: The parameter length comparison.

	Public Parameter	Private Key	Ciphertext
VCP-ABE [15]	$(\|U\|+2)\|\mathbb{G}\|+1\|\mathbb{G}_1\|$	$(\|S_K\|+2)\|\mathbb{G}\|+1\|\mathbb{Z}_p\|$	$(2\|S_C\|+1)\|\mathbb{G}\|+1\|\mathbb{G}_1\|+\ell_{H_2}$
TCP-ABE [17]	$6\|\mathbb{G}\|+1\|\mathbb{G}_1\|$	$(2\|S_K\|+3)\|\mathbb{G}\|+1\|\mathbb{Z}_p\|$	$(3\|S_C\|+2)\|\mathbb{G}\|+1\|\mathbb{G}_1\|$
VTCP-ABE	$6\|\mathbb{G}\|+1\|\mathbb{G}_1\|$	$(2\|S_K\|+3)\|\mathbb{G}\|+2\|\mathbb{Z}_p\|$	$(3\|S_C\|+2)\|\mathbb{G}\|+1\|\mathbb{G}_1\|+\ell_{H_2}$

Figure 2(a) shows the computation time in the initialization phase. In the three schemes, the computation cost is mainly incurred by computing the parameters $e(g,g)^{\alpha}$ and g^{a}.

Figures 2(b) and 2(c) show the computation time in the key generation phase and the encryption phase, respectively. It is observed that the key generation cost and encryption overhead in three schemes are linearly with the number of used attributes. More precisely, TCP-ABE and ours require more computation operation than VCP-ABE since that the combination of parameters u and $\hbar$ is employed to indicate an attribute.

Figure 2(d) shows the computation cost in the pre-process phase of our VTCP-ABE. Two exponential and multiplicative operations in group $\mathbb{G}$ are required in computing C_3 and L_3 no matter how many attributes are involved.

Figure 2(e) illustrates the computation cost comparison in the user decryption phase among three schemes. We can find that the user decryption cost in TCP-ABE scheme increases with the number of attributes. Thanks to the efficient outsourced decryption approach, the final decryption costs on the user side in VCP-ABE scheme and ours are significantly lower than that in TCP-ABE and independent of the attribute number.

Figure 2(f) gives the computation cost comparison in tracing the malicious users between TCP-ABE and ours. We can observe that the computation cost in both scheme grows with the number of attributes and our scheme only requires one additional exponential operation in group $\mathbb{G}_1$.

7. Delegate Extension

If a physician is in trouble to connect to the eHealth CSS and CDS, he/she can delegate someone to download the PHI ciphertext from the CSS and request the partial decrypted ciphertext from the CDS. However, the access privilege of delegated user has to be restricted. Inspired by [20, 39, 40], we employ a verifiable random function to limit the access of delegated users to maximum ξ times and propose a verifiable and traceable CP-ABE scheme with key delegation (VTDCP-ABE).

Setup. Besides generating PK and MSK as in VTCP-ABE, this algorithm calculates $\Omega = e(g,g)$ and chooses a hash function $HA_4 : \{0,1\}^* \to \mathbb{Z}_p$. The public parameter is $PK_D = (PK, \Omega, HA_4)$.

The **Encrypt**, **KeyGen**, **Pre-Process**, **Pre-Decrypt**, and **Trace** algorithms are as well as that in VTCP-ABE.

Delegate KeyGen. Given a transmission key $TK = (S, KK, KK_1, L_1, L_2, \{KK_{\tau,1}, KK_{\tau,2}\}_{\tau\in[k]})$ of an id for a set S, an identity information id_D and a set $S_D = \{AT_1, AT_2, \ldots, AT_{\widehat{k}}\} \subseteq S$. This algorithm generates a delegated transmission key TK_D as follows.

Randomly select $d \in \mathbb{Z}_p$ and compute $\Gamma_{pse,1} = \Omega^{1/(d+HA_4(pse))}$, $\Gamma_{pse,2} = g^{1/(d+HA_4(pse))}$ and $\Gamma_{pse,3} = g^{d}$, where pse refers to the unique and random pseudonym of a delegated user. Set ξ as the maximum number of predecryption request that a delegated user can make.

Then compute

$$KK_{pse} = KK^{1/d} = g^{\alpha/bd(a+\delta)} w^{\beta/d},$$
$$KK_{pse,1} = KK_1 = \delta,$$
$$L_{pse,1} = (L_1)^{1/d} = g^{\beta/d},$$
$$L_{pse,2} = (L_2)^{1/d} = g^{a\beta/d}.$$

For each $\tau \in [\widehat{k}]$, compute $KK_{pse,\tau,1} = (KK_{\tau,1})^{1/d} = g^{\beta_\tau/d}$ and $KK_{pse,\tau,2} = (KK_{\tau,2})^{1/d} = (u^{A_\tau}\hbar)^{\beta_\tau/d}\gamma^{-(a+\delta)\beta/d}$.

The ξ-times delegated transmission key is set as

$$TK_D = \Big(\xi, pse, S_D, \Gamma_{pse,1}, \Gamma_{pse,2}, \Gamma_{pse,3}, KK_{pse}, KK_{pse,1}, L_{pse,1}, L_{pse,2}, \{KK_{pse,\tau,1}, KK_{pse,\tau,2}\}_{\tau\in[\widehat{k}]}\Big). \quad (3)$$

Delegate Pre-Process. The same as **Pre-Process**, the delegated user requests $CT = (CT_\chi, CT_M)$ from eHealth CSS and computes $C_{pse,3} = C_1^{KK_{pse,1}} C_2 = g^{(a+\delta)s}$ and $L_{pse,3} = (L_{pse,1})^{KK_{pse,1}} L_{pse,2} = g^{(a+\delta)r/d}$.

The delegated user then sends PCT_D and PTK_D to the eHealth CDS, where

$$PCT_D = (\mathbb{T}, C_{pse,3}, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i\in[m]})$$
$$\text{and } PTK_D = \Big(\xi, pse, S_D, \Gamma_{pse,1}, \Gamma_{pse,2}, \Gamma_{pse,3}, KK_{pse}, L_{pse,3}, \{KK_{pse,\tau,1}, KK_{pse,\tau,2}\}_{\tau\in[\widehat{k}]}\Big). \quad (4)$$

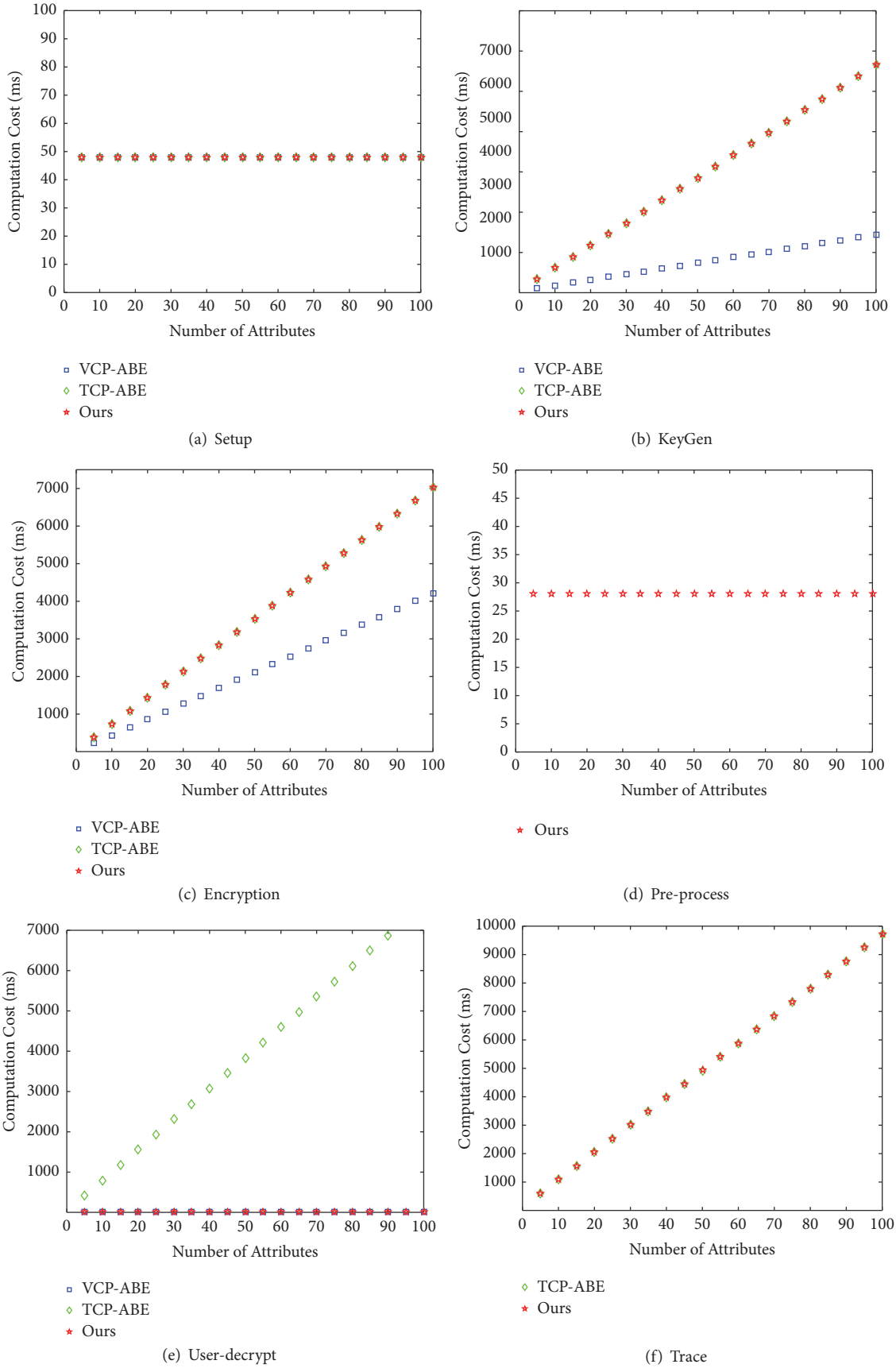


Figure 2: Comparisons of computation cost.

Delegate Pre-Decrypt. The eHealth CDS first initializes a counter $cou = 0$ and a set $S_{pse} = \{\Gamma_{pse,1}\}$ for each delegated user and stores the tuple (cou, S_{pse}) in a delegation list DL. Once receiving the Pre-Decrypt request from a delegated user, the CDS responds by the following way.

If S does not match $\mathbb{T}$, it outputs $\perp$.

Otherwise, it searches (cou, S_{pse}) in DL related to PTK_D and checks

(1) $e(g^{HA_4(pse)} \cdot \Gamma_{pse,3}, \Gamma_{pse,2}) = \Omega$ and $\Gamma_{pse,1} = e(g, \Gamma_{pse,2})$;

(2) $cou + 1 \leq \xi$;

(3) $\Gamma_{pse,1} \in S_{pse}$.

If the above three conditions do not hold, it aborts. Otherwise, it updates $cou \leftarrow cou + 1$ and computes the partial decryption ciphertext as

$$C'_D = \frac{e\left(KK_{pse}, C_{pse,3}\right)}{\prod_{i \in I}\left(e\left(L_{pse,3}, C_{i,1}\right) e\left(KK_{pse,\tau,1}, C_{i,2}\right) e\left(KK_{pse,\tau,q}, C_{i,3}\right)\right)^{\psi_i}} = e(g, g)^{\alpha s/bd} \quad (5)$$

Finally, the CDS responds the delegated user by $PDC_D = C'_D$. Then the delegated user gives PDC_D and CT_M to the physician.

Decrypt. If the physician interacts with the CSS and CDS directly, this algorithm acts exactly as in the **Decrypt** algorithm of VTCP-ABE. If the physician asks a delegated user to get the ciphertext and request the outsourced decryption service, χ is recovered by $\chi = C/(C'_D)^{bd}$. The verification and PHI decryption operations are identical to that of VTCP-ABE.

Since that the DK of physician and d are kept secretly, the delegated user can not obtain any content of the PHI ciphertext except a partial decrypted ciphertext.

8. Conclusion

In this paper, we have constructed a verifiable and traceable CP-ABE (VTCP-ABE) scheme for eHealth cloud applications, which also achieves the properties of large universe and delegation. With VTCP-ABE, the patient can enforce fine-grained access control over his/her PHI in a cryptographical way. Before submitting the encrypted PHI to the eHealth cloud decryption server, a pre-process on the ciphertext and transmission key is employed to preserve the identity privacy of the physician. The correctness of returned ciphertext can be efficiently verified. Moreover, the malicious physician who leaks the private key can be precisely tracked. Besides, we extend the proposed VTCP-ABE to support the delegation property, with which a resource-limited physician can authorize someone else to obtain a partial decrypted ciphertext without exposing the PHI content. The security of VTCP-ABE is proved in the selective model. The extensive experiments illustrate that our VTCP-ABE scheme efficiently achieves verifiability, traceability, and large attribute universe.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grants nos. 61502248, 61427801, u1405255, and 61602365, China Postdoctoral Science Foundation (Grant no. 2018M632350), and NUPTSF (Grant no. NY215008).

References

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT 2005*, R. Cramer, Ed., vol. 3494 of *Lecture Notes in Computer Science*, pp. 457–473, Springer Berlin Heidelberg, 2005.

[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 89–98, New York, NY, USA, November 2006.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, May 2007.

[4] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[5] J. Zhou, Z. Cao, X. Dong, and X. Lin, "TR-MABE: white-box traceable and revocable multi-authority attribute-based encryption and its applications to multi-level privacy-preserving e-healthcare cloud computing systems," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks (IEEE INFOCOM '15)*, pp. 2398–2406, April 2015.

[6] C. Hahn, H. Kwon, and J. Hur, "Efficient attribute-based secure data sharing with hidden policies and traceability in mobile health networks," *Mobile Information Systems*, vol. 2016, Article ID 6545873, 13 pages, 2016.

[7] Q. Li, J. Ma, R. Li, X. Liu, J. Xiong, and D. Chen, "Secure, efficient and revocable multi-authority access control system in cloud storage," *Computers & Security*, vol. 59, pp. 45–59, 2016.

[8] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Information Sciences*, vol. 379, pp. 42–61, 2017.

[9] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology—EUROCRYPT 2010*, H. Gilbert, Ed., vol. 6110 of *Lecture Notes in Computer Science*, pp. 62–91, Springer Berlin Heidelberg, 2010.

[10] K. Xue, Y. Xue, J. Hong et al., "RAAC: Robust and Auditable Access Control with Multiple Attribute Authorities for Public

Cloud Storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 953–967, 2017.

[11] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: a robust and verifiable threshold multi-authority access control system in public cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1484–1496, 2016.

[12] S. Wang, K. Liang, J. K. Liu, J. Chen, J. Yu, and W. Xie, "Attribute-based data sharing scheme revisited in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1661–1673, 2016.

[13] M. Green, S Hohenberger, and B. Waters, "Outsourcing the decryption of abe ciphertexts," in *Proceedings of the 20th USENIX Conference on Security (SEC '11)*, pp. 34-34, USENIX Association, Berkeley, CA, USA, 2011.

[14] J.-Z. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, 2013.

[15] B. Qin, R. H. Deng, S. Liu, and S. Ma, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1384–1393, 2015.

[16] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 76–88, 2013.

[17] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1274–1288, 2015.

[18] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*, pp. 463–474, New York, NY, USA, November 2013.

[19] S. Lin, R. Zhang, H. Ma, and M. Wang, "Revisiting attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2119–2130, 2015.

[20] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei, "Auditable sigma-time outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 94–105, 2018.

[21] B. Waters, *Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization*, vol. 6571 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2011.

[22] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, March 2010.

[23] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 456–465, New York, NY, USA, November 2007.

[24] J. Li, Q. Huang, X. Chen, S. S. M. Chow, D. S. Wong, and D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability," in *Proceedings of the 6th International Symposium on Information, Computer and Communications Security (ASIACCS '11)*, pp. 386–390, New York, NY, USA, March 2011.

[25] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant-size keys for lightweight devices," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 763–771, 2014.

[26] Q. Li, J. Ma, R. Li, J. Xiong, and X. Liu, "Large universe decentralized key-policy attribute-based encryption," *Security and Communication Networks*, vol. 8, no. 3, pp. 501–509, 2015.

[27] Q. Li, J. Ma, R. Li, J. Xiong, and X. Liu, "Provably secure unbounded multi-authority ciphertext-policy attribute-based encryption," *Security and Communication Networks*, vol. 8, no. 18, pp. 4098–4109, 2015.

[28] Q. M. Malluhi, A. Shikfa, and V. C. Trinh, "A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption," in *Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security (ASIA CCS '17)*, pp. 230–240, New York, NY, USA, April 2017.

[29] J. Ning, Z. Cao, X. Dong, K. Liang, L. Wei, and K. R. Choo, "CryptCloud+: secure and expressive data access control for cloud storage," *IEEE Transactions on Services Computing*, vol. 99, 2018.

[30] Z. Liu, Z. Cao, and D. S. Wong, "Blackbox traceable CP-ABE: how to catch people leaking their keys by selling decryption devices on eBay," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*, pp. 475–486, New York, NY, USA, November 2013.

[31] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," in *Financial Cryptography and Data Security*, R. Böhme and T. Okamoto, Eds., vol. 8975 of *Lecture Notes in Computer Science*, pp. 315–332, Springer Berlin Heidelberg, 2015.

[32] J. Ning, Z. Cao, X. Dong, J. Gong, and J. Chen, "Traceable CP-ABE with short ciphertexts: How to catch people selling decryption devices on ebay efficiently," in *Computer Security—ESORICS 2016*, I. Askoxylakis, S. Ioannidis, S. Katsikas, and C. Meadows, Eds., vol. 9879, pp. 551–569, Springer International Publishing, 2016.

[33] G. Yu, Z. Cao, G. Zeng, and W. Han, "Accountable ciphertext-policy attribute-based encryption scheme supporting public verifiability and nonrepudiation," in *Provable Security*, vol. 10005 of *Lecture Notes in Computer Science*, pp. 3–18, Springer International Publishing, Cham, Switzerland, 2016.

[34] A. Beimel, *Secure Schemes for Secret Sharing and Key Distribution [Mater Thesis]*, 1996.

[35] D. Boneh and X. Boyen, *Short Signatures Without Random Oracles*, vol. 3027 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2004.

[36] V. Goyal, *Reducing Trust in the PKG in Identity Based Cryptosystems*, vol. 4622 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2007.

[37] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs, 2004," shoup@cs.nyu.edu 13166 received 30 Nov 2004, last revised 18 Jan 2006.

[38] A. de Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications (ISCC '11)*, pp. 850–855, Kerkyra, Corfu, Greece, July 2011.

[39] D. Yevgeniy and A. Yampolskiy, "A verifiable random function with short proofs and keys," in *Public Key Cryptography—PKC 2005*, S. Vaudenay, Ed., pp. 416–431, Springer Berlin Heidelberg, 2005.

[40] T. H. Yuen, J. K. Liu, M. H. Au, X. Huang, W. Susilo, and J. Zhou, "k-times attribute-based anonymous access control for cloud computing," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 64, no. 9, pp. 2595–2608, 2015.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9437936, 13 pages
https://doi.org/10.1155/2018/9437936

Research Article

A Rational Exchange Protocol under Asymmetric Information in Wireless Sensor Networks

Zhen Lv,[1] Changgen Peng,[2] Yanguo Peng,[3] and Junwei Zhang[4]

[1]*Department of Basic Sciences, Air Force Engineering University, Xi'an 710051, China*
[2]*Guizhou Province Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China*
[3]*School of Computer Sciences and Technology, Xidian University, Xi'an 710071, China*
[4]*School of Cyber Engineering, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Yanguo Peng; ygpeng@xidian.edu.cn

Received 7 March 2018; Revised 22 April 2018; Accepted 30 April 2018; Published 31 May 2018

Academic Editor: Kim-Kwang Raymond Choo

P2P network is one of the most extensive network frameworks for wireless sensor network (WSN) in Internet of Things (IoT). The peers in WSN are rational and often free ride to save power of electricity and calculation, due to the fact that the usability is of great variability and unpredictability. Such a phenomenon tremendously reduces the quality of service (QoS) in WSN. Rational exchange protocol aims at promoting QoS and guaranteeing security and fairness. However, existing schemes have taken only complete information into account, which is not up to realistic environment. The peers in realistic environment indeed possess incomplete information, which is, however, still not thoroughly investigated so far. In this paper, under asymmetric information (a typical incomplete information), an entropy based incentive model is well designed based on Markov model and QoS evaluation model to help peers cooperate in WSNs. A concrete utility function with entropy is constructed to evaluate decision utility in P2P network. Finally, an entropy based rational exchange protocol is proposed based on the presented incentive model and concrete utility function, with analysis of correctness, security, fairness, and robustness, respectively. The proposed protocol can facilitate rational peers positively and sensibly participating in services and prevent free riding for rational peers. Hence, it further promotes QoS and guarantees security and fairness simultaneously in WSNs.

1. Introduction

Wireless sensor network (WSN) contains massive static and dynamic wireless sensors that extremely lack electricity and calculation. Due to that, sensors are highly willing to intentionally free ride [1–4] for better efficiency selfishly. Especially, in P2P network, which is one of the most extensive network frameworks for WSNs, peers (i.e., wireless sensors) are rational in fact. Rational peers make decisions to maximize their own benefit first and tremendously reduce the quality of service (QoS) of WSNs. To prevent free riding of rational peers and promote QoS, rational protocols have been widely investigated in security community [5–8].

Rational cryptography is a fresh and important branch in cryptographic research field, where rational exchange protocol is compatible and applicable for WSNs. Rational exchange protocol, which promotes peers to positively provide various services (data transmitting, data sharing, data distributing, etc.), aims at promoting QoS in WSNs. Simultaneously, rational exchange protocol guarantees both security and fairness for peers in WSNs, which are practical and necessary requirement in realistic environments.

Existing rational exchange protocols have been widely investigated since they were proposed by Syverson [9] in 1998. After that, extensive research work has been done under complete information [10–14], which means that each peer in WSN is fully aware of characteristics of participants, strategic space, and utility function about others. In realistic environments, however, the information is not always available (type, reputation, QoS, etc.), in which information is incomplete. A typical phenomenon is under asymmetric information [15], where, between two communicating peers, one possesses private information while the other is not aware of that. Unfortunately, rational exchange protocol under asymmetric

information is not well considered so far and should be investigated thoroughly.

We focus on, in this paper, proposing a rational exchange protocol with security and fairness under asymmetric information. Also, it is applied in wireless sensor networks aiming at promoting QoS of peers. The dominating contributions are as follows.

(1) To quantify utilities of participants, a dynamic game model with entropy is presented. In such a model, the utility for participants can be quantitatively analyzed. Furthermore, a selective basis of strategies in a game is explicitly provided (Section 2.1).

(2) Under asymmetric information, a Markov-based entropy function and a QoS evaluation model are designed. A utility function with entropy is further concreted to measure fairness of rational exchange protocol (Section 3).

(3) A concrete rational exchange protocol under asymmetric information is presented with through analysis of correctness, security, fairness, and robustness. Additionally, the proposed utility function with entropy is simulated and appropriateness is declared (Section 4).

The rest of this paper is organized as follows. Section 2 introduces the problem definition and preliminaries. Under asymmetric information, Section 3 presents a utility function with entropy based on Markov-based entropy function and QoS evaluation model. The concrete rational exchange protocol is proposed in Section 4 with thorough analysis. The simulation of utility function with entropy is presented in Section 5. Finally, the related work is summarized in Section 6 and this paper is concluded in Section 7.

2. Problem Definition and Preliminaries

In this section, we formally define a dynamic game model in which a utility function with entropy can be derived. Following that, a rational exchange protocol is formalized.

For facile understanding, the primary notations are listed in Table 1.

Table 1: Primary notations.

Notation	Meaning
$\mathbb{P}, P_i$	Set of participants and participant.
n	Number of participants in $\mathbb{P}$.
inf	Exchanging Information.
A	Set of optional actions.
$H(\bullet)$	Mixed strategy entropy function.
T	Participant's contribution value.
c	Unitary cost for participating in an exchange.
w_{ij}	Credit that participant P_j rates P_i.
Z	Participant's number of positive feedbacks.
U	Utility.
$\mathtt{T}$	Transfer matrix of participant's type.
$\mathtt{R}$	Possibility matrix of service for all participants.
PUB	Bulletin board.
$E(), D()$	Asymmetric encryption and decryption functions.
$\widehat{E}(), \widehat{D}()$	Symmetric encryption and decryption functions.
$\omega(x)$	Weakly secret bit commitment function.
PK, SK, k	Public key, private key, and session key.

2.1. Definition of Dynamic Game with Entropy. In this section, a dynamic game model with entropy is introduced to pave the way to quantify utilities of participants.

Definition 1 (dynamic game model with entropy). A dynamic game model with entropy contains a seven-tuple $\{\mathbb{P}, Q, (I_i)_{P_i\in\mathbb{P}}, A, f_p, (\geq_i)_{P_i\in\mathbb{P}}, (H_i(\bullet))_{P_i\in\mathbb{P}}\}$, where the elements are formally defined as follows.

(i) $\mathbb{P}$ is a set of participants, in which p_i is a participant.

(ii) Q is a set of sequences of actions containing all participants' actions. A sequence consists of participant's specific actions in a dynamic game. Q satisfies the following characteristics.

(1) The empty sequence is $\emptyset \in Q$.

(2) If a sequence of actions q satisfies that $q = (a_j)_{j=1}^{w} \in Q$ and $0 < v < w \in N^*$, then $q' = (a_j)_{j=1}^{v} \in Q$ holds.

(3) If any $v \in N^*$ and $q' = (a_j)_{j=1}^{v} \in Q$ holds, then the infinite sequence of actions $q = (a_j)_{j=1}^{\infty} \in Q$ holds.

(iii) $(I_i)_{P_i\in\mathbb{P}}$ is participant P_i's information set denoting the previous information of other participants in a game. An information set is formally denoted as $(I_i)_{P_i\in\mathbb{P}} = \{x_1, x_2, \ldots, x_m\}$, where x_i is the previous m action sequences of the other participants and the corresponding probability distribution is $\{p(x_1), p(x_2), \ldots, p(x_m)\}$ such that $\sum_{i=1}^{m} p(x_i) = 1$.

(iv) A is a set of optional actions, in which $A = \cup A_i$ and A_i is the optional actions of participant P_i.

(v) f_p is a participant function, to determine the next participant of nonterminal sequence of actions.

(vi) $(\geq_i)_{P_i\in\mathbb{P}}$ is a preference relation for participant P_i under a set of mixed strategies. Generally, the utility function $U_i(q)$ of participant P_i is the preference. The preference relation $q^* \geq_i q$ means that $U_i(q^*) \geq U_i(q)$ and q^* is the willing action sequence for participant P_i.

(vii) $(H_i(\bullet))_{P_i\in\mathbb{P}}$ is a mixed strategy entropy function for participant P_i. It is a probability distribution function of strategies on $(I_i)_{P_i\in\mathbb{P}}$. Given the round number r of a game, the entropy function of participant P_i in rth round is $H_i(r) = -\sum p(x_i) \log p(x_i)$, in which $H_i(\bullet) = \sum_{r=1}^{N} H_i(r) \geq 0$ is the total entropy of participant P_i during a whole game.

So far, we define a novel dynamic game model to analyze the rational exchange protocol that will be proposed in the following.

2.2. Definition of Rational Exchange Protocol. In this paper, rational exchange protocol is thoroughly investigated under two participants. Such a protocol can be facilely generalized into multiple participants in theory.

Definition 2 (rational exchange protocol with entropy). A rational exchange protocol with entropy contains a five-tuple $\{H(X \mid Y), P_A, P_B, PUB, \pi\}$, in which $H(X \mid Y)$ is a conditional entropy, P_A and P_B are the participants, PUB is a bulletin board, and π is the concrete exchange protocol. Additionally, π is formally defined as follows.

(i) *Setup*: generate the secret and public keys for P_A and P_B and other public parameters.

(ii) *Rational exchange*: participants P_A and P_B own information inf_A and inf_B, respectively. P_A and P_B rationally exchange their information, which means that P_A and P_B correctly get inf_B and inf_A, respectively, or get nothing.

A rational exchange protocol must satisfy the following four pivotal requirements.

Correctness. Such a requirement provides a guarantee that, when π is finished, P_A and P_B can successfully exchange inf_A and inf_B. Additionally, both P_A and P_B receive the positive feedbacks and credits that are defined later, in time.

Security. Both inf_A and inf_B must be prevented from adversary's cracking. That means there is no illegal participant that can derive anything from encrypted information during the exchange. Additionally, such a protocol can not reveal anything valuable during interactive processes.

Fairness. Since all participants are rational and aim at maximizing their own benefit, a rational exchange protocol is fair when the expected utilities for all the participants are maximized.

Robustness. The protocol is steady. That means even when the protocol is destabilized or interrupted, the fairness is still satisfied.

3. Utility Function with Entropy under Asymmetric Information

Before designing and representing a formal rational exchange protocol, a utility function capability for the dynamic game model with entropy is proposed in this section. Specifically, a Markov-based entropy function and a QoS evaluation model under asymmetric information are sequentially designed. Based on both of them, a utility function with entropy is concreted under asymmetric information.

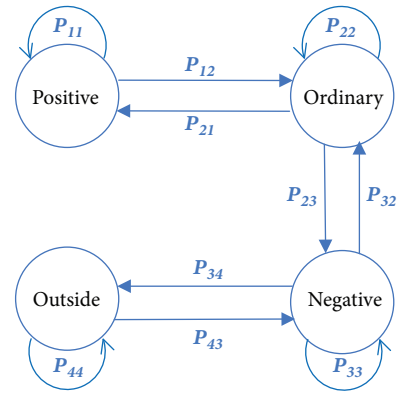


Figure 1: The transfer types for participants.

3.1. Markov-Based Entropy Function. Under asymmetric information, participant does not completely know, before exchanging information, about other participants' a priori information (i.e., participant's type). Additionally, participant's status relies only on the last past status. Hence, we introduce Markov chain to evaluate participant's information entropy.

Participants in our model are categorized into four types according to the contribution of participant in the system.

(i) *Positive*: such a participant provides service with a higher probability (e.g., $p \geq 80\%$). Here, positive participant's QoS is preferable. He can transfer to be ordinary.

(ii) *Ordinary*: such a participant provides service with a moderate probability (e.g., $50\% \leq p < 80\%$). He can transfer to be positive or negative.

(iii) *Negative*: such a participant provides service with a lower probability (e.g., $0\% < p < 50\%$). He can transfer to be ordinary or outside.

(iv) *Outside*: such a participant does not provide any service (e.g., $p = 0\%$) and does always free ride. He can transfer to be only negative.

In the transfer, each participant stays in a specific type with a specific probability and each transfer occurs with a specific probability too. The transfer between participants with different types is illustrated in Figure 1.

The constraint rules for participant's action are as follows.

(1) Allow participant to be outside when he first participates in a game.

(2) Every other period with time $t > 0$, the mechanism figures out the distribution of all participants' types.

By following the above constraints, every period, the probability R_{ij} for participant P_j serving participant P_i is estimated based on Markov chain. Based on the distribution in Step (2), the transfer matrix $\mathtt{P}_{4\times4}$ of participants' types can be resolved. Specifically, at jth period, the transfer matrix is $\mathtt{P}^{[j]} = \mathtt{P}(0) \bullet \mathtt{P}^j$, and the steady-state vector $\mathtt{D}$ is derived by resolving a system of linear equations $\mathtt{D} = \mathtt{D} \bullet \mathtt{P}$. Here,

$\mathtt{D} = (\mathtt{d}_i)_{i=1,2,3,4}$. The probability of service is $R_{ij} = \mathtt{D} \bullet \mathtt{g}^T$, where $\mathtt{g} = (\mathtt{g}_i)_{i=1,2,3,4}$ is the vector containing all membership functions corresponding to all participants' types (i.e., by embedding membership functions, the values of probabilities that participant provides service can be estimated in advance; thus, participant will be further convinced of his decision). In our construction, $\mathtt{g}_i$ $(1 \le i \le 4)$ stands for the tendency of maintaining a specific type (positive, ordinary, negative, and outside) for a participant. Finally, the probability matrix of service for all participants is $\mathtt{R}_{n\times n} = \{R_{ij}\}$, where $1 \le i, j \le n$.

According to the analysis of Shannon entropy and perceptive information, when a system achieves steady, P_i's total quantity of services coming from other participants is $H_i = -\sum_{j=1}^{n} R_{ij} \ln R_{ij}$, where $i \ne j$. From the aspect of expectation, H_i is larger; also P_i's willingness to participate in exchanging information is stronger. The expectation for the whole system is $H = \sum_{P_i \in \mathbb{P}} H_i$, which is changing with periods.

The ultimate entropy H_∞ of m-orderly discrete information source with memory is the m-orderly conditional entropy for a steady system. That means $H_\infty = \lim_{n\to\infty} H(X_n \mid X_1 \dots X_{n-1}) = H(X_{m+1} \mid X_1 \dots X_m) = H_{m+1}$, in which $H_{m+1} = -\sum_{i=1}^{n} \sum_{j=1}^{n} R_{ij} \ln R_{ij}$. The ultimate entropy H_∞ weighs the average quantity of information of all symbols that the information source sends. That means all the participants in an exchange protocol achieve steady.

3.2. QoS Evaluation Model under Asymmetric Information. Asymmetric information is a typically incomplete information. It means that, in an exchange protocol, a participant's information is complete, and the other's is incomplete. The former is advantageous participant, and the latter is disadvantageous participant. A rational participant makes decision through observing information that other participants reveal and deducing other participants' actions, in order to make his own expected utility maximum.

In this section, a deducing method is presented to turn asymmetric information into weakened-symmetric information. Based on that, a QoS evaluation model under asymmetric information is proposed for guaranteeing disadvantage and advantageous participants' fairness simultaneously.

For a disadvantageous participant (i.e., receiver) and an advantageous participant (i.e., sender) in a rational exchange protocol, the receiver derives a random variable Y about the exchanging information according to his own prior knowledge. The realistic information X is estimated by Y. In such a case, $H(X \mid Y)$ is the uncertainty of X on the condition that Y is known. Here, let $H(X)$ be the prior uncertainty and $H(X \mid Y)$ be the posterior uncertainty. The mutual information entropy is $H(X;Y) = H(X) - H(X \mid Y)$, which stands for the average quantity of information that is derived from Y about X. If $H(X \mid Y) \to 0$, then $H(X;Y) \to H(X)$ and asymmetry of two participants is removed. The random variable Y can be corrected according to Bayes rules with exchanging information. The transform process is shown in Figure 2.

Let $H_q(\bullet) = H(X)$ be the quantity of exchanging information that the receiver does not know. To overcome

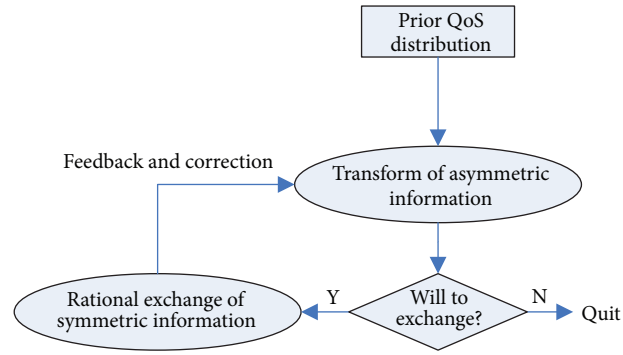


Figure 2: The transform process from asymmetric information to symmetric information.

the asymmetry of information, in fact, the receiver corrects service quantity by Bayes probability principle $p(x \mid y) = p(y \mid x)p(x)/p(y)$ with other participants' valuation about the exchanging information and derives the posterior probability distribution $\widehat{\pi}_q$.

Definition 3 (QoS evaluation mechanism). In a QoS evaluation mechanism, a list is maintained along with a whole rational exchange protocol. For participant P_i, the element in the list contains a three-tuple $\{P_i, Z_i, RP_i\}$. Here, P_i is the unique identifier, Z_i is the number of positive feedbacks, $RP_i = \cup_{l=1}^{m} \{A_{il}, \delta_{il}, (w_{ij})_{P_j \in \mathbb{P}\setminus\{P_i\}}\}$ is the set of records for lth round, in which A_{il} is the set of lth round actions, δ_{il} is the comment about quality of information in lth round, and $(w_{ij})_{P_j \in \mathbb{P}\setminus\{P_i\}}$ is the vector of credits for lth round information exchange.

Specifically, the list above is maintained at a bulletin board. After lth round exchange between sender P_S and receiver P_R, the following steps are carried out.

(1) P_R comments on such an exchange and uploads δ_{Sl} to the bulletin board. The bulletin board records actions in this exchange as A_{Sl}.

(2) P_R rates the quality of the exchanged information as w_{SR} and uploads it to the bulletin board.

(3) P_R can make a positive feedback by setting $Z_S = Z_S + 1$ if w_{SR} is greater than a threshold and a negative feedback by setting $Z_S = Z_S - 1$ if w_{SR} is smaller than a threshold.

(4) If there are multiple receivers $P_j \in \mathbb{P}\setminus\{P_R\}$ $(1 \le j \le m)$ in the system, P_j rates the quality of the exchanged information as w_{Sj} and makes a positive feedback for the other participants who make a consistent credit w_{Sj}.

By embedding such a QoS evaluation mechanism, all participants are further more willing to really evaluate the quality of exchanged information, in order to maximize credit and number of positive feedbacks. It is vital to measure participant's long-term utility.

3.3. Utility Function with Entropy. By comprehensive consideration of long-term utility and short-term utilities, the utility function with entropy is proposed for participant P_i.

$$U_i = -c_i T_i + H_i \left(\sum_{j=1}^{n} w_{ij} + \ln Z_i \right) T_i. \quad (1)$$

Here, c_i is the unitary cost for participating in an exchange and obtaining a unitary contribution value. $T_i = \alpha \sum_{j=1}^{n} R_{ij}$ is the contribution value in which the contribution weight α falls in $(0, 1)$.

In the equation, w_{ij} reflects short-term benefit and Z_i reflects long-term benefit for participant. In fact, in realistic scenarios, short-term benefit is more crucial than long-term benefit. In order to respond to such a correlation, it is necessary to take the logarithm of Z_i in the equation. By logarithm, the protocol is able to prevent participants with long-term benefit doing one-time deceive and provides rational fair from a new point of view. Through the combination of both, the utility function is more practical than others proposed in existing schemes.

Additionally, the product of information entropy H_i, which is the quantization of QoS, and $\sum_{j=1}^{n} w_{ij} + \ln Z_i$ constitutes the unitary revenue. Furthermore, T_i multiplied by $H_i(\sum_{j=1}^{n} w_{ij} + \ln Z_i)$ is P_i's revenue. According to the definition, $c_i T_i$ is the cost in total. Hence, the eventual utility for participant P_i is formalized and quantized by (1).

4. The Rational Exchange Protocol

In this section, the proposed concrete rational exchange protocol under asymmetric information is presented at the beginning. Following that, the detailed analysis is thoroughly stated. Additionally, theoretical comparisons between rational exchange protocols are represented to declare the presented rational exchange protocol's superiority.

4.1. Construction. Assume that there is no trusted third party in the proposed rational exchange protocol with multiple participants. That means each participant makes decision by maximizing his utility. Such a participant is rational and enjoys equal status, without considering compromising between participants.

In the concrete protocol, any two participants directly exchange information with each other. Assume that a sender P_A and a receiver P_B exchange inf_A and inf_B in the rational exchange protocol. Under asymmetric information, P_A is the advantageous participant and P_B is the disadvantage one. Additionally, inf_A is channel-sensitive and takes a long time t to be transmitted, but the time to transmit inf_B is short and negligible. There is also a bulletin board in the protocol to only manage all participants' account information and nothing else.

Setup. A weakly secret bit commitment function $\omega(x)$ is adopted, in which x can only be derived over time $t_0 > t$ since it is encrypted. An asymmetric encryption algorithm E and a symmetric encryption algorithm $\widehat{E}$ are adopted.

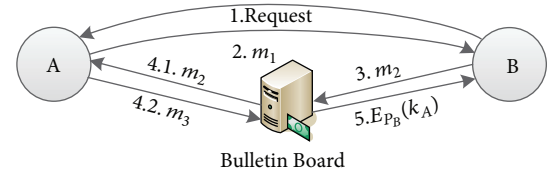


Figure 3: The concrete rational exchange protocol.

The corresponding decryption algorithms are D and $\widehat{D}$. According to the adopted asymmetric encryption algorithm E, $\langle PK_A, SK_A \rangle$ is generated for participant P_A as the public-private key pair, and $\langle PK_B, SK_B \rangle$ is generated for participant P_B too. Additionally, the bulletin board PUB is initialized by maintaining QoS evaluation mechanism with an empty list.

Rational Exchange. P_A and P_B rationally exchange information inf_A and inf_B by executing the following steps. The interactive processes for two participants are illustrated in Figure 3.

(1) The disadvantageous participant P_B calculates the conditional entropy $H(X \mid Y)$ according to the advantageous participant P_A's number of positive feedbacks Z_A. He makes decision by checking $H(X \mid Y)$. If he wishes to exchange information, then he continues to do the following steps. Otherwise, the protocol halts.

(2) P_A randomly chooses a session key k_A for $\widehat{E}$ and sends m_1 to P_B, in which $m_1 = E_{PK_B}(E_{SK_A}(B, \widehat{E}_{k_A}(inf_A)), B, \widehat{E}_{k_A}(inf_A))$.

(3) After receiving $m_1 = E_{PK_B}(Y_1, Y_2, Y_3)$, P_B decrypt m_1 and derive $B = Y_2$ and $\widehat{E}_{k_A}(inf_A) = Y_3$ by checking $D_{PK_A}(Y_1) = (Y_2, Y_3)$. If $D_{PK_A}(Y_1) = (Y_2, Y_3)$ does not hold, then P_B halts. Otherwise, P_B randomly chooses a session key k_B for $\widehat{E}$ and sends m_2 to PUB, in which $m_2 = E_{PK_A}(E_{SK_B}(A, \widehat{E}_{k_B}(inf_B), \omega(k_B)), A, \widehat{E}_{k_B}(inf_B), \omega(k_B))$. Such a process provides an evidence for possible disputation in future.

(4) A downloads $m_2 = E_{PK_A}(Y_1, Y_2, Y_3, Y_4)$ from the bulletin board PUB. P_A derives $A = Y_2$, $\widehat{E}_{k_B}(inf_B) = Y_3$, and $\omega(k_B) = Y_4$ by decrypting m_2. If $D_{PK_B}(Y_1) = (Y_2, Y_3, Y_4)$ does not hold, then P_A halts. Otherwise, P_A sends m_3 to PUB and P_B simultaneously, in which $m_3 = (\omega(k_B), E_{PK_B}(E_{SK_A}(B, k_A), B, k_A))$.

(5) After receiving $m_3 = (Y_1, E_{PK_B}(Y_2, Y_3, Y_4))$, if $D_{PK_A}(Y_2) = (Y_3, Y_4)$ does not holds, P_B sends m_3 and k_A to PUB. Then, PUB checks the validation of received information from P_B. If the information is valid, PUB marks inf_B invalid and rates a negative feedback for P_A according to the QoS evaluation mechanism proposed in Section 3.2. Additionally, the feedback information about P_A from other participants is also rated in a similar way.

According to the principle of long-term utility maximization for rational participants, rational participants will

not deceive others in a game. We analyze possible deceiving behaviors in the following.

Step (2). The possible deceiving is that P_A sends a fake inf_A. In such a case, in Step (5), P_B can verify the truth of inf_A and figure out the deceiving. Indeed, P_A can not deny such a deceiving. Meanwhile, $\omega(k_B)$ is still secretive and P_A still does not get inf_B. At the moment, P_B reports the deceiving of P_A to *PUB* and posts a negative feedback for P_A. *PUB* invalidates information inf_A and keeps a record. Other participants will deny the validation of inf_A. In this way, P_A does not get inf_B and indeed gets a serious negative feedback. In a long-term game, P_A will not send a fake inf_A ever.

Step (3). The possible deceiving, in this process, is that P_B sends a fake k_B. Due to the characteristics of $\omega(k_B)$, over time t_0, P_A can not derive inf_B using k_B. Other participants will also observe that P_B is deceiving and will not exchange anything else with P_B. In a long-term game, P_B's utility decreases dramatically, and hence he will not send a fake inf_B ever.

Step (4). P_A deceives P_B by not sending m_3 even over time t or sending a fake k_A. P_B can post a negative feedback for P_A and gain the supports from other participants. In such a case, P_A's utility is $U_A = -c_A T_A + H_A(\sum_{j=1}^{n} w_{Aj} + \ln Z_A)T_A$, in which $w_{AA} = 0$. Obviously, the loss of utility for P_A is great. Hence, P_A will not deceive in this process.

4.2. Theoretical Analysis. The correctness, security, fairness, and robustness of the proposed rational exchange protocol are analyzed, respectively, in this section.

Correctness. According to the concrete rational exchange protocol, the sender P_A finally receives inf_B and the corresponding rating w_{AB}. In one hand, the possible deceiving is that P_B send a fake session key $\widehat{k}_B$ to *PUB* in Step (3). In such a case, after a period t_0, all participants will observe the deceiving since $\omega(k_B)$ will be decrypted due to the inherent property of weakly secret bit commitment function. For long-term utility, P_B will also never do deceiving. On the other hand, due to the high relation between w_{AB} and Z_B, participant P_B will rate w_{AB} to acquire more positive feedbacks.

In the whole protocol, all participants are constrained by ratings and utility. In an exchange, specifically, Z_i is highly correlated to participant P_i's reputation and $H(X \mid Y)$ is prerequisite for participant to exchange information. For long-term utility, all participants are of high willingness to positively and honestly participate in an exchange. Hence, both Z_i and $H(X \mid Y)$ can prompt all participants to positively participate the protocol.

Hence, the protocol is correct.

Security. On one hand, security of the presented rational exchange protocol relies on the security of adopted asymmetric and symmetric encryption algorithms, which are assumed to be secure. Specifically, in Step (2), m_1 is in the encrypted form, whose security relies on the security of adopted encryption algorithms. It means that inf_A will not be revealed until the adopted encryption algorithms are breaking. In fact, there is no adversary that can break such algorithms. Hence, information in Step (2) is secure. Similarly, information in Step (3) and (4) is secure too.

Also, on the other hand, security relies on the interactive process during the whole protocol. In the following, the presented rational exchange protocol is proved to be secure under BAN logic [16]. For facile analysis, the presented rational exchange is formalized as follows. Note that Step (1) and (5) are not interactive and hence can be ignored in the analysis of security. However, it does not put any negative effect on security of the presented rational exchange protocol.

(2) $A \rightarrow B$: $\{\{\{inf_A\}_{k_A}\}_{SK_A}\}_{PK_B}$.
(3) $B \rightarrow A$: $\{\{\{inf_B\}_{k_B}, \omega(k_B)\}_{SK_B}\}_{PK_A}$.
(4) $A \rightarrow B$: $\{\{\omega(k_B), k_A\}_{SK_A}\}_{PK_B}$.

The following assumptions are obvious and reasonable in the rational exchange protocol.

$$A \mid\equiv \overset{SK_B}{\longmapsto} B$$

$$A \mid\equiv \overset{SK_A}{\longmapsto} A$$

$$A \mid\equiv \overset{PK_B}{\longmapsto} B$$

$$A \mid\equiv \overset{PK_A}{\longmapsto} A$$

$$A \mid\equiv \overset{PK_B}{\longmapsto} A$$

$$A \mid\equiv \overset{PK_A}{\longmapsto} B$$

$$A \mid\equiv \overset{k_A}{\longmapsto} A$$

$$A \mid\equiv \left(B \Longmapsto \{inf_B\}_{k_B}\right)$$

$$A \mid\equiv \#\{inf_B\}_{k_B}$$

$$B \mid\equiv \overset{SK_B}{\longmapsto} B$$

$$B \mid\equiv \overset{SK_A}{\longmapsto} A$$

$$B \mid\equiv \overset{PK_B}{\longmapsto} B$$

$$B \mid\equiv \overset{PK_A}{\longmapsto} A$$

$$B \mid\equiv \overset{PK_B}{\longmapsto} A$$

$$B \mid\equiv \overset{PK_A}{\longmapsto} B$$

$$B \mid\equiv \overset{k_B}{\longmapsto} B$$

$$B \mid\equiv \left(A \Longmapsto \{inf_A\}_{k_A}\right)$$

$$B \mid\equiv \#\omega(k_B)$$
$$B \mid\equiv \#\{inf_A\}_{k_A}. \tag{2}$$

Theorem 4 (security). *Under specific assumptions listed above, the presented rational exchange protocol is secure under BAN logic. Specifically, there are four potential objectives: ① $A \mid\equiv \{inf_B\}_{k_B}$; ② $B \mid\equiv \{inf_A\}_{k_A}$; ③ $A \mid\equiv \omega(k_B)$; and ④ $B \lhd k_A$.*

If the protocol is secure under BAN logic, when the protocol is completely finished, all objectives are concluded. Additionally, after Step (2), objective ② is concluded. Furthermore, after Step (3), objectives ① and ③ are additively concluded.

Proof of Theorem 4. For Step (2),

$$\frac{B \lhd \left\{\left\{\{inf_A\}_{k_A}\right\}_{SK_A}\right\}_{PK_B}, B \mid\equiv\overset{SK_B}{\longmapsto} B}{B \lhd \left\{\{inf_A\}_{k_A}\right\}_{SK_A}}$$
$$\frac{B \lhd \left\{\{inf_A\}_{k_A}\right\}_{SK_A}, B \mid\equiv\overset{PK_A}{\longmapsto} B}{B \lhd \{inf_A\}_{k_A}}$$
$$\frac{B \lhd \{inf_A\}_{k_A}, B \mid\equiv\overset{k_A}{\longmapsto} A}{B \mid\equiv A \mid\sim \{inf_A\}_{k_A}}$$
$$\frac{B \mid\equiv A \mid\sim \{inf_A\}_{k_A}, B \mid\equiv \#\{inf_A\}_{k_A}}{B \mid\equiv A \mid\equiv \{inf_A\}_{k_A}}$$
$$\frac{B \mid\equiv A \mid\equiv \{inf_A\}_{k_A}, B \mid\equiv \left(A \longmapsto \{inf_A\}_{k_A}\right)}{B \mid\equiv \{inf_A\}_{k_A}} \tag{3}$$

For Step (3),

$$\frac{A \lhd \left\{\left\{\{inf_B\}_{k_B}, \omega(k_B)\right\}_{SK_B}\right\}_{PK_A}, A \mid\equiv\overset{SK_A}{\longmapsto} A}{A \lhd \left\{\{inf_B\}_{k_B}, \omega(k_B)\right\}_{SK_B}}$$
$$\frac{A \lhd \left\{\{inf_B\}_{k_B}, \omega(k_B)\right\}_{SK_B}, A \mid\equiv\overset{PK_B}{\longmapsto} A}{A \lhd \left(\{inf_B\}_{k_B}, \omega(k_B)\right)}$$
$$\frac{A \lhd \left(\{inf_B\}_{k_B}, \omega(k_B)\right), A \mid\equiv\overset{k_B}{\longmapsto} B}{A \mid\equiv B \mid\sim \left(\{inf_B\}_{k_B}, \omega(k_B)\right)}$$
$$\frac{A \mid\equiv \#\{inf_B\}_{k_B}}{A \mid\equiv \#\left(\{inf_B\}_{k_B}, \omega(k_B)\right)}$$
$$\frac{A \mid\equiv B \mid\sim \left(\{inf_B\}_{k_B}, \omega(k_B)\right), A \mid\equiv \#\left(\{inf_B\}_{k_B}, \omega(k_B)\right)}{A \mid\equiv B \mid\equiv \left(\{inf_B\}_{k_B}, \omega(k_B)\right)}$$
$$\frac{A \mid\equiv B \mid\equiv \left(\{inf_B\}_{k_B}, \omega(k_B)\right), A \mid\equiv \left(B \longmapsto \{inf_B\}_{k_B}\right)}{A \mid\equiv \left(\{inf_B\}_{k_B}, \omega(k_B)\right)}$$
$$\frac{A \mid\equiv \left(\{inf_B\}_{k_B}, \omega(k_B)\right)}{A \mid\equiv \{inf_B\}_{k_B}, A \mid\equiv \omega(k_B)} \tag{4}$$

For Step (4),

$$\frac{B \lhd \left\{\left\{\omega(k_B), k_A\right\}_{SK_A}\right\}_{PK_B}, B \mid\equiv\overset{SK_B}{\longmapsto} B}{B \lhd \left\{\omega(k_B), k_A\right\}_{SK_A}}$$
$$\frac{B \lhd \left\{\omega(k_B), k_A\right\}_{SK_A}, B \mid\equiv\overset{PK_A}{\longmapsto} B}{B \lhd (\omega(k_B), k_A)} \tag{5}$$
$$\frac{B \lhd (\omega(k_B), k_A)}{B \lhd \omega(k_B), B \lhd k_A}$$

□

Fairness. When the protocol is finished, participant's utility consists of two parts. The first part comes from the exchanged information. Let U^+ and U^- be positive and negative utility, respectively. There is an assumption, in the protocol, that both inf_A and inf_B are equal-value. It means that both $U^+(inf_B) = U^+(inf_A)$ and $U^-(inf_A) = U^-(inf_B)$ hold. Hence, after exchanging information, P_A's total utility is $U^+(inf_B) + U^-(inf_A) = 0$ and P_B's total utility is $U^+(inf_A) + U^-(inf_B) = 0$.

The remainder part is the utility derived from the contribution value. Here, participant P_i's utility is $U_i = -c_i T_i + H_i(\sum_{j=1}^{n} w_{ij} + Z_i)T_i$. In this protocol, w_{AB} is P_B's credit on P_A's QoS. In a single interaction between P_A and P_B, for participant P_A, the larger w_{AB} is, the larger U_A is. Hence, P_A will positively promote QoS. Certainly, the larger Z_A is, the larger U_A is. Here, Z_A is the reference level for other participants that interacts with P_A in future. Obviously, payments are proportional to utility for all participants. Hence, the presented rational exchange protocol is rationally fair. The quantitative analysis of fairness is elaborated in Section 5.

In a word, in the proposed rational exchange protocol, participants positively participate in executing the protocol. In such a way, all participants can obtain maximum expected utility. Hence, the protocol satisfies fairness.

Robustness. In the presented rational exchange protocol in Section 4.1, only Steps (2), (3), and (4) may suffer destabilization or interruption.

(i) In Step (2), participant P_A can abort the protocol by sending nothing. In such a case, P_B receives nothing and will not be executing the following steps. So, the utility for P_A and P_B is 0. The fairness is satisfied.

(ii) In Step (3), after receiving m_1 sent by P_A, P_B can interrupt the protocol by not sending m_2. Now, P_B can

Table 2: Theoretical comparisons between rational exchange protocols.

Scheme	Information type	With entropy	Quantification of utility	Rational fairness
Syverson's [9]	Complete	✘	✘	✘
Buttyán's [10]	Complete	✘	✘	✘
Alcaide's [11]	Imperfect[1]	✘	✘	✔
Alcaide's [12]	Complete	✘	✘	✔
Ours	Asymmetric	✔	✔	✔

[1]Imperfect information is not suitable for realistic application scenarios.

not derive inf_A since the unawareness of k_A. Hence, P_B still knows nothing about inf_A and the protocol is still fair.

(iii) In Step (4), P_A can interrupt the protocol by not sending m_3 after receiving m_2 sent by P_B. In such a case, P_B can make an argument on this exchange, and *PUB* will mark inf_B invalid and rate a negative feedback for P_A according to the QoS evaluation mechanism proposed in Section 3.2. The negative feedback lowers Z_A and is a severe punishment for P_A. Simultaneously, P_B will also rate a low w_{AB}. Hence, P_A's utility $U_A = -c_A T_A + H_A(\sum_{j=1}^{n} w_{Aj} + \ln Z_A)T_A$ is inevitable lower than that before the exchange. Obviously, for long-term utility, P_A will never destabilize the protocol to maximize his own utility. Hence, the protocol is still fair.

In Steps (1) and (5), there is, obviously, no potential destabilization or interruption. Hence, in a word, the presented protocol is fair when destabilization or interruption occurs.

4.3. Theoretical Comparisons. Rational exchange protocol is a fresh and crucial branch in cryptographic research field. The greatest difference between our protocol and other related works is that related works have been investigated under complete (symmetric) information, while our protocol is very under asymmetric information. Theoretical comparisons are legibly declared in Table 2.

In rational exchange protocol, complete information is an ideal assumption which violates the requirements in realistic environment. Rational exchange protocol under imperfect information is first investigated in Alcaide's protocol [11]. The complete but imperfect information in [11] is, however, not suitable for the scenarios in Section 1. In this paper, rational exchange protocol under asymmetric information is carefully designed with clear definition and much more compatible for practice.

It is striking that the utility in existing rational exchange protocol can not be quantized so far, since the absence of entropy. Information theory is introduced in this paper to quantize all participants' utilities. Such a way is the first attempt to clearly evaluate participant's utility. Participant can further make accurate decisions by observing participants' utilities. Obviously, such a characteristic is more compatible and operable for real applications.

Rational fairness is an important characteristic for rational exchange protocol and has been attracting vastly attention in cryptographic community. Through analysis under a dynamic model with entropy in Section 4.2, rational fairness is guaranteed.

5. Simulation of Utility Function with Entropy

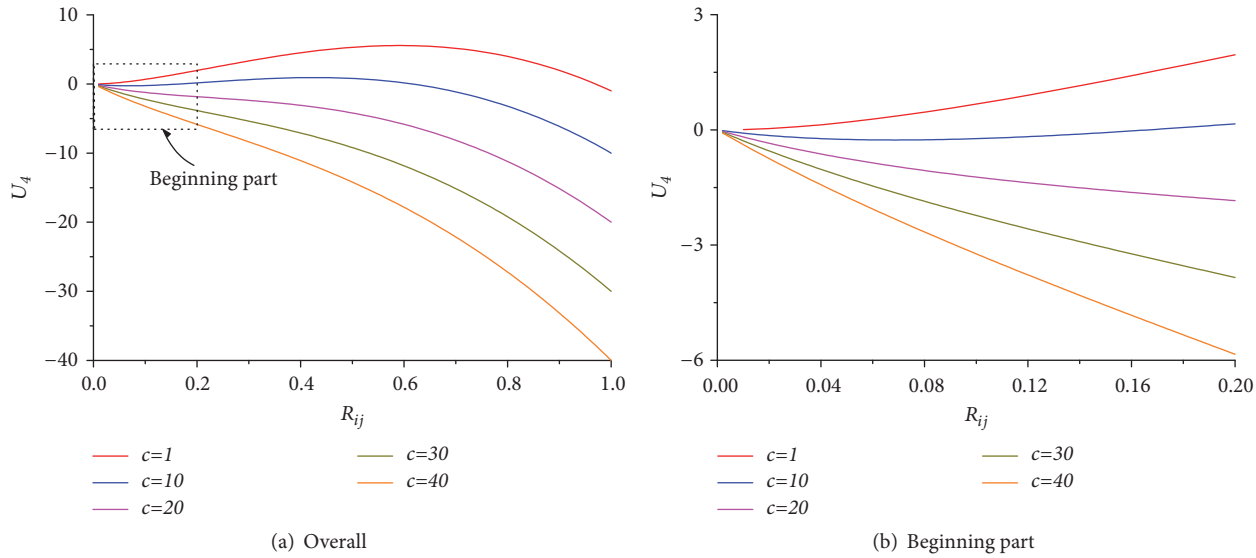
The utility function with entropy is kernel to guarantee the fairness of rational exchange protocol. In this section, through simulation, the proposed utility function with entropy is investigated in detail.

5.1. Simulation Environment. In the following simulation, the number of participants is assumed to be 10. When the system achieves stabilization after several rounds of information exchange, the current transfer matrix $\mathtt{P}$ of participants' types for each participant is assumed same and given as follows.

$$\mathtt{P} = \begin{bmatrix} 0.8 & 0.2 & 0 & 0 \\ 0.2 & 0.6 & 0.2 & 0 \\ 0 & 0.3 & 0.5 & 0.2 \\ 0 & 0 & 0.3 & 0.7 \end{bmatrix} \tag{6}$$

By resolving a system of linear equations $\mathtt{D} = \mathtt{D} \bullet \mathtt{P}$, the steady-state vector is $\mathtt{D} = \{0.32, 0.32, 0.22, 0.14\}$. The vector containing all membership functions is given as $\mathtt{g} = \{1.0, 0.75, 0.5, 0\}$, in which great $\mathtt{g}_i$ (i.e., closer to 1) indicates high-level accuracy of $\mathtt{D}_i$. Then, the probability of participant P_j serving P_i is $R_{ij} = \mathtt{D} \bullet \mathtt{g}^T = 0.67$. To evaluate participant's credits on the process of information exchange, all the vectors $(w_{ij})_{P_j \in \mathbb{P} \setminus \{P_i\}}$ of credits for participant P_i are generated randomly such that $w_{ij} \in (0, 1)$ and $w_{ii} = 0$. In a similar way, the vector Z_i of positive feedbacks is calculated through the above probabilities for all participants in the protocol.

The contribution value for participant P_i is $T_i = \alpha \sum_{j=1}^{n} R_{ij}$, in which $\alpha \in (0, 1)$. Here, let $\alpha = 0.1$, and $T_i = 0.1 \times (10 - 1) \times 0.67 = 0.603$ and $H_i = -\sum_{j=1}^{n} R_{ij} \ln R_{ij} = 2.41$ are derived. Given a unitary cost, the utility for each participant can be derived. All the values calculated in the above equations are listed below.



(a) Overall (b) Beginning part

FIGURE 4: The effectiveness on utility function while varying c_4 with $\alpha = 0.1$ for participant P_4.

The matrix W of credits is generated in the following.

$$\begin{bmatrix} 0.00 & 0.43 & 0.85 & 0.42 & 0.78 & 0.23 & 0.55 & 0.93 & 0.64 & 0.21 \\ 0.96 & 0.00 & 0.62 & 0.05 & 0.39 & 0.35 & 0.30 & 0.78 & 0.38 & 0.30 \\ 0.00 & 0.18 & 0.00 & 0.90 & 0.24 & 0.82 & 0.74 & 0.49 & 0.81 & 0.47 \\ \underline{0.77} & \underline{0.26} & \underline{0.51} & \underline{0.00} & \underline{0.40} & \underline{0.02} & \underline{0.19} & \underline{0.44} & \underline{0.53} & \underline{0.23} \\ \mathbf{0.79} & \mathbf{0.98} & \mathbf{0.83} & \mathbf{0.76} & \mathbf{0.00} & \mathbf{0.99} & \mathbf{0.83} & \mathbf{0.73} & \mathbf{0.78} & \mathbf{0.82} \\ 0.87 & 0.14 & 0.08 & 0.49 & 0.13 & 0.00 & 0.18 & 0.31 & 0.94 & 0.19 \\ 0.08 & 0.87 & 0.24 & 0.34 & 0.94 & 0.65 & 0.00 & 0.51 & 0.88 & 0.23 \\ 0.40 & 0.58 & 0.12 & 0.90 & 0.96 & 0.73 & 0.63 & 0.00 & 0.55 & 0.17 \\ 0.26 & 0.55 & 0.18 & 0.37 & 0.58 & 0.65 & 0.78 & 0.82 & 0.00 & 0.23 \\ 0.80 & 0.14 & 0.24 & 0.11 & 0.06 & 0.45 & 0.08 & 0.79 & 0.59 & 0.00 \end{bmatrix} \quad (7)$$

Assume that participant will respond to a positive feedback when the credit is greater than 0.60. Obliviously, the vector Z of positive feedbacks is simulated in the following.

$$Z = (4, 3, 4, \underline{1}, \mathbf{9}, 2, 4, 4, 3, 2). \quad (8)$$

The utility U for all participants is calculated in the following.

$$(9.28, 7.54, 8.71, \underline{4.81}, \mathbf{14.05}, 5.79, 8.84, 9.28, 7.96, 5.68). \quad (9)$$

By observing the utilities for all participants, participant P_5's utility is maximal and participant P_4's utility is minimal. It is consistent with the generated credits that P_5's credits are significantly greater than other participant's and P_4's credits are significantly smaller than other participant's. Hence, the presented utility function with entropy appropriately reflects the realistic utilities for all participants.

5.2. Effect of Parameters. The utility function with entropy $U_i = -c_i T_i + H_i(\sum_{j=1}^{n} w_{ij} + \ln Z_i)T_i$ is thoroughly investigated by observing P_4 and P_5's utilities by varying c and α, respectively.

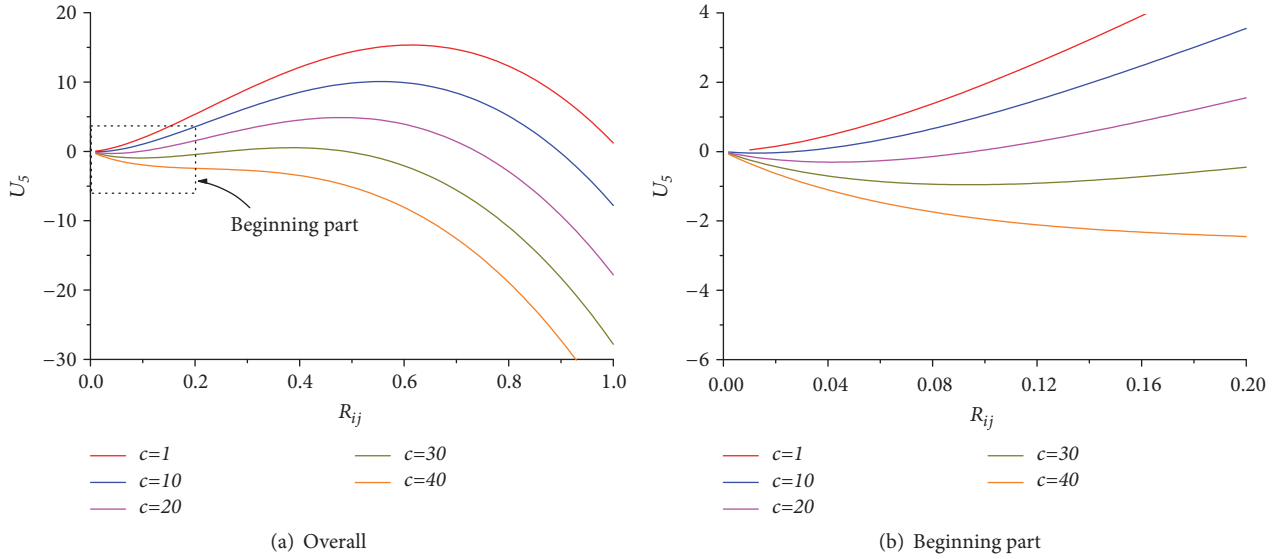
Effect of c. We vary $c \in \{1, 10, 20, 30, 40\}$ to exhibit the effectiveness on the presented utility function. The simulation is illustrated in Figures 4 and 5. Specifically, in Figure 4, Figure 4(b) illustrates the particulars of utility U_4's beginning tendency for participant P_4. In a similar way, Figure 5(b) illustrates the details of P_5's utility that is depicted in dotted rectangle in Figure 5(a). It is striking that participant's utility decreases at the beginning, increases in the middle stage, and decreases again at the end. In general, participant P_i's utility U_i is acquired in exchange with other participants. The trends of utility function confirm real-world scenarios and are elaborately described in the following content.

At the very beginning, participant P_j is considered to be outside indeed. Participant P_i's utility coming from P_j is assumed to be 0. During the first increasing of participant P_j's service probability, R_{ij} is extremely small and less than a threshold. Although the probability increases, the incrementation does not put any positive effect on U_i. P_i still, however, pays out due to the participation in the exchange. The utility coming from such an incrementation does not counteract the payment. Hence, U_i decreases in such a stage.

In the next stage, with the incrementation of P_j's service probability, the utility coming from such an incrementation goes beyond the payment. Hence, U_i increases along with the increasing of probability. Such a situation is compatible for practice and provides a guarantee of fairness for all participants.

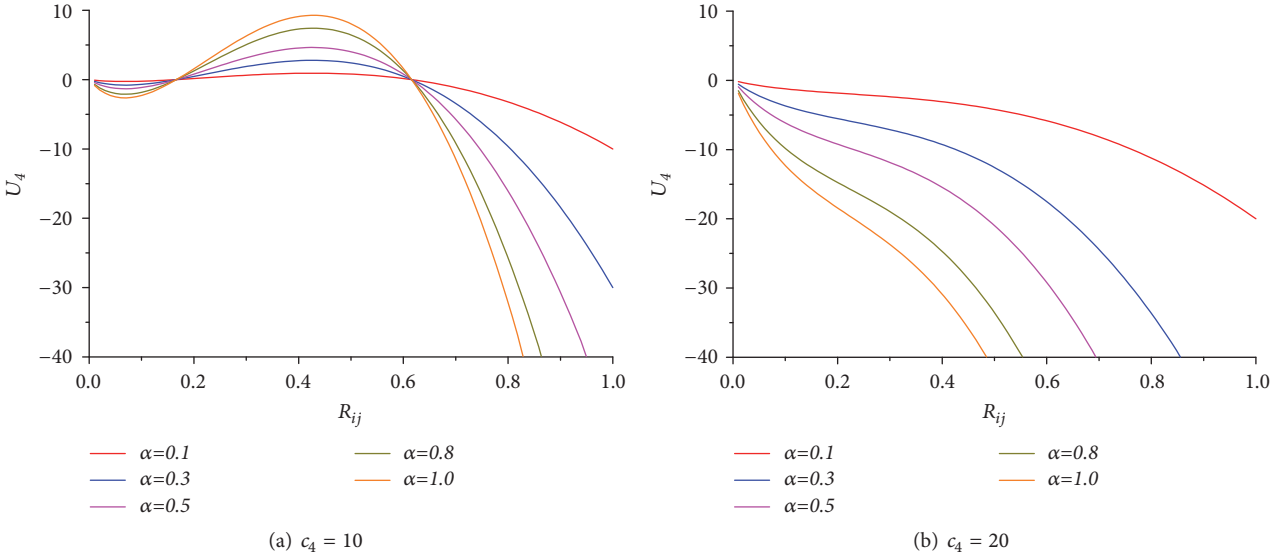
In the ending stage, if P_j's service probability is close to 1, other participants will free ride potentially. Due to that, participant P_i's credits and positive feedbacks will decrease. Following that, the utility U_i inevitably decreases in future. That is still fair for all participants.

Effect of α. We vary $\alpha \in \{0.1, 0.3, 0.5, 0.8, 1.0\}$ to exhibit the effectiveness on utility function. The simulation is illustrated in Figures 6 and 7. It is striking that the utility's tendency



(a) Overall (b) Beginning part

FIGURE 5: The effectiveness on utility function while varying c_5 with $\alpha = 0.1$ for participant P_5.



(a) $c_4 = 10$ (b) $c_4 = 20$

FIGURE 6: The effectiveness on utility function while varying α with $c_4 = 10$ and $c_4 = 20$ for participant P_4.

is consistent with that while varying c. Additionally, with greater α, the utility's discrimination is more prominent. In practice, α is determined according to consumer's realistic requirements.

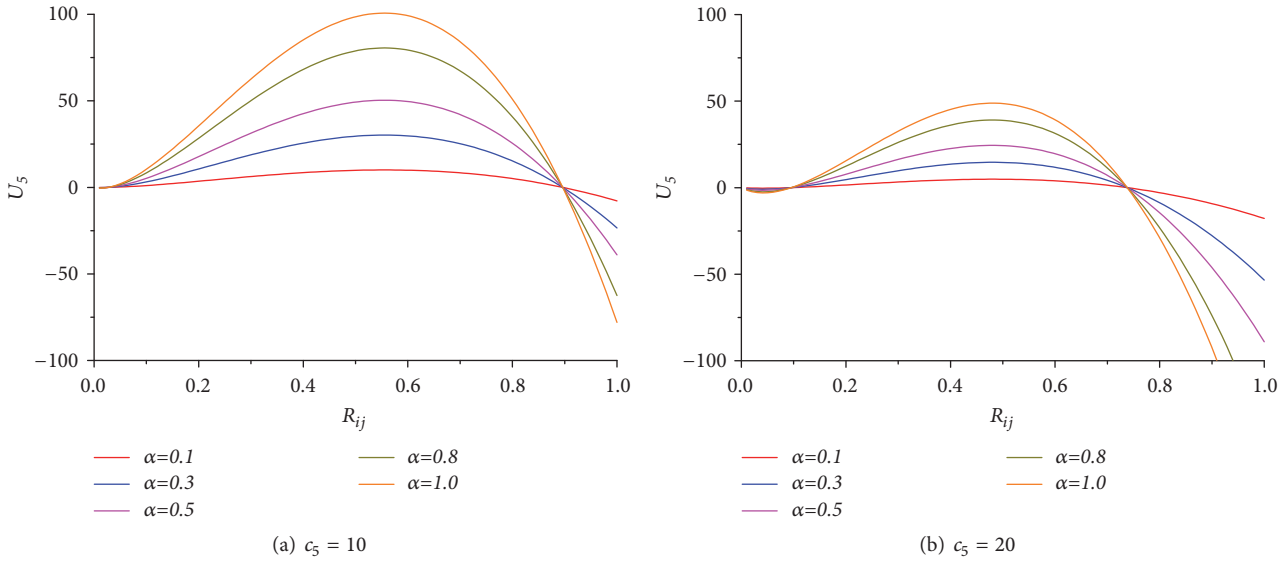
5.3. Performance for Participants. In this section, performance for all participants is elaborated in detail by varying contribution weight α and unitary cost c, respectively.

Varyingα. In the evaluation of performance, contribution weight α is varied in $(0.0, 1.0)$ in Figure 8(a). Obviously, with the increment of α, the discrimination between utilities for all participants is increasing significantly. The significant discrimination is important for practice, due to the fact that participant can facilely make decision without any doubt.

Wallrabenstein 2014 Varying c. In the evaluation of performance, contribution weight c is varied in $(0, 40)$ in Figure 8(b). Obviously, with the increment of α, the discrimination between utilities for all participants remains the same, due to the fact that a theoretical assumption that the probability of participant P_j serving P_i is R_{ij} is the same for all participants. Although the probabilities are completely different in practice, the discrimination remains the same since that the change of c only lowers the coefficient of T_i and does not change the monotonicity of utility in (1).

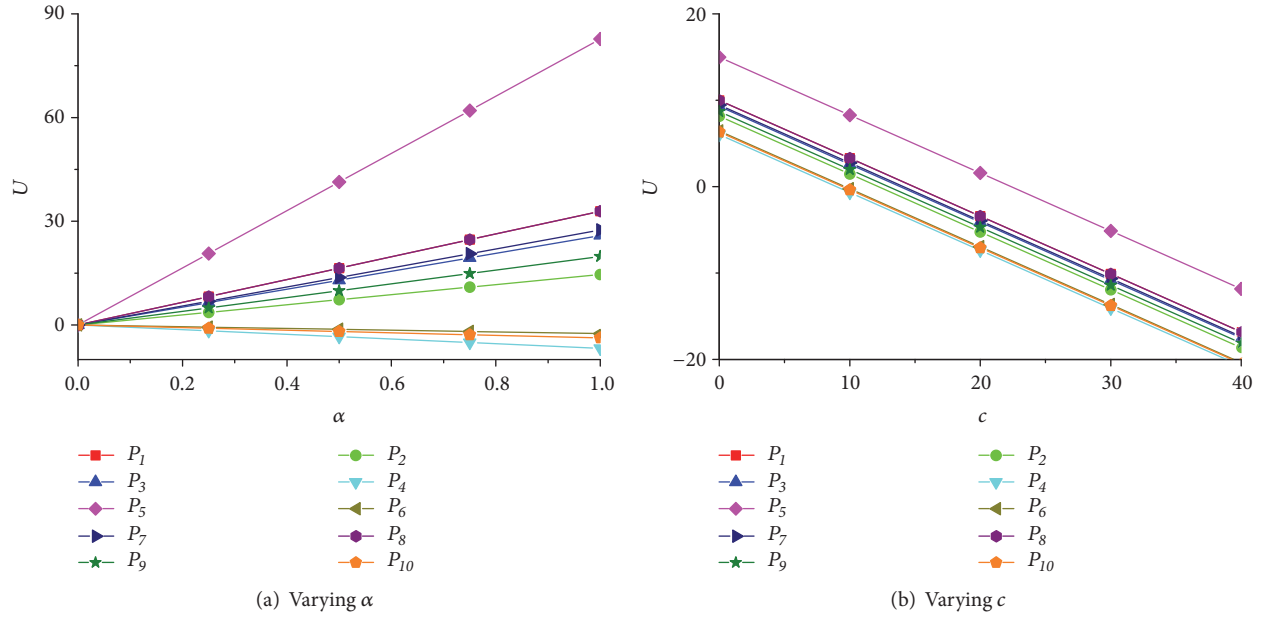
6. Related Work

Cryptography and game theory both concentrate on designing protocol, in which participants are with potential conflicts



(a) $c_5 = 10$ (b) $c_5 = 20$

FIGURE 7: The effectiveness on utility function while varying α with $c_5 = 10$ and $c_5 = 20$ for participant P_5.



(a) Varying α (b) Varying c

FIGURE 8: The performance of utility function with entropy for all participants. In Figure 8(a), α is varied with $c = 10$ and in Figure 8(b) c is varied with $\alpha = 0.1$.

of benefits [17, 18]. By combining both theories, rational secret sharing and rational secure multiparty computation are proposed by Halpern et al. [5] in 2004. In traditional cryptography, participants are assumed to be honest (strictly follow the protocol) or malicious (violate or destroy the protocol), and participants are however rational instead in practice for WSNs. That means all participants are highly willing to maximize their utilities by selecting their strategies during the whole protocol. That brings enormous challenge in designing protocols. Since the proposal of rational cryptography, there exist a number of studies [19, 20], such as rational secret sharing [21] and rational secure multiparty computation [5]. Indeed, rational exchange protocol, which guarantees security and fairness for peers in WSNs, is practical and necessary for realistic environments.

In 1998, Asokan [22] proposed an interactive protocol to exchange digital signature in a fair manner. It is the prototype of rational exchange protocol. The first real rational exchange protocol is proposed based on a weakly secret bit commitment function by Syverson [9] in the same year. In 2001, Buttyán et al. [10] analyzed the fairness of Syverson's protocol based on game theory, and, in 2004, they modeled rational exchange protocol and further analyzed the fairness of Syverson's protocol [13]. After that, Alcaide et al. modeled

rational exchange protocol based on extended game theory and Bayesian game in [11], improved Syverson's protocol in [23], and designed rational exchange protocol based on nature-inspired synthesis [12, 14]. However, all the above rational exchange protocols are constructed in environments with complete information, which is not perfectly compatible for scenarios mentioned in Section 1.

Rational participants possess asymmetric information about each other in real-world scenarios. Rational exchange protocol in such scenarios is not well investigated so far. By further combining information theory, rational exchange protocol is more practical and compatible for WSN. So far, information theory has been introduced into several cryptographic protocols [24, 25]. However, rational exchange protocol under asymmetric information is not thoroughly studied, which is striking in this paper.

7. Conclusion

In this paper, we presented a rational exchange protocol under asymmetric information, which is compatible and practical for WSNs. First of all, a dynamic game model with entropy is presented to quantify utilities of participants. In such a game, utilities for all participants can be quantized. Following that, a utility function with entropy is designed based on integrating a Markov-based entropy function and a novel QoS evaluation model. Furthermore, the concrete rational exchange protocol is presented with thorough analysis. Finally, the utility function with entropy is simulated. The simulation demonstrates effectiveness and availability of the presented rational exchange protocol.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant nos. 61702403, 61472298, 61662009, and 61472310), the Fundamental Research Funds for the Central Universities (Grant no. JB170308), the Project funded by China Postdoctoral Science Foundation (Grant no. 2018M633473), and the National Cryptography Development Foundation of China (Grant no. MMJJ20170129).

References

[1] M. Karakaya, K. Ibrahim, and U. Özgür, "Counteracting free riding in Peer-to-Peer networks," *Computer Networks*, vol. 52, no. 3, pp. 675–694, 2008.

[2] M. Karakaya, I. Korpeoglu, and Ö. Ulusoy, "Free riding in peer-to-peer networks," *IEEE Internet Computing*, vol. 13, no. 2, pp. 92–98, 2009.

[3] F. Malandrino, C. Casetti, and C.-F. Chiasserini, "Discovery and provision of content in vehicular networks," *Wireless Communications and Mobile Computing*, vol. 13, no. 3, pp. 244–254, 2013.

[4] H. Li, X. Liu, W. He, W. Yang, and W. Dou, "Delay analysis in practical wireless network coding," *Wireless Communications and Mobile Computing*, vol. 14, no. 5, pp. 497–515, 2014.

[5] J. Halpern and V. Teague, "Rational secret sharing and multiparty computation: extended abstract," in *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC 04)*, pp. 623–632, New York, NY, USA, 2004.

[6] G. Fuchsbauer, J. Katz, and D. Naccache, "Efficient rational secret sharing in standard communication networks," in *Theory of Cryptography*, D. Micciancio, Ed., pp. 419–436, Springer, Berlin, Germany, 2010.

[7] J. Garay, J. Katz, U. Maurer, B. Tackmann, and V. Zikas, "Rational protocol design: Cryptography against incentive-driven adversaries," in *Annual Symposium on Foundations of Computer SCIence—FOCS*, pp. 648–657, 2013.

[8] X. Liu, R. Deng, K. R. Choo, Y. Yang, and H. Pang, "Privacy-Preserving Outsourced Calculation Toolkit in the Cloud," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[9] P. Syverson, "Weakly secret bit commitment: applications to lotteries and fair exchange," in *Proceedings of the 11th IEEE Computer Security Foundations Workshop*, pp. 2–13, Rockport, MA, USA.

[10] L. Buttyán and H. Jean-Pierre, "Rational Exchange - A Formal Model Based on Game Theory," in *Electronic Commerce*, vol. 2232 of *Lecture Notes in Computer Science*, pp. 114–126, Springer Berlin Heidelberg, Berlin, Germany, 2001.

[11] A. Alcaide, J. M. Estevez-Tapiador, J. C. Hernandez-Castro, and A. Ribagorda, "An Extended Model of Rational Exchange Based on Dynamic Games of Imperfect Information," in *Emerging Trends in Information and Communication Security*, G. Müller, Ed., pp. 396–408, Springer, Berlin, Germany, 2006.

[12] A. Alcaide, J. M. Tapiador, J. C. Hernandez-Castro, and A. Ribagorda, "Nature–Inspired Synthesis of Rational Protocols," in *Parallel Problem Solving from Nature – PPSN X*, G. Rudolph, T. Jansen, N. Beume, S. Lucas, and C. Poloni, Eds., pp. 981–990, Springer, Berlin, Germany, 2008.

[13] L. Buttyán, J. Hubaux, S. Capkun, and S. Schneider, "A formal model of rational exchange and its application to the analysis of Syverson's protocol," *Journal of Computer Security*, vol. 12, no. 3-4, pp. 551–587, 2004.

[14] A. Alcaide, J. Estevez-Tapiador, J. Hernandez-Castro, and A. Ribagorda, "A multi-party rational exchange protocol," in *On the Move to Meaningful Internet Systems 2007:OTM2007 Workshops, ser. Lecture Notes in Computer Science*, R. Meersman, Z. Tari, and P. Herrero, Eds., vol. 4805, pp. 42-43, Springer, Berlin, Germany, 2007.

[15] S. Lauermann, "Asymmetric information in bilateral trade and in markets: an inversion result," *Journal of Economic Theory*, vol. 147, no. 5, pp. 1969–1997, 2012.

[16] M. Burrows, M. Abad, and M. Needham, "A logic of authentication," *The Royal Society A Mathematical, Physical and Engineering Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.

[17] P. Li, J. Li, Z. Huang et al., "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.

[18] J. Li, X. Chen, S. S. Chow, Q. Huang, D. S. Wong, and Z. Liu, "Multi-authority fine-grained access control with accountability and its application in cloud," *Journal of Network and Computer Applications*, vol. 112, pp. 89–96, 2018.

[19] J. Alwen, C. Cachin, J. B. Nielsen et al., "Summary report on rational cryptographic protocols," University of Aarhus, 2007.

[20] Z. Zhang and M. Liu, "Rational secret sharing as extensive games," *Science China Information Sciences*, vol. 56, no. 3, 032107, 13 pages, 2013.

[21] W. K. Moses and C. P. Rangan, "Rational secret sharing over an asynchronous broadcast channel with information theoretic security," *International Journal of Network Security and Its Applications*, vol. 3, no. 6, pp. 1–18, 2011.

[22] N. Asokan, *Fairness in electronic commerce [Ph.D. thesis]*, University of Waterloo, 1998.

[23] A. Alcaide, E.-T. U. M, H.-C. J. C, and A. Ribagorda, "Cryptanalysis of syversons rational exchange protocol," *International Journal of Network Security*, vol. 7, no. 2, pp. 151–156, 2008.

[24] U. Maurer, "Information-theoretically secure secret-key agreement by NOT authenticated public discussion," in *Advances in cryptology-EUROCRYPT '97 (Konstanz)*, W. Fumy, Ed., vol. 1233, pp. 209–225, Springer, Berlin, Germany, 1997.

[25] R. Renner and S. Wolf, "The exact price for unconditionally secure asymmetric cryptography," in *Advances in Cryptology-EUROCRYPT 2004*, C. Cachin and and J. L. Camenisch, Eds., pp. 109–125, Springer, Berlin, Germany, 2004.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7805408, 10 pages
https://doi.org/10.1155/2018/7805408

Research Article

Gleer: A Novel Gini-Based Energy Balancing Scheme for Mobile Botnet Retopology

Yichuan Wang, Yefei Zhang, Wenjiang Ji, Lei Zhu, and Yanxiao Liu

Xi'an University of Technology, Xi'an, China

Correspondence should be addressed to Yichuan Wang; chuan@xaut.edu.cn

Received 6 March 2018; Accepted 17 April 2018; Published 15 May 2018

Academic Editor: Kim-Kwang Raymond Choo

Mobile botnet has recently evolved due to the rapid growth of smartphone technologies. Unlike legacy botnets, mobile devices are characterized by limited power capacity, calculation capabilities, and wide communication methods. As such, the logical topology structure and communication mode have to be redesigned for mobile botnets to narrow energy gap and lower the reduction speed of nodes. In this paper, we try to design a novel Gini-based energy balancing scheme (Gleer) for the atomic network, which is a fundamental component of the heterogeneous multilayer mobile botnet. Firstly, for each operation cycle, we utilize the dynamic energy threshold to categorize atomic network into two groups. Then, the Gini coefficient is introduced to estimate botnet energy gap and to regulate the probability for each node to be picked as a region C&C server. Experimental results indicate that our proposed method can effectively prolong the botnet lifetime and prevent the reduction of network size. Meanwhile, the stealthiness of botnet with Gleer scheme is analyzed from users' perspective, and results show that the proposed scheme works well in the reduction of user' detection awareness.

1. Introduction

With the improvement of integrated-circuit technology, smartphones begin to provide better support for applications and services. Since users are increasingly exploiting smartphones for sensitive transactions (especially online shopping and banking), these mobile devices contain more sensitive and privacy information than legacy computer. Their unique features, including mobility, portability, and wide connectivity options, also play a significant role in promoting the rapidly growing popularity of smartphones. However, due to the lack of sufficient security and privacy protection mechanisms, smartphones are inevitably becoming the hot target of hackers. Among these threats from hackers, the mobile botnet [1] (evolved from traditional botnet) is the most destructive one, which not only steals user's privacy, but also attacks other network devices (e.g., DDOS). The motivation of this work is to shed light on potential botnet threats that are targeting smartphones. Since there are differences between computer and smartphone in many aspects, such as system structure and communication mode, existing techniques against computer botnets may not be applicable to mobile botnets. Thus, in the paper, we propose a Gini-based energy balancing scheme for defending mobile botnet attack, which can promote security researchers to investigate and develop new countermeasures before mobile botnets become a major threat [2].

The botnet life-cycle can be divided into six stages: botnet conception, botnet recruitment, botnet interaction, botnet marketing, attack execution, and attack success [3]. In the first stage, after confirming the motivation for creating a botnet, the following steps are processed for implementing the desired botnet. Various aspects should be carefully considered in this step, especially those regarding bot infection and botnet communications. From botmasters' perspective, improving the stealthy and robust of mobile botnet becomes the key objective in design process, while understanding the deployment strategy of a mobile botnet is critical for defending against malicious attacks on network in runtime from operators' perspective. Recently, there exist many literatures focused on mobile botnet communications design. Singh et al. [4] developed a mobile botnet on the basis of node popularity and leveraged publicly available data to demonstrate that the Bluetooth technology can be used as

C&C channel in a mobile botnet. Zeng et al. [5] proposed a mobile botnet that makes the most of mobile services and is resilient to disruption, utilizing SMS message for C&C and imitating the P2P fashion in the PC world. It is finally demonstrated that the structured architecture is a best choice for the mobile botnet topology. And in [6], Chen et al. proposed a novel multiple-push service based botnet, which significantly outperforms existing push-styled mobile botnet by exploring the design space of exploiting such services.

In the design process, the botnet architecture is the key decision, which determines the operation of subsequent element and thus the whole body of botnet [7]. The mobile botnet node, i.e., smartphone, constitutes the main component of mobile botnet architecture. The mobile botnet composed of smartphones is similar to sensor networks (SNs) in the architecture. (1) Topology design: their architecture can be centralized, distributed, or hybrid. Since the centralized scheme may encounter a single-point-of-failure, and distributed one is with high communication cost, most works on mobile botnet topology design are based on heterogeneous multilayer [8, 9], similar to cluster division of SNs. (2) Node management: due to the constrained power of smartphone and sensor, designing a node management scheme to reduce the node energy consumption and lowering the atomic network energy gap are in need. However, due to the fact that there exist some differences in constituent nodes and practical significance, the SNs management scheme cannot be applied to mobile botnet directly, such as low-energy adaptive clustering hierarchy (LEACH) and stable election protocol (SEP).

In view of the mobile botnet features and differences mentioned above, in this paper, we propose a region command and conquer (C&C) server selection scheme. At each start of operation cycle, we firstly divide smartphones of an atomic network into two categories according to their remaining capacity by the dynamic energy partitioning threshold. Then, the Gini coefficient is introduced to evaluate the power gap of nodes in the atomic network. Finally, the above coefficient values are leveraged to adjust and assign the selection probability to each type. Experimental results indicate that our proposed scheme can narrow the power gap of nodes in the atomic network and lower the reduction rate of network scale simultaneously. Meanwhile, the diversity of mobile network energy distribution at each experimental stage can lower the awareness of users effectively, which is beneficial to the concealment of mobile botnet. Therefore, the defensive strategies, which target the mobile botnet designed based on Gleer, should consider comprehensive factors instead of single factor like node power and use machine learning to explore potential insecurity features to detect the mobile botnet.

The rest of the paper is organized as follows: Section 2 describes the current researches on the mobile botnet topology and some cluster-head selection schemes for SNs; Section 3 shows the proposed C&C server selection scheme; Section 4 shows the experiment and result analysis; Section 5 presents the overall conclusions of this paper.

2. Related Work

2.1. Mobile Botnet Model. Different from legacy botnets, mobile botnets have to address new challenges from the unique features of mobile Internet and smartphones. More precisely, smartphones typically are with limited battery power, computation, and communication capabilities. If a bot consumes too much power, network traffic, or computation resource, then it will cause owner's awareness immediately. Many researches solve this problem by allocating hardware resources. For example, Chen et al. [10] proposed a novel cloud-based technology to overcome existing issues of mobile botnets. Meanwhile, due to the similarity between mobile botnets and SNs, we can also derive the network topology management scheme in SNs as follows. (1) The limited energy: despite being battery-operated, it may be unrealistic to recharge them due to either the inhospitable terrain for sensors or the high-mobility for smartphones. As such, the network lifetime maximization is of prime importance for SNs and mobile botnets [11]. (2) Cooperation [12]: they both involve a large number of nodes. Despite constrained capability of single node due to its limited energy capacity and communication capabilities, the collaboration among hundreds of nodes could offer unlimited possibilities. (3) High confidentiality: as SNs have been widely used in military, industrial, and health-care applications, the data transmitted among sensors are typically with significant values similar to the delivery in the mobile botnet for botnet controller. Thus, a secure communication with high confidentiality is perquisite [13].

It follows that SNs and mobile botnets share common goals: lifetime prolonging, coverage extension, seamless integration, and high reliability. Since data communication is basically an energy-intensive activity, the distribution of communication load among sensors contributes to their energy-consuming equilibration [14]. Currently, there are many existing works [15–19] with respect to clustering nodes to balance energy depletion and extending network lifetime further. Typically, nodes are divided into groups, and then a specific one is selected as the CH, similar to the hybrid scheme in the mobile botnet. In [20], a typical structure is designed, showed as in Figure 1, where the multilayer heterogeneous mode is favored, and the basic composition is atomic network which likes cluster in sensors. In Figure 1, the botmaster generates and sends commands to C&C servers, which distributes received commands to some region C&C servers further. Next, by analyzing each command, the region C&C server controls some bots to execute it, i.e., with a similar functionality to the cluster head (CH) in SNs.

2.2. Cluster-Head Selection Scheme. In the clustering procedure, selecting a CH for each cluster is of vital significance. Since CHs are responsible for aggregating reports from cluster members and then forwarding collected messages to the sink, they would consume more energy than non-CH nodes. For network retopology, the CH selection is always designed to be dynamic; e.g., LEACH [15] switches the CH role among nodes based on a prior optimal probability.

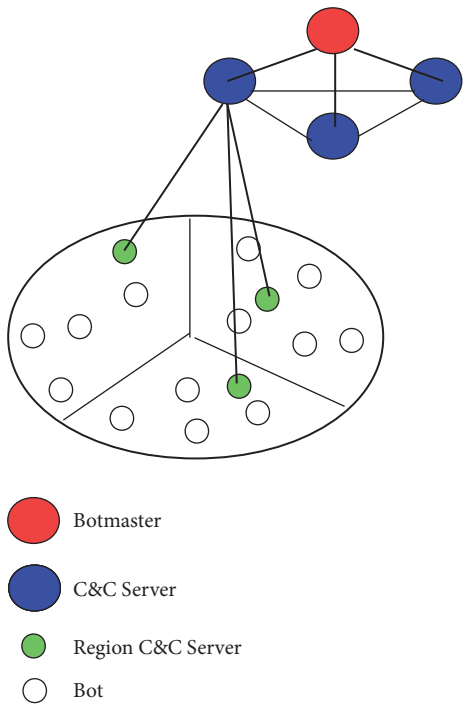


Figure 1: Mobile botnet logical topology.

Moreover, HEED [17] selects CH according to a hybrid of node residual energy and a secondary parameter, such as the distance between nodes or node degree. Furthermore, SEP [16] and P-SEP [18] are heterogeneity-aware protocols, introducing a fixed value α to represent the initial energy capacity difference ratio between two levels of deployed nodes.

To balance energy consumption, a cluster-head periodicity is favored in the atomic network. The selection used in SNs, nevertheless, does adapt to mobile botnet, with main factors as follows. (1) Mobile botnets need high stealthiness. The abnormal can be more easily awarded by a smartphone than a sensor; e.g., smartphone users are more sensitive to energy consumption. Thus, a fixed ratio (introduced in LEACH) between the remaining capacity and node selection probability cannot be applied to mobile botnets directly. Due to the difference between initial energy of nodes, a fixed value α introduced in SEP does not fit the mobile botnet scene. (2) Traditional CH selection schemes do not consider network energy gap, which may increase the network reduction rate. (3) There exist differences in component nodes and practical significance, so the selection scheme should necessitate a combination of their feathers. In view of these factors, we propose an energy balancing CH selection scheme based on Gini coefficient (Gleer). In particular, the energy partitioning threshold, i.e., α-value, is introduced, to categorize nodes of an atomic network into two groups: energy-sufficient node and energy-deficient node. Then, Gini coefficient is leveraged as an indicator to estimate botnet energy gap, as well as adjusting and optimizing the selection probability for each type.

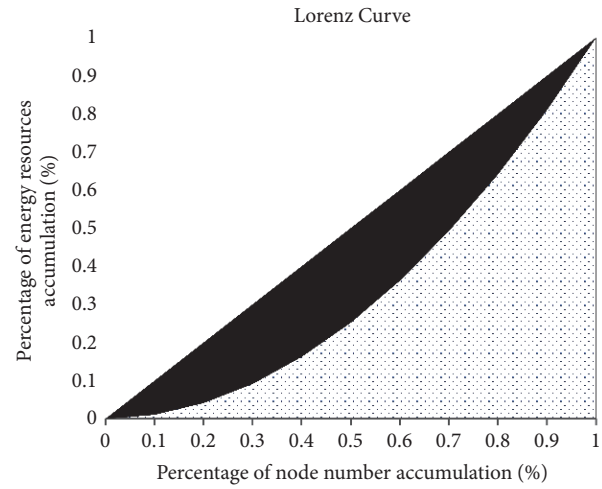


Figure 2: Lorenz Curve.

3. The Model

3.1. Gini Coefficient for Cluster-Head Selection. Gini coefficient was proposed by famous Italian economist C. Gini on the basis of the Lorenz curve in 1912 and has been deemed as a comprehensive index prevailing in the world showing the inequality degree of income distribution [21]. In this paper, it has been used as the main indicator widely. Figure 2 shows the Gini coefficient intuitively referring to the proportion of area A divided by A + B. In our work, we regard each node in an atomic network as a resident and its surplus energy as the current resident income and introduce the Gini coefficient as the measurement index for energy difference degree. As such, Figure 2 can be explained as follows: A is the area surrounded by actual Lorenz curve and energy absolute no difference curve, and A + B is the surrounded area by absolute no difference curve and absolute difference curve. Here, the Lorenz curve qualitatively reflects the energy difference degree roughly.

The calculation of Gini mean deviation was given as follows:

$$\Delta = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\left|x_j - x_i\right|}{n(n-1)}, \quad 0 < \Delta < 2u, \tag{1}$$

where Δ is the Gini mean deviation; $|x_j - x_i|$ is an absolute value of any pair of income sample deviation; n is the sample size; and u is the average income. Since Δ is obviously a monotone increasing function of income inequality, Gini stipulated

$$G = \frac{\Delta}{2u}, \quad 0 \le g \le 1 \tag{2}$$

as a measure of income inequality. According to (1) and (2), we can have

$$G = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\left|x_j - x_i\right|}{2n(n-1)u}. \tag{3}$$

Equation (3) offers a direct calculation method for Gini coefficient valued between 0 and 1, and the smaller, the more fair and the bigger the more unfair. Since this formula just involves the arithmetic operation of income data, this estimation method can be used unconditionally in theory without errors [22, 23]. Based on the Lorenz curve and the calculation method, a simple formula was put forward by Jianhua [24] as follows:

$$G = 1 - \frac{1}{n}\left(2\sum_{i=1}^{n-1} W_i + 1\right). \tag{4}$$

From the above calculation, it follows that n nodes are permuted from small to large according to their remaining energy, and W_i denotes the proportion of the accumulated energy of the first node to i node in all the nodes resources.

3.2. Proposed Scheme

3.2.1. Energy Partitioning Threshold. Considering the numerical diversity of smartphone residual capacity, we introduce an energy partitioning threshold α and divide nodes of an atomic network into two types, namely, energy-sufficient node and energy-deficient node, based on the comparison between α and the remaining energy of each node. To ensure that all nodes exhaust at approximately the same time, the nodes with more energy should be CHs more often than those with less energy, so the α-value should be set greater than energy mean. Suppose that $n \in N$ represents the number of smartphones with nonzero power. The remaining power of node i is E_i, $i = 1, \ldots, n$, and the value of partition coefficient $k \in [0, 1)$, $k \in R$ decides the range of α from the average to maximum energy of nodes; namely,

$$\alpha = \frac{\sum_{i=1}^{n} E_i}{n} + k \cdot \left(\max\left(E_i\right) - \frac{\sum_{i=1}^{n} E_i}{n}\right). \tag{5}$$

3.2.2. Classification and Statistics. Formula (6) is utilized to classify each node: $m_i = 1$ represents the remaining energy of node i which is larger than α; otherwise, the value is smaller than α. Table 1 shows the total number and energy for each node category.

$$m_i = \begin{cases} 1, & \dfrac{(E_i - \alpha) + |E_i - \alpha|}{2} > 0, \\ 0, & \dfrac{(E_i - \alpha) + |E_i - \alpha|}{2} = 0. \end{cases} \tag{6}$$

3.2.3. Regulating CH Selection Probability. Sort nodes based on their residual energy in ascending order and calculate Gini coefficient (G-value) of the current mobile botnet energy situation by Formula (4). According to the provisions of the relevant organization of United Nations [25], the Gini coefficient below 0.2 denotes the income absolute average; 0.2 to 0.3 denotes relative average; 0.3 to 0.4 denotes relatively reasonable; 0.4 to 0.5 denotes the income inequality relatively large; more than 0.5 denotes a huge income gap. Theoretically, there is a warning threshold, which is upper-bounded by 0.4 empirically; beyond 0.4, the society would be not in harmony, such as regional unbalance, rural and remote poverty, discrimination, hostility, crime, and environmental degradation [26, 27]. Since the surplus energy of each node is regarded as resident income, to balance node energy consumption and lower network size reduction rate, we use $G = 0.4$ as the dividing line to regulate the cluster-head selection probability following Table 2, where P_S and P_D, respectively, represent the total selection probability for each category.

Table 1: Classification and statistic for each category.

	Nodes number	Energy
Sufficient	$K_S = \sum_{i=1}^{n} m_i$	$K_S = \sum_{i=1}^{n} m_i$
Deficient	$K_D = n - K_S$	$E_{totD} = \sum_{i=1}^{n} \cdot E_i - E_{totS}$

Table 2: Total selection probability.

	P_S	P_D
$G < 0.4$	$\dfrac{E_{totS}}{E_{tot}}$	$\dfrac{E_{totD}}{E_{tot}}$
$G \geq 0.4$	$\dfrac{1}{2} + \dfrac{\lvert E_{totS} - E_{totD}\rvert}{2E_{tot}}$	$1 - P_S$

3.2.4. Calculating the Selection Probability for Each Node. To prevent the defender from detecting botnet and hide the nodes selection rule, the selection probability is identical between nodes of the same kind. The actual selection probability for each category is, respectively, represented by P_{SA}, P_{DA}.

$$\begin{aligned} P_{SA} &= \frac{P_S}{K_S} \\ P_{DA} &= \frac{P_D}{K_D}. \end{aligned} \tag{7}$$

4. Model Analysis

In [28], Tremblay et al. presented an easy-to-use battery model applicable to dynamic simulation. Figure 3 shows one of the typical lithium battery consumption curves for smartphones. Due to the similarity of smartphone users belonging to the same time zone, for example, most users are accustomed to charging smartphones fully at night and terminating charging between 6:30 and 8:00 a.m., therefore, if phones are approximately homogeneous, then they have almost the same capacity of lithium battery, application software, and usage habits. We can thus infer that the battery consumption curve of their smartphones is almost the same. Then, the remaining power of a smartphone at a certain time follows normal distribution. To analyze the performance of Gleer directly, we conduct experiments in MATLAB testbed and expand the experiment to atomic network with G-value larger than 0.4. The initial node number is represented by n.

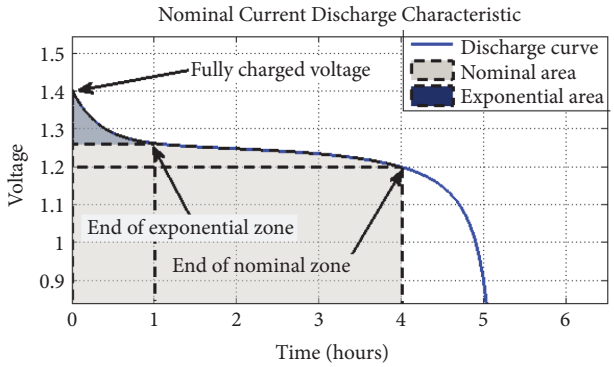


Figure 3: A typical discharge curve.

(1) Since CH nodes consume much more energy than normal ones, the atomic network should perform the CH selection process periodically to prevent the reduction of botnet size. Figure 4 shows the variation of Gini coefficient throughout the botnet lifetime. From Figure 4, it reveals that LEACH shows a significant increase, while Gleer is nearly stable. Meanwhile, it is obvious that Gleer outperforms LEACH in terms of lowering energy gap. That is, the proposed scheme could benefit the energy balance distribution of mobile botnet.

(2) Figure 5 represents the G-value versus the partition coefficient (k). As Figure 5(a) shows, when the initial sample and experiment conditions are identical, the energy distribution (G-value) almost remains unchanged with partition coefficient (k-value). Since k-value is an important parameter of Gleer scheme, the diversity of mobile botnet can prevent defenders cracking its internal rules and benefit the stealthiness of mobile botnet. Figure 5(b) also reveals the relationships between different initial conditions. From Figures 5(a) and 5(b), it follows that there exists a certain range for the multiexperiments with the same sample. However, boundaries emerge with different initial samples.

5. Simulation

NS-3 is an open-source and widely used network simulation platform [29] for networking research; it contains many models which simulate the packet data networks working and performing scene and provides a simulation engine for users to conduct simulation experiments. In this section, NS-3 platform is used to validate our model.

For simplicity, we only consider one cell as an atomic network with multiple nodes, and the base station is responsible for the connections between these nodes and the C&C server. Based on the models in [30] provided by NS-3, we build the network topology as shown in Figure 6. The battery model parameter for each node is from [28, 31], and Table 3 shows the detailed simulation configuration. The main task of this simulation is to select a node in the cell to be the region C&C server.

(1) Figure 7 shows mobile botnet performance in terms of energy node survival rate, energy consumption ratio, energy consumption ratio differences, and energy consumption ratio differences cumulative sum. With identical initialization (i.e., $n = 50$, G-value = 0.473), it can be observed from Figure 7(a) that the mobile botnet with Gleer scheme has larger lifetime than that with LEACH scheme. Since the proposed scheme involves the botnet energy distribution, it could benefit the prevention of botnet scale reduction caused by improper node selection. Figures 7(b), 7(c), and 7(d) show the energy consumption comparison; since Gleer contributes to the maintenance of botnet size and the lowering of number of death nodes, it can save much more energy and execute much more attacks with limited energy resources.

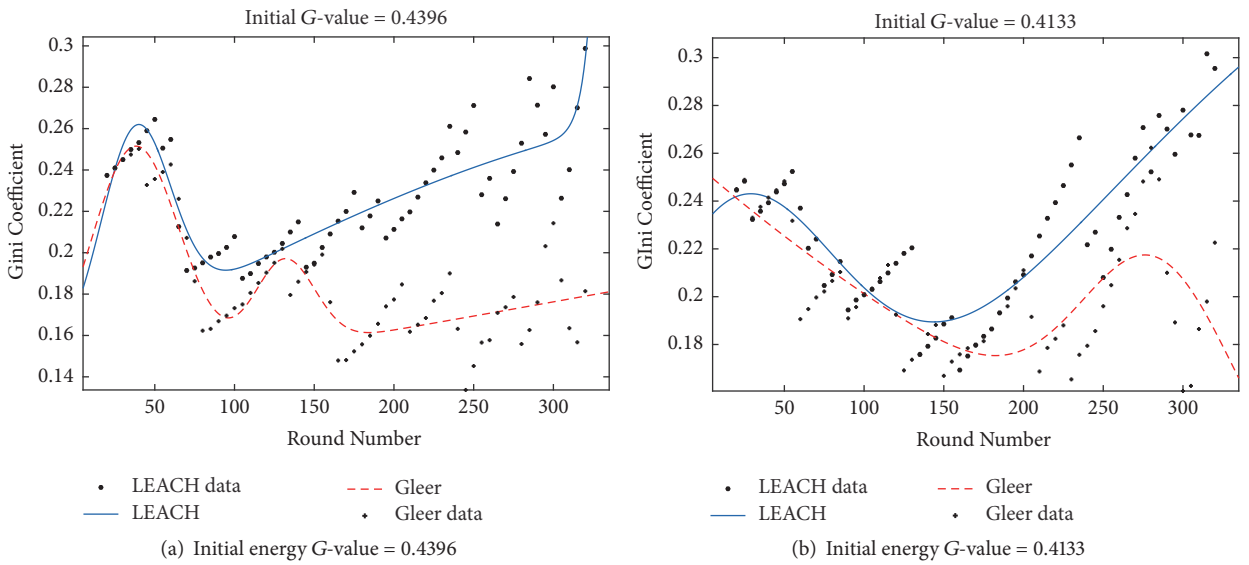
Table 3: NS-3 simulation configuration.

Name	Configuration
UE nodes	50
ENB nodes	1
Data rate between EPC and C&C server	100 G/s
Delay	0.01
App protocol	TCP
Packet size	1280
Simulated interval	2 s
Initial energy	100~31752 J
Initial Cell Voltage	3.45 A
Normal nodes current draw	2~4 A
Cluster nodes current draw	5~8 A
Nominal cell voltage	3.3 V
Exp Cell Voltage	3.6 V
Rated Capacity	2.45 Ah
Nom Capacity	1.1 Ah
Exp Capacity	1.2 Ah
Internal Resistance	0.145 Ohms

(2) For some abnormal cases, for example, strange SMS with a website URL linked to virus and emails with an attachment cheating, the intuition for smartphones users to judge whether a cell phone is safe is to observe the battery power variation. In order to show users' view obviously, the simulation is conducted six times with the same initial user group ($n = 50$, G-value = 0.437) and k-value ($k = 0.5$), and six users are picked randomly to observe their energy consumption. As Figure 8 shows, when the simulation reaches 30 s, there exists an energy bound for No. 4 user, ranging from $2.8 * 10^4$ J and $2.92 * 10^4$ J. For No. 25 user, the energy of six simulations is almost the same. And for No. 5 user, the time of arriving at the lowest energy varies. From these simulation results, it is observed that one user (infected by the mobile botnet with the proposed scheme) cannot judge whether his smartphone is abnormal or not, by analyzing the energy consumption. The energy consumption diversity across users shows that our proposed scheme can improve the stealthiness of mobile botnet effectively.

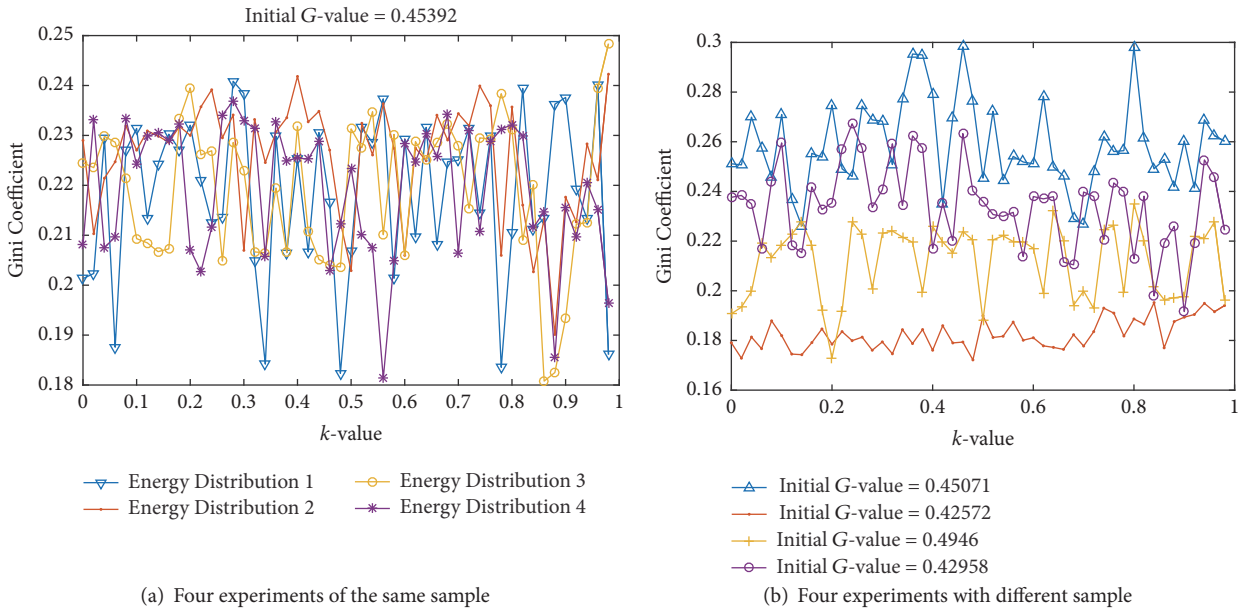
6. Defensive Strategies

As described in [32], smartphones not only provide capabilities of legacy computers, but also offer a large number of



(a) Initial energy G-value = 0.4396 (b) Initial energy G-value = 0.4133

Figure 4: The relationship between the G-value and round number.



(a) Four experiments of the same sample (b) Four experiments with different sample

Figure 5: The relationship between the G-value and k-value under different conditions.

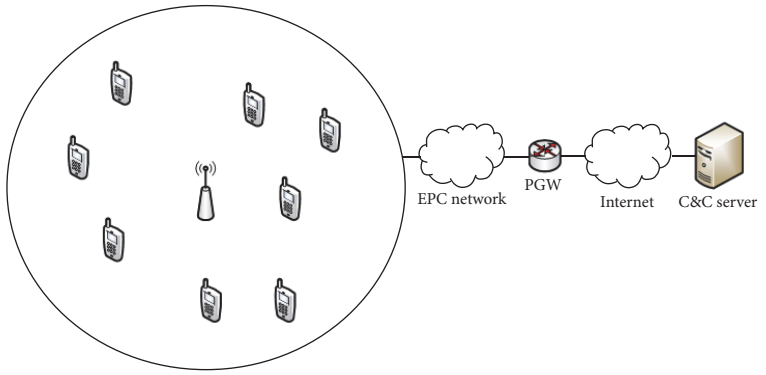


Figure 6: Simulation topology network.

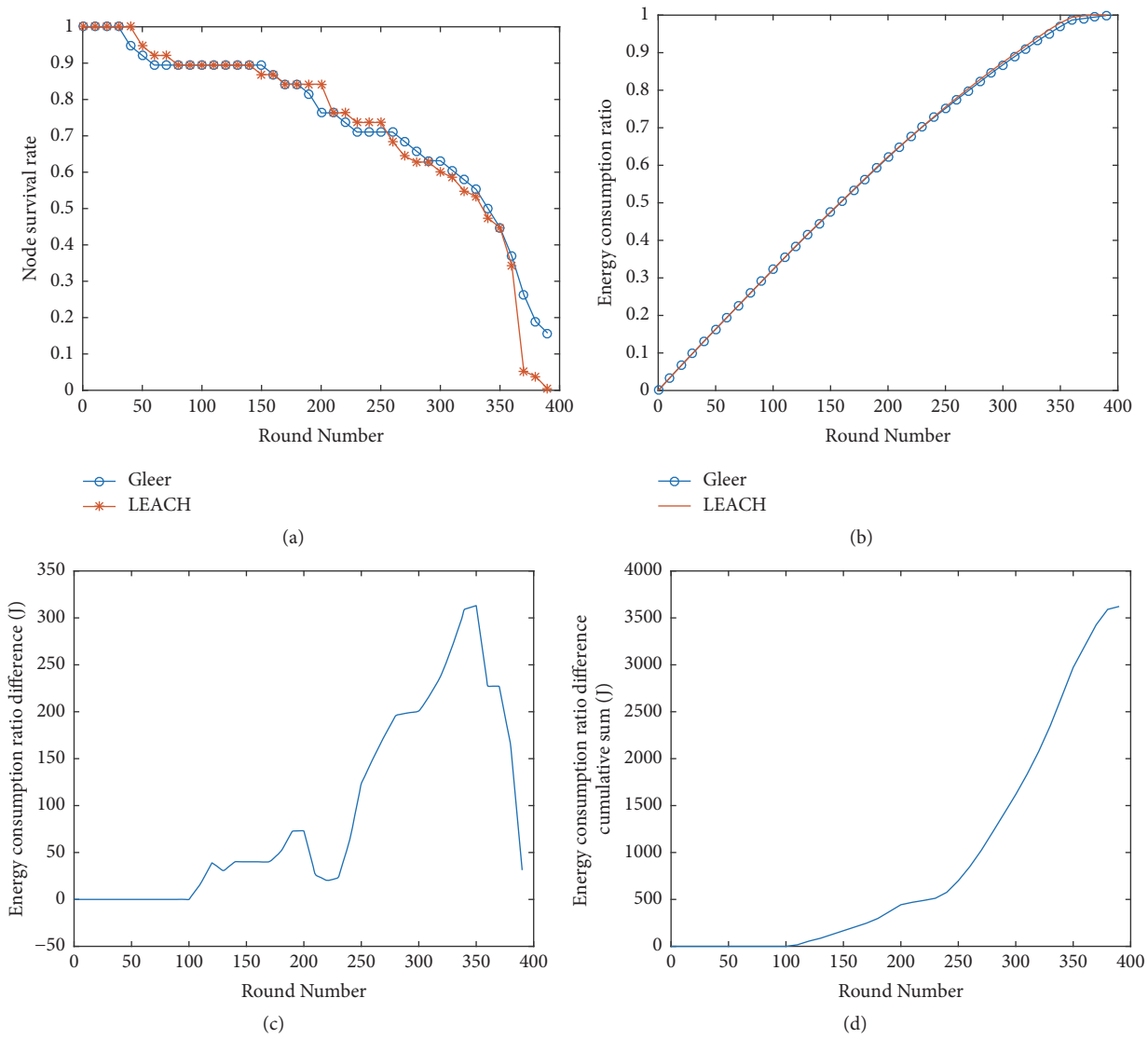


Figure 7: The comparison of node survival rate and energy consumption ratio.

connectivity options, i.e., IEEE 802.11, Bluetooth, and GPRS. In addition, smartphones have stronger personalization. Thus, detection methods, such as traffic detection [33, 34], with good effects to computer world may not be applicable to mobile botnet. Different from computers, smartphones are characterized by limited resources, e.g., CPU and memory. Among these characteristics, the battery is predominant for users to discover the abnormal of their smartphones.

In Gleer, node energy consumption is closely related to botnet structure and energy distribution, which significantly lowers the probability that defenders can observe or disrupt networks. For one node, the probability to be CH is related to the botnet energy gap and division line, rather than its remaining energy. Legacy detection methods, such as power consumption [35] and application signature analysis [36], are unlikely to help protect cellular providers against such activity. Therefore, based on the characteristics of Gleer, exploring features with a combination of unsupervised (clustering) and supervised (classification) machine learning is more appropriate, e.g., the feature extraction from battery usage statistics and attack model. The main steps can be summed up as collecting enough data, analyzing their intrinsic characteristics, adjusting learning algorithms, and finding their internal relations further.

7. Conclusion and Future Work

In this paper, we proposed a new CH selection scheme (Gleer) for heterogeneous multilayer mobile botnet. We first utilized the dynamic energy partitioning threshold α to categorize atomic network nodes into two groups. Then, we introduced the Gini coefficient to estimate the current power gap of nodes and regularized the probability of becoming the region C&C server for each kind node based on the above gap coefficient value. The experimental results show that the proposed scheme works well in narrowing the gap than traditional ones. In order to confirm our results more accurately, we

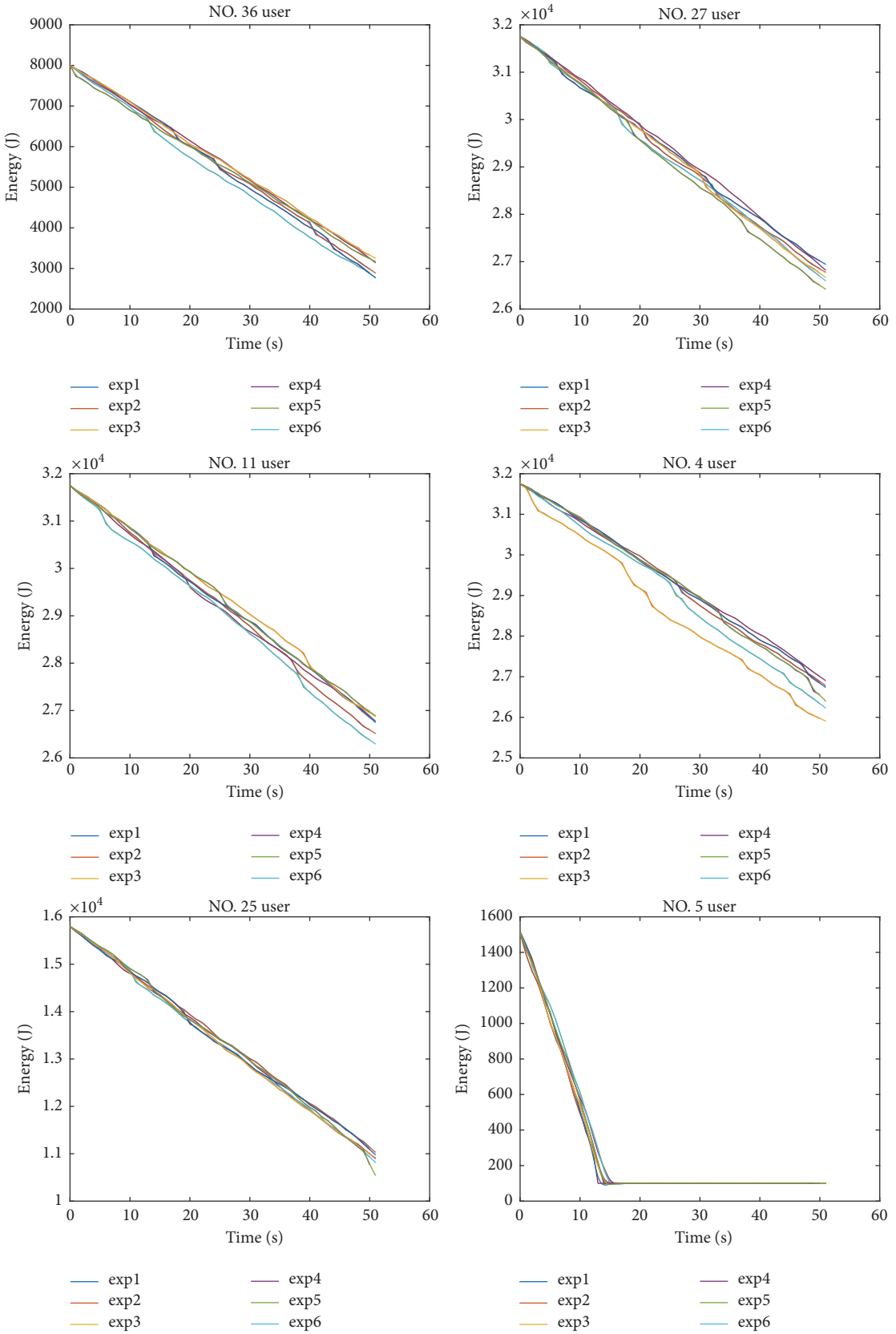


Figure 8: The relationship between the energy and time.

compared Gleer and LEACH in the NS-3 network simulation platform, and the results demonstrate that the mobile botnet with the proposed scheme has longer lifetime and save more energy; that is, the botmaster can execute more attacks with limited energy resources. Meanwhile, we also analyzed the diversity of network energy distribution with different conditions and show the energy consumption variation in users' perspective. And results show that the proposed scheme can maintain the performance diversity of the botnet energy distribution, which benefits the concealment of the mobile botnet.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China (61602376, U1334211, U1534208, 61602374, and 61702411), Natural Science Foundation of Shanxi Province (2017JQ6020), Science Technology Project of Shaanxi Education Department (16JK1573), Ph.D. Research Startup Funds of Xi'an University of Technology (112-256081504), College Research Funds of Xi'an University of Technology (112-451016007), National Foundation of China (U1534208), and Science and Technology Innovation Project of Shanxi Province (2015KTZDGY0104).

References

[1] I. Vural and H. Venter, "Mobile Botnet detection using network forensics," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6369, pp. 57–67, 2010.

[2] J. Milosevic, F. Regazzoni, and M. Malek, "Malware threats and solutions for trustworthy mobile systems design," *Hardware Security and Trust: Design and Deployment of Integrated Circuits in a Threatened Environment*, pp. 149–167, 2017.

[3] R. A. Rodriguez-Gomez, G. Macia-Fernandez, and P. Garcia-Teodoro, "Survey and taxonomy of botnet research through life-cycle," *ACM Computing Surveys*, vol. 45, no. 4, Article ID 2501659, 2013.

[4] K. Singh, S. Sangal, N. Jain, P. Traynor, and W. Lee, "Evaluating Bluetooth as a medium for botnet command and control," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6201, pp. 61–80, 2010.

[5] Y. Zeng, K. G. Shin, and X. Hu, "Design of SMS commanded-and-controlled and P2P-structured mobile botnets," in *Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec'12*, pp. 137–148, usa, April 2012.

[6] W. Chen, X. Luo, C. Yin, B. Xiao, M. H. Au, and Y. Tang, "Muse: Towards robust and stealthy mobile botnets via multiple message push services," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9722, pp. 20–39, 2016.

[7] M. La Polla, F. Martinelli, and D. Sgandurra, "A survey on security for mobile devices," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 446–471, 2013.

[8] L. Cao and X. Qiu, "ASP2P: An advanced botnet based on social networks over hybrid P2P," in *Proceedings of the 22nd Wireless and Optical Communications Conference, WOCC 2013*, pp. 677–682, chn, May 2013.

[9] A. Malatras, E. Freyssinet, and L. Beslay, "Mobile Botnets Taxonomy and Challenges," in *Proceedings of the European Intelligence and Security Informatics Conference, EISIC 2015*, pp. 149–152, gbr, September 2015.

[10] W. Chen, X. Luo, C. Yin, B. Xiao, M. H. Au, and Y. Tang, "CloudBot: Advanced mobile botnets using ubiquitous cloud technologies," *Pervasive and Mobile Computing*, vol. 41, pp. 270–285, 2017.

[11] Chu, I. Shao, C. Y. Lien et al., "A Survey of Localization in Wireless Sensor Network," *International Journal of Distributed Sensor Networks*, vol. 1, pp. 385–391, 2012.

[12] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 553–576, 2016.

[13] J. Zhao, "Topological properties of secure wireless sensor networks under the q-composite key predistribution scheme with unreliable links," *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1789–1802, 2017.

[14] M. Rathee and S. Kumar, "Quantum inspired genetic algorithm for multi-hop energy balanced unequal clustering in wireless sensor networks," in *Proceedings of the 9th International Conference on Contemporary Computing, IC3 2016*, ind, August 2016.

[15] 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.

[16] G. Smaragdakis, M. Ibrahim, and A. Bestavros, "SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks," in *Proceedings of the International workshop in San Patrignano*, Citeseer, 2004.

[17] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[18] P. G. V. Naranjo, M. Shojafar, H. Mostafaei, Z. Pooranian, and E. Baccarelli, "P-SEP: a prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks," *The Journal of Supercomputing*, pp. 1–23, 2016.

[19] M. Jan, P. Nanda, M. Usman, and X. He, "PAWN: A payload-based mutual authentication scheme for wireless sensor networks," *Concurrency Computation*, 2016.

[20] G. Geng, G. Xu, M. Zhang, Y. Guo, G. Yang, and W. Cui, "The design of SMS based heterogeneous mobile botnet," *Journal of Computers*, vol. 7, no. 1, pp. 235–243, 2012.

[21] J. Han, Q. Zhao, and M. Zhang, "China's income inequality in the global context," *Perspectives in Science*, vol. 7, pp. 24–29, 2016.

[22] Y. Jiye, J. Shen, H. Ye et al., "Gini coefficient constraint method for making monthly trade schedule of directly dispatched thermal power generation units," in *Proceedings of the 2016*

IEEE PES Asia Pacific Power and Energy Engineering Conference, APPEEC 2016, pp. 2000–2006, chn, October 2016.

[23] Hoque, A. Anisul, and J. A. Clarke, "On variance estimation for a Gini coefficient estimator obtained from complex survey data," *Communications in Statistics Case Studies Data Analysis & Applications*, vol. 1, no. 1, pp. 39–58, 2015.

[24] Z. Jianhua, "An convenient method to calculate Gini coefficient," *Journal of Shanxi Agricultural University: Social Science Edition*, vol. 6, no. 3, pp. 275–278, 2007.

[25] D. Wu, G. Zeng, L. Meng, W. Zhou, and L. Li, "Gini coefficient-based task allocation for multi-robot systems with limited energy resources," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 155–168, 2018.

[26] Z. Fang, J. Zhu, and R. Deng, "Estimating Gini Coefficient Based on Hurun Report and Poverty Line," *Open Journal of Statistics*, vol. 03, no. 03, pp. 167–172, 2013.

[27] F. Teng, J. He, X. Pan, and C. Zhang, "Metric of carbon equity: Carbon Gini index based on historical cumulative emission per capita," *Advances in Climate Change Research*, vol. 2, no. 3, pp. 134–140, 2011.

[28] O. Tremblay, L.-A. Dessaint, and A.-I. Dekkiche, "A generic battery model for the dynamic simulation of hybrid electric vehicles," in *Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC '07)*, pp. 284–289, IEEE, Arlington, Va, USA, September 2007.

[29] S. G. R. Prasad, R. Vivek, J. Mungara, and E. J. Sebastian, "NS3 simulation studies for optimized neighbour discovery in 6LoWPAN networks," in *Proceedings of the 3rd IEEE International Symposium on Wireless Systems within the IEEE International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS 2016*, pp. 15–18, deu, September 2016.

[30] W. Li, X. Ma, J. Wu, K. S. Trivedi, X.-L. Huang, and Q. Liu, "Analytical Model and Performance Evaluation of Long-Term Evolution for Vehicle Safety Services," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 1926–1939, 2017.

[31] C. M. Shepherd, "Design of Primary and Secondary Ceils: II. An Equation Describing Battery Discharge," *Journal of The Electrochemical Society*, vol. 112, no. 7, pp. 657–664, 1965.

[32] Y. Wang, K. Streff, and S. Raman, "Smartphone security challenges," *The Computer Journal*, vol. 45, no. 12, Article ID 6269870, pp. 52–58, 2012.

[33] J. Zhang, R. Perdisci, W. Lee, X. Luo, and U. Sarfraz, "Building a scalable system for stealthy P2P-botnet detection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 1, pp. 27–38, 2014.

[34] Q. Yan, Y. Zheng, T. Jiang, W. Lou, and Y. T. Hou, "PeerClean: Unveiling peer-to-peer botnets through dynamic group behavior analysis," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks, IEEE INFOCOM 2015*, pp. 316–324, hkg, May 2015.

[35] L. Liu, G. Yan, X. Zhang, and S. Chen, "VirusMeter: Preventing your cellphone from spies," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5758, pp. 244–264, 2009.

[36] T. Oh, S. Jadhav, and Y. H. Kim, "Android botnet categorization and family detection based on behavioural and signature data," in *Proceedings of the 6th International Conference on Information and Communication Technology Convergence, ICTC 2015*, pp. 647–652, kor, October 2015.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 4746020, 10 pages
https://doi.org/10.1155/2018/4746020

Research Article

Reliable Ant Colony Routing Algorithm for Dual-Channel Mobile Ad Hoc Networks

YongQiang Li,[1,2] Zhong Wang,[1] QingWen Wang,[1] QingGang Fan,[1] and BaiSong Chen[1]

[1] *Xi'an Research Institute of High Technology, Xi'an 710025, China*
[2] *Science and Technology on Communication Networks Laboratory, Shijiazhuang 050000, China*

Correspondence should be addressed to QingWen Wang; wqw013890@163.com

Received 2 March 2018; Accepted 11 April 2018; Published 15 May 2018

Academic Editor: Ximeng Liu

For the problem of poor link reliability caused by high-speed dynamic changes and congestion owing to low network bandwidth in ad hoc networks, an ant colony routing algorithm, based on reliable path under dual-channel condition (DSAR), is proposed. First, dual-channel communication mode is used to improve network bandwidth, and a hierarchical network model is proposed to optimize the dual-layer network. Thus, we reduce network congestion and communication delay. Second, a comprehensive reliable path selection strategy is designed, and the reliable path is selected ahead of time to reduce the probability of routing restart. Finally, the ant colony algorithm is used to improve the adaptability of the routing algorithm to changes of network topology. Simulation results show that DSAR improves the reliability of routing, packet delivery, and throughput.

1. Introduction

Ad hoc networks do not rely on communication infrastructures. Their nodes have the simultaneous function of routing and terminal computing. This creates a multihop dynamic wireless network quickly via self-organization [1]. In recent years, ad hoc networks have been widely used in disaster rescue, vehicular networks, battlefields, remote mountain areas, fire zones, earthquake scenarios, multihop satellite networks, and underwater acoustic sensor networks [2]. Owing to the dynamic changes of network topology caused by the high-speed movement of nodes, the exhaustion of battery energy, and the multihop characteristics of ad hoc networks, routing links are broken easily. Thus, the reliability of the network is reduced. The key to improving the reliability in mobile ad hoc networks is selecting reliable routing and nodes with large remaining energy. Some routing protocols, such as AODV [3] and LOADng [4], use the hops mechanism. However, when nodes move quickly, the path may be broken easily. To ensure reliable data packet transmission and reduce energy consumption, a novel reliability prediction model is proposed. The ant colony algorithm is applied to select a reliable path. DSAR takes the integrated path's reliable value as the routing standard, which can better adapt to the changes of network topology.

In mobile wireless ad hoc networks, when the communication load and the number of nodes increase, the single-channel technology will cause network capacity limitations and performance degradation. The multichannel mode has the advantages of low latency and high capacity [5]. Thus, the dual-channel technology is adopted in this paper.

The major contributions of this paper are as follows:

(i) We propose a novel dual-channel communication mode to separate the control and data layers.

(ii) We provide a novel metric to measure the reliability of the integrated link and node.

(iii) A heuristic technique is adopted in DSAR to find the optimal and most reliable route faster than existing schemes.

(iv) Both forward ant and backward ant simultaneously update via pheromones.

(v) The significance of DSAR is evaluated by comparing simulation results with AODV and EEABR.

The paper is designed as follows. The related work of the paper is in Section 2. Section 3 proposes an ant colony routing algorithm with reliability prediction under dual-channel conditions. Section 4 proposes a route discovery process, based on joint optimization of dual-channel networks, and Section 5 reveals simulation results and discussions. Finally, Section 6 concludes this paper.

2. Related Work

Routing protocols designed for wireless ad hoc networks face many challenges, especially in highly dynamic networks. Many researchers have proposed algorithms to address this [6–13]. These routing protocols can be divided into three categories based on the different route discovery strategies: proactive, on-demand, and hybrid protocols. In the active routing protocols (e.g., DSDV [11], FSR [12], GSR [14], and HSR [6]), each network node periodically exchanges routing information with other nodes, and each node must save the routing table, whereas the proactive routing protocol has a low delay; it saves unnecessary routing information and consumes many network resources. With on-demand routing protocols (e.g., DSR [15], AODV [3], RDMAR [16], and LAR [17]), the routing information is created only when nodes are needed. However, there is a large delay with this type of protocol. Both proactive and on-demand routing protocols cannot completely solve routing problems. Therefore, many scholars have proposed hybrid routing protocols that combine the features of active and on-demand routing (e.g., ZRP [18], HWMP [19], and AOHR [20]).

2.1. Representative Schemes for Comparison. In the past few years, swarm intelligence algorithms have become very popular with ad hoc networks, especially the ant colony optimization (ACO) algorithm. The ACO algorithm is based on the ants' foraging behavior in nature to find the optimal path. Graphically, it has been applied widely in various fields [21]. In the ant colony algorithm, the intelligence of a single ant is limited, but the ants can accomplish complex tasks via group cooperation. ACO has the same characteristics as the routing design of mobile ad hoc networks [22]. G. Dicaro proposed ACO via AntNet [23]. AntNet is based on the wired network and the hop mechanism for data transmission. When it is transplanted into ad hoc networks, the power consumption of some nodes grows too high, leading to unstable links. Thus, it is not suitable for mobile ad hoc networks.

ARA protocol [24], proposed by Günes et al. in 2002, was the earliest application of ACO in ad hoc networks. However, the ARA protocol does not consider the energy balance of nodes.

Correia and Vazão proposed the SARA protocol [25], which was improved based on the ARA protocol. SARA uses a restricted neighbor broadcast mechanism, in which each node broadcasts the ant to all the neighbor nodes, but only one neighbor is selected to send the forward ant. SARA uses the same routing maintenance and routing error handling mechanism as the ARA protocol. With routing maintenance, to reduce routing consumption, the data packets also update pheromones on the link. During routing repair, the depth search algorithm is used to control the number of nodes to repair the routing and reduce the routing overhead caused by the source node restart routing discovery process. Compared to the ARA algorithm, the SARA algorithm has high throughput and low routing overhead, but the route establishment process of SARA takes a long time.

AntHocNet [26] is a hybrid algorithm that combines the advantages of AntNet and ARA protocols. The routing process is carried out on demand, and the routing maintenance process is carried out actively. Compared with AODV, the packet delivery rate of AntHocNet protocol has been improved significantly. However, because of the active routing maintenance mechanism, the control overhead is much higher than AODV.

The ACECR algorithm [27] was proposed by Zhou et al. ACECR is based on the ACO algorithm, whose pheromone updates depend on two aspects: the number of hops and the remaining energy. However, this algorithm does not consider stability and reliability.

The R-ACO algorithm [28] preselects highly stable links for packet sending as far as possible to avoid the low link stability of node processing packets. Compared to LAR and AntHocNet, R-ACO improves the success rate of data transmission while reducing communication overhead. However, R-ACO algorithm increases the length of the path and requires nodes to carry GPS devices.

In [2], the level of pheromone is determined according to the routing length, its congestion, and the end-to-end path reliability. This protocol provides high data delivery rates with low end-to-end delay.

For AOCR [29], the authors proposed an on-demand ant colony clustering routing protocol based on a weakly connected dominating set (AOCR). AOCR adopts the weakly connected dominating set as the auxiliary structure of clustering nodes. The forward ants are only broadcast by the heads of each cluster. The pheromone intensity depends on the average residual energy of the network and the minimum energy value of the nodes on the path. AOCR adopts a pseudorandom proportional rule to select the most efficient routing. Compared to other ant colony protocols, AOCR requires less storage space and fewer network transmission resources to perform intelligent routing search.

The authors in [30] proposed FTAR, which avoids forwarding nodes with erroneous tendencies, thus improving the performance of the network. Compared with the DSR algorithm, E2FT [31], and AntHocNet, FTAR improves data transmission success rate and throughput. However, FTAR uses an iterative method to obtain path confidence, which increases the processing time of packets.

The authors in [32] proposed an ant colony optimization routing algorithm (i.e., POSANT), based on geographic location information. For the problem of the excessive number of control packets and high transmission delays in ACO, POSANT combined the advantages of ACO and geographic location information. Therefore, it reduced the time of discovering routing and the number of ants generated by using location information. Compared to the GPSR algorithm, POSANT reduced the delay and increases the success rate of

data transmission, but it did not consider the reliability of the path.

The authors in [33] proposed an enhanced congestion control multipath routing method with ACO optimization for ad hoc networks, which addressed the problem of link blockage. Additionally, the load rapidly increases on the link. They proposed an ACO-based multipath congestion control technique that varies the queue according to the load in a dynamic network. Simulation showed that the proposed ACO protocol had good performance. However, the algorithm did not use multiple channels and does not fundamentally solve the problem of network congestion.

The authors in [5] presented a dual-channel network clustered routing protocol (DNCRP), a hybrid routing protocol. DNCRP uses both 2-hop-distance neighbor cluster IDs and node attribution information to search the path of intercluster routing within the 2-hop-distance neighbor clusters per the source cluster. Simulation results show that DNCRP is suitable for high mobility networks.

In [34], the energy of nodes was considered as a prior factor for route choice. However, this scheme does not consider link reliability as a factor for route choice and increase end-to-end delay.

In summary, there are two major limitations of the current routing algorithm. First, it does not use dual-channel strategies. Second, there are very few routing schemes that consider the reliability of the integrated nodes and links as a prior factor for route-choosing.

Hence, there is a need to develop a unified routing protocol, which can fulfill the low end-to-end delay, low routing overhead, and high reliability. In this paper, a reliable ant colony algorithm based on dual-channel conditions (DSAR) is proposed for ad hoc networks. DSAR overcomes all limitations of the previous schemes.

3. An Ant Colony Routing Algorithm with Reliability Prediction under Dual-Channel Conditions

3.1. Dual-Channel Joint Optimization Model. In ad hoc networks, owing to the limited bandwidth of the nodes, the data transmission delay is high. With technological development, a node in ad hoc networks can be configured with two channels, which can reduce collisions, increase bandwidth, ease network congestion, and improve network performance [35]. The dual-channel network communication model is simplified to the routing problem with a hierarchical map, without considering the channel assignment problem in the network [36]. In this paper, we use a dual-channel layered transmission mode: one channel as the control layer and another as the data layer. The control packets are transmitted in the control layer, and the data packets are transmitted in the data layer. This double channeling eliminates message conflict and reduces the delay of channel handoff. If the control layer is congested and the data layer has enough bandwidth resources, the dual-channel joint optimization mode can transfer control packets in the control layer to the data layer in real time to complete the joint scheduling of the double-layer network and reduce congestion. The dual-channel model is shown in Figure 1.

3.2. Basic Ant Colony Algorithm for Ad Hoc Networks. When an ant walks into an intersection, it randomly selects a path that has not been passed and releases pheromones. The size of the pheromone is related to the path length. The longer the path, the smaller the pheromone. When ants pass by this intersection, they will choose the path where the pheromone is large. Thus, a positive feedback is formed, and the pheromone quantity on the optimal path is larger, and the pheromone on other paths will become less over time. Simultaneously, the whole ant colony can adapt to the change of environment. When ants suddenly encounter obstacles along the way, they can quickly adjust their path. Thus, in the process of the entire colony finding the ants' path, a single ant's optimal path selection ability is limited, but the ant colony has good self-organization because of the global pheromone. Sharing path information, the ants find the optimal path via collective behavior of the ant community. The ant colony algorithm has distributed parallel computer control, which is easy to combine with other algorithms and has strong robustness.

The ant colony optimization algorithm has been successfully applied to many optimization combinatorial problems [37]. The ant foraging process is very similar to the routing problem of ad hoc networks. In this paper, the nest and food are compared to the source node and the destination node in ad hoc networks. The ant colony algorithm uses an ant decision table, which comprises a node selection probability from a path and relevant local information. The ants use this decision table to guide their search of the mobile space in the optimal region, which is the process of forming the routing table. Thus, the ant colony algorithm can be used in ad hoc networks. Through the pheromone mechanism, the ants search for and maintain optimal routing. The mechanism of evaporation updates the pheromone of each node, which can quickly adapt to the needs of the dynamic changes of ad hoc networks.

In these networks, the network topology model is the wireless graph, $G(V, E)$, where V is a network node and E is the link between two nodes. At time t, there are $a_i(t)$ ants. The total number of ants in the network is $m = \sum_{i=1}^{n} a_i(t)$; $P_{ij}(t)$ is the probability of choosing link E_{ij} for ant K at time t.

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{j\in \text{allowed}_k}\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}, & j \in \text{allowed}_k \\ 0 & \text{else} \end{cases}, \quad (1)$$

where $\tau_{ij}(t)$ is the strength of the pheromone in the link E_{ij}; α is a parameter to measure the trajectory of pheromones; η_{ij} is visibility between node i and node j, which is generally defined as $1/d_{ij}$ (d_{ij} is the distance between node i and node j); β is a parameter that measures visibility; and allowed_k is a collection of nodes that have not been visited.

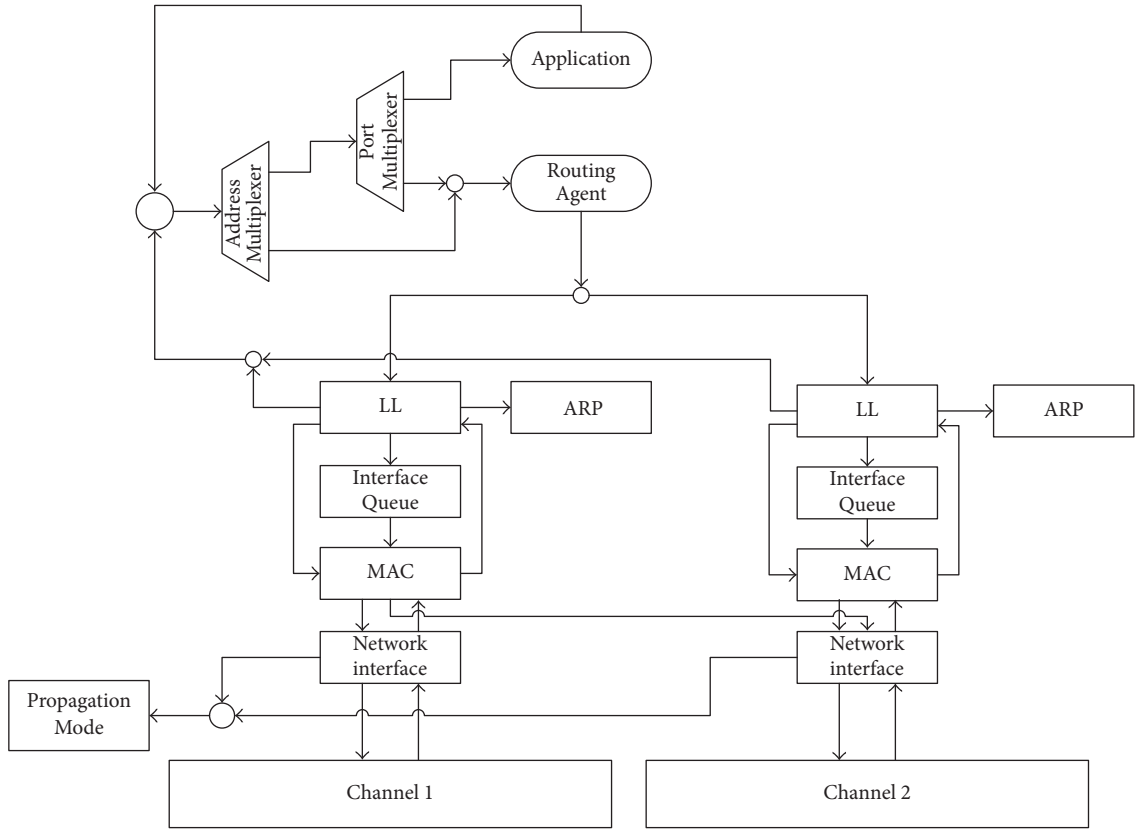


Figure 1: Dual-channel joint optimization mode.

The pheromone update formula on each path in ad hoc networks is as follows:

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \Delta\tau_{ij}, \tag{2}$$

where ρ is the pheromone volatilization coefficient, which is a constant between 0 and 1, and $\Delta\tau_{ij}$ is the increment of the pheromone of ants passing through links i and j.

$$\Delta\tau_{ij} = \sum_{k=1}^{l} \Delta\tau_{ij}^{k}. \tag{3}$$

3.3. Route Reliability. In ad hoc networks, there are two main reasons for path breaking. One is the movement of nodes on the communication path, and the other is the nodes withdrawing from the network because of energy depletion. Thus, we select relatively reliable nodes and links. Then the path stability (PS) factor is introduced to judge the stability of the path. During the establishment of QoS routing, the path with the strongest stability is selected from the multiple paths satisfying the QoS requirements, reducing the probability of path breaking. In this paper, the path stability factor is the function of the link stability factor and the node energy stability factor.

3.3.1. The Node Energy Stability Factor. Suppose that there are j intermediate nodes in the path, P_i; $N_i = \{n_{i1}, n_{i2}, \ldots, n_{ij}\}$. In this paper, we define the node energy stability factor.

$$\mathrm{ES}_{ij} = \frac{E_{\mathrm{currrent}}(ij)}{E_{\mathrm{initial}}(ij)}, \tag{4}$$

where $E_{\mathrm{initial}}(ij)$ is the initial energy of node j in path P_i and $E_{\mathrm{currrent}}(ij)$ is the current remaining energy of node j in path P_i.

Because a node in the path cannot be used, owing to the exhausted energy, the energy stability factor of the path is the minimum energy stability factor of all nodes in the path, P_i.

$$\mathrm{ES}_i = \min\left\{\mathrm{ES}_{ij}\right\}, \quad l_{ij} \in p_i. \tag{5}$$

3.3.2. The Link Stability Factor. In this paper, we define the link reliability factor as the remaining lifetime of the link. The communication radius of each node is R. Each node is equipped with GPS; thus, every node can perceive the location, speed, time of nodes, and period and send its own coordinates and speed information to neighbor nodes. According to the location information of nodes, the remaining lifetime of each link can be predicted, and the link stability factor can be obtained. As shown in Figure 2, the initial distance between the two nodes, M and N, is d.

The relative motion of the two nodes is equivalent to one node moving, while the other node is stationary. The coordinates of node N, relative to the stationary node, M, are

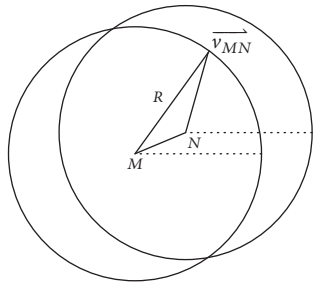


Figure 2: Calculate the lifetime of the link MN.

$(x_N - x_M, y_N - y_M)$. Thus, the distance between M and N is $d = \sqrt{x_{MN}{}^2 + y_{MN}{}^2}$. According to the relative movement of the two nodes, after time t, node M, relative to position N, is (x'_{MN}, y'_{MN}).

$$\begin{aligned} x'_{MN} &= x_{MN} + \left|\overrightarrow{v_{MN}}\right| t \cos\theta_{MN}. \\ y'_{MN} &= y_{MN} + \left|\overrightarrow{v_{MN}}\right| t \sin\theta_{MN}, \end{aligned} \tag{6}$$

When the distance between M and N is R, the link between M and N is broken.

$$d' = \sqrt{\left(x'_{MN}\right)^2 + \left(y'_{MN}\right)^2} = R. \tag{7}$$

Take (6) into (7):

$$\begin{aligned} &\left|\overrightarrow{v_{NM}}\right|^2 t^2 + 2\left|\overrightarrow{v_{NM}}\right| \left(x_{NM}\cos\theta_{NM} + y_{NM}\sin\theta_{NM}\right) t \\ &\quad + d^2 - R^2 = 0, \end{aligned} \tag{8}$$

where

$$\begin{aligned} \overrightarrow{v_M} &= \left(\left|\overrightarrow{v_M}\right|\cos\theta_M\right)\overrightarrow{i} + \left(\left|\overrightarrow{v_M}\right|\sin\theta_M\right)\overrightarrow{j}, \\ \overrightarrow{v_N} &= \left(\left|\overrightarrow{v_N}\right|\cos\theta_N\right)\overrightarrow{i} + \left(\left|\overrightarrow{v_N}\right|\sin\theta_N\right)\overrightarrow{j}, \\ \overrightarrow{v_{NM}} &= \overrightarrow{v_N} - \overrightarrow{v_M} \\ &= \left(\left|\overrightarrow{v_N}\right|\cos\theta_N - \left|\overrightarrow{v_M}\right|\cos\theta_M\right)\overrightarrow{i} \\ &\quad + \left(\left|\overrightarrow{v_N}\right|\sin\theta_N - \left|\overrightarrow{v_M}\right|\sin\theta_M\right)\overrightarrow{j} \\ \left|\overrightarrow{v_{NM}}\right| &= \sqrt{\left(\left|\overrightarrow{v_N}\right|\cos\theta_N - \left|\overrightarrow{v_M}\right|\cos\theta_M\right)^2 + \left(\left|\overrightarrow{v_N}\right|\sin\theta_N - \left|\overrightarrow{v_M}\right|\sin\theta_M\right)^2}, \\ \theta_{NM} &= \tan^{-1}\left(\frac{\left|\overrightarrow{v_N}\right|\sin\theta_N - \left|\overrightarrow{v_M}\right|\sin\theta_M}{\left|\overrightarrow{v_N}\right|\cos\theta_N - \left|\overrightarrow{v_M}\right|\cos\theta_M}\right). \end{aligned} \tag{9}$$

Because t cannot be negative, t becomes the following:

$$\begin{aligned} t = &\frac{\sqrt{R^2 - \left(x_{NM}\sin\theta_{NM} - y_{NM}\cos\theta_{NM}\right)^2}}{\left|\overrightarrow{v_{NM}}\right|} \\ &- \frac{\left(x_{NM}\cos\theta_{NM} + \theta_{NM}\sin\theta_{NM}\right)}{\left|\overrightarrow{v_{NM}}\right|}. \end{aligned} \tag{10}$$

Link stability is

$$\begin{aligned} \mathrm{LS}_{MN} &= \frac{t}{t_{\max}} = \frac{t}{R/\left|\overrightarrow{v_{NM}}\right|} \\ &= \frac{\sqrt{R^2 - \left(x_{NM}\sin\theta_{NM} - y_{NM}\cos\theta_{NM}\right)^2}}{R} \\ &\quad - \frac{\left(x_{NM}\cos\theta_{NM} + \theta_{NM}\sin\theta_{NM}\right)}{R}. \end{aligned} \tag{11}$$

There is a k link in a path between the source node, S, and the destination node, D. The link stability of the path depends on the minimum link stability factor in the k-segment link. Thus, we define the link reliability factor of path, P_i, as the minimum link stability factor.

$$\mathrm{LS}_i = \min\left\{\mathrm{LS}_{MN}\right\}, \quad l_{MN} \in p_i. \tag{12}$$

3.3.3. The Path Reliability Factor. Considering the link stability factor and node energy stability factor, the comprehensive reliability factor of the path is defined as

$$\mathrm{PS}_i = \frac{1}{1/\mathrm{ES}_i + 1/\mathrm{LS}_i} = \frac{\mathrm{ES}_i \times \mathrm{LS}_i}{\mathrm{ES}_i + \mathrm{LS}_i}. \tag{13}$$

Whether the communication path between the source node and the destination node is stable depends on the worst links and nodes in the path, because the entire communication is interrupted when there is a broken link or when a node exits the network because of exhausted battery energy. After using the above method, if either of the two stability factors is small, the value of PS_i will be small. Thus, path reliability will be poor. When the source node finds multiple paths satisfying the QoS requirements, the one with the largest PS is selected. Thus, $\mathrm{PS} = \max\left\{\mathrm{PS}_i\right\}$ $P_i \in P$.

4. Route Discovery Process Based on Joint Optimization of Dual-Channel Networks

To improve channel utilization, based on the double-channel model and ant colony optimization algorithm, the two-interlayer joint optimization routing mode is proposed. It can increase bandwidth and make full use of idle resources between different layers. In Mode 1, the routing service in the control layer can only be transmitted in the control layer. If the control layer does not have enough channel resources, the service will be rejected. To reduce the blocking rate of the control layer, joint optimization Mode 2 is proposed. When the data layer has enough idle resources, the control packets in the control layer can be transmitted to the data layer in real time to realize the joint optimization of the two-layer network. The specific routing process of the two modes is as follows.

Mode 1. It is the route discovery mode on the control layer of DSAR.

(1) Broadcast a Hello Message. At the control layer, each network node periodically sends a 1-hop hello message. The

Table 1: The format of the hello message.

Type	Src_addr	Scr_X	Scr_Y	Scr_V	N_energy

Table 2: The structure of neighbor.

N_addr	N_energy	LS	Hops
N_X	Nr_Y	N_V	Phenomenon

format of the hello message is shown in Table 1, where "Type" denotes packet type, "Src_addr" is the address of the source node that sends the hello message, "Scr_X" are the X coordinates of node, "Scr_Y" are the Y coordinates of node, "Scr_V" is velocity vector of node, and "N_energy" is residual energy of node.

(2) Establish the Neighbor List. Each node establishes its neighbor list by receiving hello messages sent by neighbor nodes in real time. In this paper, each network node has a GPS positioning device to obtain its geographic location information. Each network node sends hello packets periodically and accepts those sent by neighbor nodes. Each hello packet contains node coordinates and speed information and the neighbor list of each node. The neighbor list contains the location vector information of all neighboring nodes. Thus, each network node can obtain the location information of neighbor nodes. The frequency of packet transmission can be set according to different motion scenarios, and the transmission frequency of hello messages can be higher for scenarios where network topology changes rapidly. In this paper, we design hello messaging with a sending interval of 1 s. If a node does not receive neighbor hello packets in 1 s, the link of this node is broken, and the node deletes it from its neighbor table. The node updates its neighbor list immediately after the hello message is received. Otherwise, the comprehensive reliability of the node is calculated according to formula (13). The format of neighbor message is shown in Table 2, where "N_addr" is the address of the source node that sends the hello message, "N_energy" is residual energy of neighbor node, "N_X" are the X coordinates of neighbor node, "N_Y" are the Y coordinates of neighbor node, N_V" is velocity vector of neighbor node, "LS" is the stability of neighbor node, "Hops" denotes total hops of a route passing through this neighbor node, and "Phenomenon" denotes the value of pheromone.

(3) Send a Forward Ant and Update Pheromone. When the source node, S, has data packets sent to the destination node, D, the source node, S, looks at the routing information table, namely, the pheromone table. If there is no routing information and the control layer has enough bandwidth, the forward ant is broadcast in the control layer. If there is a route to node D, the packet is sent directly to the data layer. In the initial stage of routing establishment, the source node broadcasts a certain number of forward ants at the control layer. The unicast or broadcast of each intermediate node between source node S and destination node D depends on whether the intermediate node has pheromone. If there is

Table 3: The format of forward ant message.

Type	Fant_addr	Fant_Seqno	TTL
Scr_Y	Scr_V	Esum	Fdk
Seqno 0			
Seqno 1			
⋮			
Seqno n			

pheromone at each intermediate node, the probability that ant K selects the next hop neighbor node, j, is calculated according to the following equation:

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}(t)\right]^{\beta}}{\sum_{j\in \text{allowed}_k}\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}(t)\right]^{\beta}}, & j \in \text{allowed}_k \\ 0 & \text{else,} \end{cases} \tag{14}$$

where $P_{ij}^{k}(t)$ is the probability that ant K selects the next hop neighbor node, j, when node i moves toward the destination node. $\tau_{ij}(t)$ is the size of the pheromone of node i at time t in link ij; $\eta_{ij}(t)$ is a visual function of node i at time t from node i to node j; allowed_k is a collection of neighbor nodes of node i; α and β are adjustment coefficients; α is a relative importance coefficient of the residual pheromone; β is a relative importance coefficient of heuristic information. The pheromone of node is updated as equation:

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t). \tag{15}$$

The format of the forward ant message is shown in Table 3, where "Type" denotes the packet type, "Fant_addr" is the address of source node, "Fant_Seqno" is the sequence number of forward ants generated by source nodes, "Seqno" denotes node address visited by the forward ant, "TTL" is the survival time of forward ant, and "Fdk" denotes the number of hops that the forward ant moves to the current node.

(4) Send a Backward Ant and Update Pheromone. When a forward ant arrives at the destination node, it turns into a backward ant. The backward ant returns to the source node along its former path. If a fault occurs in the link of the next hop, caused by the movement of a node in the path to the source node, then the backward ant will be discarded. When the backward ant returns to node i from node j, the pheromone of node i is updated according to the pheromone update of (16).

$$\tau_{ij}(t+1) = \frac{\tau_{ij}(t)}{1-\rho} + \Delta\tau_{ij}(t), \tag{16}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t), \tag{17}$$

Table 4: The format of backward ant message.

Type	Visitednode	*E*min	*E*avg	Bd*k*

$$\Delta\tau_{ij}^{k}(t) = \mathrm{PS}_i, \tag{18}$$

$$\eta_{ij}^{k}(t) = \frac{1}{\mathrm{Fd}_k}. \tag{19}$$

The format of the backward ant message is shown in Table 4, where "Type" denotes the packet type, "Visitednode" denotes the ID of node visited, "*E*min" is the energy value of the minimum energy node on the path that the backward ant passes through, "*E*avg" is the residual average energy of the ant K to the current node, and "Bd*k*" denotes the number of hops experienced by the backward ant K to node j.

Mode 2. It is the route discovery process, based on joint optimization of dual-channel networks

When the control layer is congested, joint optimization Mode 2 is proposed. That is, when the data layer has enough idle resources, the control packets in the control layer can be transmitted to the data layer in real time to realize the joint optimization of the two-layer network. The specific routing process of Mode 2 is as follows.

(1) If the control layer has enough network resources, the forward ant is routed via Mode 1 in the control layer. Otherwise, turn to (2).

(2) When the forward ant, K, moves to node i in the data transport layer, the forward ant, K, looks at whether there are available channel resources for ants to find paths with neighboring nodes of i. If not, ant K stops and refuses to perform the routing lookup service. Otherwise, it turns to (3).

(3) Ant K performs the routing service in the data transmission layer and finds the next hop node, j, in Mode 1.

(4) When reaching node j, ant K first investigates whether there are enough channel resources between node j and its neighbor nodes in the control layer to perform the routing service. If not, ant K continues to perform the path-finding service in the data transport layer. Otherwise, ant K returns to the control layer and searches the optimal path of service according to Mode 1.

5. Simulation and Analysis

In this paper, to verify the reliability of the DSAR protocol, NS-2 is selected, and the DSAR algorithm is compared to the EEABR algorithm [34] and the AODV algorithm. AODV is a classic routing algorithm, and EEABR is a successful application of the ant colony algorithm in wireless ad hoc networks.

5.1. Simulation Setup. In a wireless simulation environment, each mobile network node is randomly distributed in the 1,000 m × 1,000 m area; 50 nodes are randomly arranged according to the random way-point model. The communication radius of each node is 250 m. MAC layer adopts dual-channel mode. The packet length is 512 b, and the send rate varies from 1 to 16 packets/s. The evaporation of pheromone occurs every 1 s. The evaporation rate, ρ, is set to 0.2. Each value of α and β is set to 20 and 15, respectively. The simulation time is 1,000 s. To reduce random errors, the experimental results will be the average of the 10 experiments. Simulation algorithm routing layers are (1) DSAR, (2) AODV, and (3) EEABR.

5.2. Simulation Analysis

5.2.1. Performance Metrics for Evaluating Routing Protocol. The performance of routing protocol is evaluated by means of end-to-end delay, average throughput, packet delivery rate, routing overhead, and so forth. The statistical methods are introduced as follows.

(1) End-to-End Delay. The average end-to-end delay is the time required from the start of routing to the end of data transmission. We calculate it with the following formula:

$$\tau = \frac{1}{N}\sum_{i=1}^{N}\left(t_{ai} - t_{bi}\right), \tag{20}$$

where τ is the average end-to-end delay; N is the number of successful packet transmissions; t_{ai} is the time that packet i arrives at the destination node; and t_{bi} is the time packet i was generated.

(2) Throughput. Throughput is the maximum number of packets that a network successfully transmits per unit time.

$$T = \frac{1}{T_{\mathrm{RE}} - T_{\mathrm{RS}}}\sum_{i=1}^{N} R_b(i) \times 8, \tag{21}$$

where T represents the throughput; $R_b(i)$ represents the number of bytes of packet i received successfully; N represents the total number of packets received from the destination; T_{RE} represents the reception time of the data packet; and T_{RS} represents the beginning of the data packet reception.

(3) Packet Delivery Rate. Packet delivery rate is the ratio of the total number of sending packets to the total number of receiving packets.

(4) Routing Overhead.

$$N = \frac{P_C}{P_D}, \tag{22}$$

where N represents routing overhead; P_C represents the total number of node send control packets; and P_D represents the total number of destination node receive data packets.

5.2.2. The Network Performance Varies with the Packet Send Rate of the Source Node. The performance of the three algorithms varies with the average packet sending rate of the source node in the network, as shown in Figures 3–6.

Figure 3 shows the relationship between the average end-to-end delay and the packet delivery rate of the source nodes

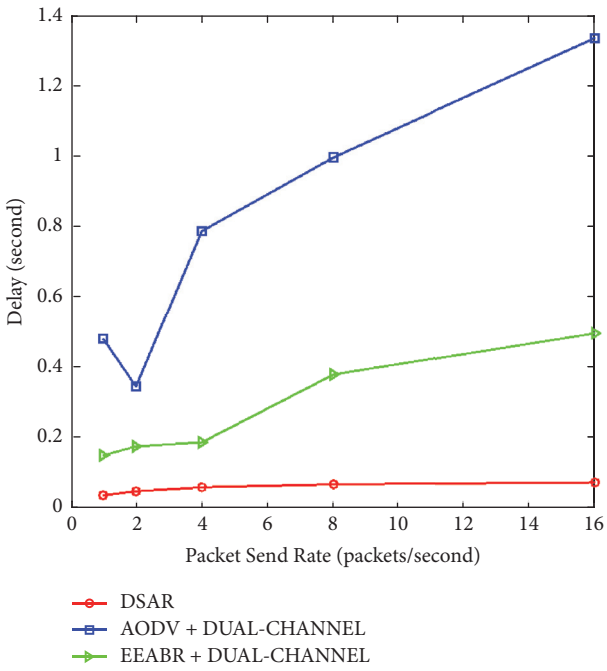


Figure 3: End-to-end delay.

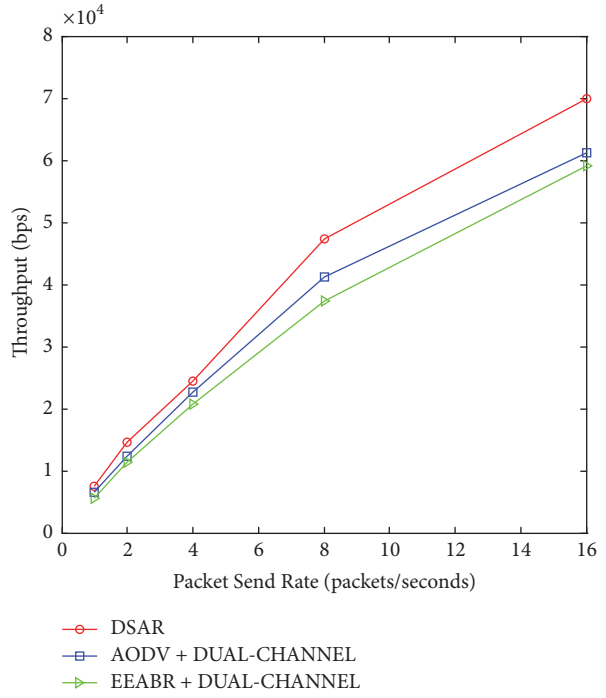


Figure 4: Throughput.

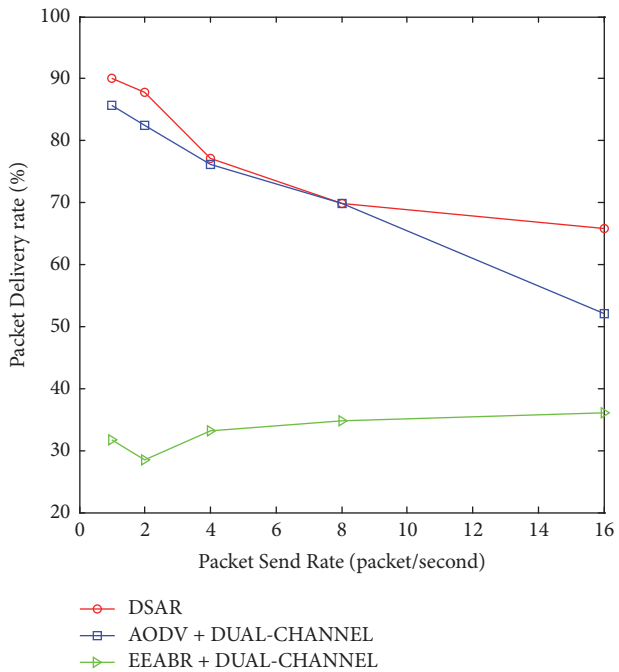


Figure 5: Packet delivery rate.

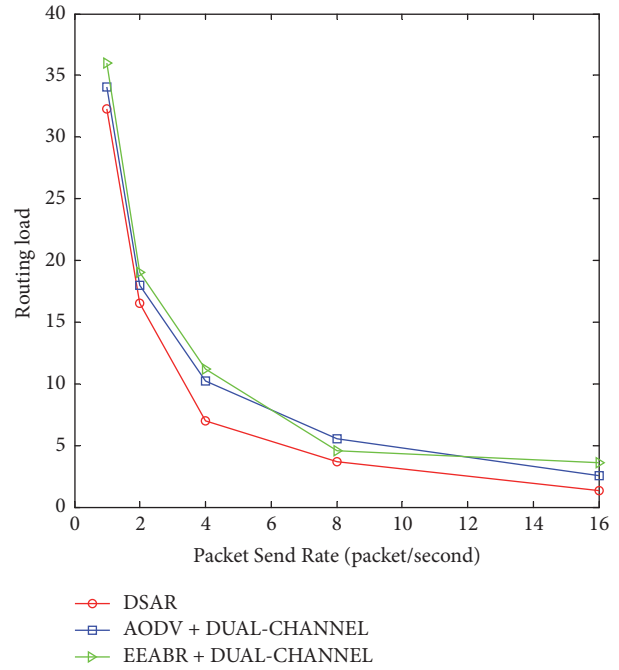


Figure 6: Routing overhead.

in the three algorithms. As shown in the figure, the average end-to-end delay of the three algorithms increases with the increase of the sending speed of the source node. The average end-to-end delay of DSAR is significantly smaller than that of EEABR and AODV. This is because, with the increasing packet sending rate of the source node and the congestion of the network, the DSAR selects the nodes having large stability and large residual power to transmit data packets. This reduces the delay caused by link interruption and routing restart repair. Simulation results show that, compared with the classical EEABR and AODV algorithms, the average end-to-end delay of the RSAR algorithm is reduced.

Figure 4 shows the relationship between the throughput of the three algorithms and the packet delivery rate of the source node. As can be seen in the graph, the throughput

of each routing discovery increases with the increase of the packet delivery rate of the source node from the three algorithms. The routing discovery throughput of DSAR is significantly higher than EEABR and AODV. This is because the dual-channel mechanism is adopted by DSAR to separate control packets from data packets, which reduces the channel switching and data collision probability. DSAR uses a comprehensive stability prediction mechanism to select the path having good stability and fewer hops and establishes high-quality routing, which reduces the probability of routing restarts and improves throughput.

Figure 5 shows the relationship between the packet delivery rate of the three algorithms and the packet delivery rate of the source node. From the graph, it can be seen that, with the increase of routing load, the packet delivery rate of DSAR is higher than that of AODV and EEABR. However, DSAR and EEABR decrease rapidly with the increase of packet sending rate, whereas AODV remains unchanged. DSAR has packet delivery rates higher than AODV and EEABR. This is because the control packets and the data packets are transmitted over different channels, which reduce packet collision and increase network bandwidth. The poor performance of EEABR is caused by the increase of transmission packet collisions and the periodic transmission of ant packets.

Figure 6 shows the relationship between the routing overhead of three algorithms and the packet delivery rate of source nodes. From the simulation results, the routing overhead is reduced with the increased packet sending rate. EEABR generates a large number of ant packets, which increases the cost of route discovery. However, the overhead of ADOV is lower than that of EEABR and DSAR, because AODV uses on-demand routing. DSAR needs to send periodic probe packets to find stable nodes and links, so the cost of DSAR will be slightly higher than EEABR. With the increase of load, the routing overhead of DSAR approaches that of AODV, because the frequent retransmission caused by the instability in AODV leads to the increase of routing overhead.

6. Conclusion

To improve the reliability of routing protocol in wireless ad hoc networks, a reliable ant colony algorithm for dual-channel systems was proposed. In the DSAR algorithm, the double-layer mechanism of control layer and data layer separation was established, which reduced packet collision and channel handoff delay and increased network bandwidth. Simultaneously, when the data layer had enough idle resources, it transferred the blocked routing service over the control layer to the data layer in real time, completing the joint scheduling of the double-layer network and reducing the congestion rate. Moreover, the reliability prediction mechanism was proposed, which enhanced link reliability and reduced the probability of routing restart. Also, for the dynamic change of topology in ad hoc networks, the ant colony algorithm was used to adapt the dynamic changes of network topology. The comprehensive reliability value of the proposed reliability prediction model was used as one of the bases of pheromone updates for the ant colony algorithm. Simulation results show that, compared with the classic AODV and EEABR models, DSAR improved the reliability of routing protocols.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61601475).

References

[1] F. Khan, Q. Jabeen, S. Khan, and M. Ahmad, *Performance Improvement in Multihop Wireless Mobile Adhoc Networks*, 2016.

[2] G. Li, L. Boukhatem, and J. Wu, "Adaptive Quality-of-Service-Based Routing for Vehicular Ad Hoc Networks with Ant Colony Optimization," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3249–3264, 2017.

[3] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," No. RFC 3561, 2003.

[4] T. H. Clausen and A. C. D. Verdiere, "The LLN On-demand Ad hoc Distance-vector Routing Protocol -Next Generation (LOADng," in *Networking & Internet Architecture*, 2015.

[5] D. U. Chuan-Bao, H. D. Quan, L. I. Zhao-Rui, and P. Z. Cui, "Design and Analysis of Hierarchical Routing Protocol for Wireless Dual-Channel Ad Hoc Networks," in *Journal of Command & Control*, 2015.

[6] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang, "A wireless hierarchical routing protocol with group mobility," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 1538–1542, IEEE, New Orleans, La, USA, September 1999.

[7] J. J. Garcia-Luna-Aceves and M. Spohn, "Source-tree routing in wireless networks," in *Proceedings of the 7th International Conference on Network Protocols (ICNP '99)*, pp. 273–282, IEEE, November 1999.

[8] S. Murthy and J. J. Garcia-Luna-Aceves, "Routing protocol for packet radio networks," in *Proceedings of the 1st Annual International Conference on Mobile Computing and Networking (MobiCom '95)*, pp. 86–95, Berkeley, Calif, USA, November 1995.

[9] W. Guo, J. Li, G. Chen, Y. Niu, and C. Chen, "A PSO-Optimized Real-Time Fault-Tolerant Task Allocation Algorithm in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3236–3249, 2015.

[10] X. Luo, D. Zhang, L. T. Yang, J. Liu, X. Chang, and H. Ning, "A kernel machine-based secure data sensing and fusion scheme in wireless sensor networks for the cyber-physical systems," *Future Generation Computer Systems*, vol. 61, pp. 85–96, 2016.

[11] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *Computer Communication Review*, vol. 24, no. 4, pp. 234–244, 1994.

[12] M. Gerls, Fisheye State Routing (FSR) for Ad Hoc Networks, Internet Draft, draft-ietf-manet-fsr-03.txt, 2002.

[13] B. Xu and F. Sun, "Composite intelligent learning control of strict-feedback systems with disturbance," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–12, 2017.

[14] . Tsu-Wei Chen and M. Gerla, "Global state routing: a new routing scheme for ad-hoc wireless networks," in *Proceedings of the ICC '98 1998 IEEE International Conference on Communications. Conference Record*, pp. 171–175, Atlanta, GA, USA, 1998.

[15] D. B. Johnson, "The dynamic source routing in ad hoc wireless networks (DSR)," in *Mobile Computing*, 1996.

[16] G. Aggelou and R. Tafazolli, "RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks," in *Proceedings of the 2nd ACM International Workshop on Wireless Mobile Multimedia, WOWMOM 1999*, pp. 26–33, usa.

[17] B. K. Young and H. V. Nitin, "Location-Aided Routing (LAR) in mobile ad hoc networks," *Wireless Networks*, vol. 6, no. 4, pp. 307–321, 2000.

[18] L. Barolli, Y. Honma, A. Koyama, A. Durresi, and J. Arai, "A selective border-casting zone routing protocol for ad-hoc networks," in *Proceedings of the 15th International Workshop on Database and Expert Systems Applications*, pp. 326–330, September 2004.

[19] K. Yang and J.-F. Ma, "Hybrid wireless mesh protocol," *Tongxin Xuebao/Journal on Communication*, vol. 30, no. 11 A, pp. 133–139, 2009.

[20] S. Wu, X. Tan, and S. Jia, "AOHR: AODV and OLSR hybrid routing protocol for mobile ad hoc networks," in *Proceedings of the 2006 International Conference on Communications, Circuits and Systems, ICCCAS*, pp. 1487–1491, chn, June 2006.

[21] S. Kashef and H. Nezamabadi-pour, "An advanced ACO algorithm for feature subset selection," *Neurocomputing*, vol. 147, no. 1, pp. 271–279, 2015.

[22] A. George, *Performance Analysis of Energy Efficient Location Based ACO Routing Algorithm for Mobile Ad Hoc Networks using Bonn Motion Mobility Models*, 2015.

[23] G. Di Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 2011.

[24] M. Günes, U. Sorges, and I. Bouazizi, "ARA—the ant-colony based routing algorithm for MANETs," in *Proceedings of the International Conference on Parallel Processing Workshops*, pp. 79–85, British Columbia, Canada, August 2002.

[25] F. Correia and T. Vazão, "Simple ant routing algorithm strategies for a (Multipurpose) MANET model," *Ad Hoc Networks*, vol. 8, no. 8, pp. 810–823, 2010.

[26] G. Di Caro, F. Ducatelle, and L. M. Gambardella, "AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 443–455, 2005.

[27] J. Zhou, H. Tan, Y. Deng, L. Cui, and D. D. Liu, "Ant colony-based energy control routing protocol for mobile ad hoc networks under different node mobility models," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, article no. 105, 2016.

[28] D. Kadono, T. Izumi, F. Ooshita, H. Kakugawa, and T. Masuzawa, "An ant colony optimization routing based on robustness for ad hoc networks with GPSs," *Ad Hoc Networks*, vol. 8, no. 1, pp. 63–76, 2010.

[29] K. H. Li and J. S. Leu, *Weakly connected dominating set-assisted ant-based routing protocol for wireless ad-hoc networks*, Pergamon Press, Inc, 2015.

[30] S. Misra, S. K. Dhurandher, M. S. Obaidat, K. Verma, and P. Gupta, "A low-overhead fault-tolerant routing algorithm for mobile ad hoc networks: A scheme and its simulation analysis," *Simulation Modelling Practice and Theory*, vol. 18, no. 5, pp. 637–649, 2010.

[31] Y. Xue and K. Nahrstedt, "Fault tolerant routing in mobile ad hoc networks," in *Proceedings of the 2003 IEEE Wireless Communications and Networking Conference: The Dawn of Pervasive Communication, WCNC 2003*, pp. 1174–1179, usa, March 2003.

[32] S. Kamali and J. Opatrny, "POSANT: a position Based Ant Colony Routing Algorithm for Mobile Ad-hoc Networks," *Journal of Networks*, vol. 3, 21 pages, 2008.

[33] S. Rathore and M. R. Khan, "Enhance congestion control multipath routing with ANT optimization in Mobile ad hoc Network," in *Proceedings of the 2016 International Conference on ICT in Business, Industry, and Government, ICTBIG 2016*, ind, November 2016.

[34] I. Woungang, M. S. Obaidat, S. K. Dhurandher, A. Ferworn, and W. Shah, "An ant-swarm inspired energy-efficient ad hoc on-demand routing protocol for mobile ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '13)*, pp. 3645–3649, Budapest, Hungary, June 2013.

[35] A. Biradar, R. C. Thool, R. Velur, and T. S. Indumathi, "Dual channel based multi-objectives genetic routing protocol for ad-hoc networks and optical networks using power aware clustered topology," in *Proceedings of the International Conference on Optical Engineering*, pp. 1–6, 2013.

[36] K. Liu, S. Liu, and H. Jiao, "Routing algorithm based on ant colony optimization in the dual-channel wireless sensor network," *Journal of Xidian University*, vol. 40, pp. 58–62, 2013.

[37] I. Alaya, C. Solnon, and K. Ghedira, "Ant Colony Optimization for Multi-Objective Optimization Problems," in *Proceedings of the IEEE International Conference on TOOLS with Artificial Intelligence*, pp. 450–457, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 4392524, 10 pages
https://doi.org/10.1155/2018/4392524

Research Article

Mining the Relationship between Spatial Mobility Patterns and POIs

Liping Huang,[1] Yongjian Yang,[1] Xuehua Zhao,[2] Hepeng Gao,[3] and Limin Yu[4]

[1]*College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[2]*School of Digital Media, Shenzhen Institute of Information Technology, Shenzhen 518172, China*
[3]*College of Software, Jilin University, Changchun 130012, China*
[4]*Shandong Agriculture and Engineering University, Jinan 250100, China*

Correspondence should be addressed to Xuehua Zhao; lcrlc@sina.com

Received 24 February 2018; Revised 16 March 2018; Accepted 29 March 2018; Published 10 May 2018

Academic Editor: Ximeng Liu

Passengers move between urban places for diverse interests and drive the metropolitan regions as the aggregation of urban places to group into network communities. This paper aims to examine the relationship between the spatial patterns (represented by the network communities) of mobility flows and places of interest (POIs). Furtherly, it intends to identify the categories of POIs that play the most significant role in shaping the spatial patterns of mobility flows. To achieve these purposes, we partition the study area into disjoint regions and construct the network with each partitioned region as a node and connection between them as links weighted by the mobility flows. The community detection algorithm is implemented on the network to discover spatial mobility patterns, and the multiclass classification based on the logistic regression method is adopted to classify spatial communities featured by POIs. Taking the taxi systems of Shanghai and Beijing as examples, we detect spatial communities based on the movement strengths among regions. Then we investigate their correlations with POIs. It finds that communities' modularity correlates linearly with POIs; particularly governments, hotels, and the traffic facilities are of the most significance for generating the mobility patterns. This study can provide valuable insight into understanding the spatial mobility patterns from the perspective of POIs.

1. Introduction

People move in a city, generating the population mobility flows between places. Acquiring the volume of mobility flows in different places in a city is particularly important as it benefits a convergence of applications, such as location selecting for a retail store to allow increasing customers to shop around and advertisement casting for capturing as many consumers as possible [1, 2]. Technological advances allow for precise measurements of mobility flows on large datasets [3–13], including taxi trajectories [14–16], mobile phone trajectories [17, 18], and transport smart cards [19].

By solving the privacy-preserving problem of mobility traces [20–24], retrospective studies of mobility flows which focus on modeling the mobility flows from a place to another, such as the universal model, called radiation model [25], are proposed and applied to predict human movements [26]. Though the model is parameter-free that requires only population distribution as input, it disregards the spatial cluster features of mobility flows, which means that most people travel in a specific range of regions instead of the whole city and some of the citizens share similar regional scope. To analyze the spatial variability of urban mobility flows, we construct the spatial network with the metropolitan regions as nodes and the connections between them as links weighted by the aggregated strengths of interregion movements [1, 17, 27]. The community in the spatial network is applied to further analysis of the spatial patterns of mobility flows as it offers a visual representation of spatial cluster features of mobility flows, where a spatial community is a set of nodes which have more connections among themselves than with the rest of nodes [1, 28]. The community in the spatial network is then named as the spatial community in this paper, representing the spatial patterns of urban mobility flows. The community detection allows one to identify the innercommunity links which plays a very important role

in understanding the travel pattern and interaction among urban regions [29, 30]. For example, based on the mobility flows around the city area of Shanghai, Liu et al. [16] built the spatial network and adopted the community detection to model spatial patterns around the city area.

Combined with the techniques of the network, applications based on mobility flows are widely developed in the field of urban computing [31, 32]. For example, the centrality metrics of the network are used to estimate the importance of road segments [14], and the network connectivity is applied to reveal new latent links among urban regions [33]. Studies mentioned above provide insights into using mobility flows in networks to reveal the mobility patterns or urban structures. However, these have not given the underlying mechanisms that motivate the urban mobility flows from the land-use aspects and socioeconomic views.

Actually, urban mobility flows are rooted in people's traveling activities (e.g., work or entertainment) [19, 34], reflected by specific POIs. Retrospective studies of spatial communities improve our ability to analyze the mobility flows from the perspective of the network. However, they do not provide insight into the factors that motivate the population mobility dynamics. As each urban movement contains the origin and the destination that is determined by the travel motivation [35], the regions as the origin and destination of a trip are the cause of mobility flows. In [19, 36, 37] POIs are collected to explain the activity patterns and model the dynamic decision-making process that shapes individual's movements. Besides, POIs are combined with mobility flows to discover functional regions [2], where the segmented regions of the city area carry socioeconomic functions as people live in the regions and POIs fall in regions. In [38], POIs are applied to find the characteristics of resident trips based on the clustering method. It finds that the residents' travel pattern in the working day can be expressed as "spatial relative dispersion-spatial aggregation-spatial relative dispersion." The effectiveness of these proposed models indicates that mobility flows are related to the POI distribution among urban regions.

However, there has not existed research concentrating on the relationship between spatial communities and POIs, which should be taken into consideration in the future urban developing planning of POIs for the prediction of community changes. In this study, we aim to study the relationship between spatial communities and POIs. And we intend to find the group of specific categories of POIs to explain the identified communities. Taking the large-scale and real-world datasets of Shanghai and Beijing in China as examples, we construct the networks with the urban regions and interregion movements, where the communities are detected. We collect POIs for each node to characterize people's mobility motives and study the relationship between spatial communities and POI features by adopting the stepwise logistic regression.

Researching the inherent relationship between spatial communities of mobility flows and POIs provides a new insight for understanding the underlining mechanism of urban movements. In accordance with the research aim of our work, the rest of this paper is organized as follows: Section 2 presents the methods used in this paper, including the relationship estimation model and the significant POIs identification method. Experiments are implemented in Section 3. We discuss the experimental findings in Section 4. Finally, we briefly conclude the paper in Section 5.

2. Methods

2.1. Relationship Estimation Model. A mobility used in this paper is a 2-tuple $\langle (x_o, y_o), (x_d, y_d) \rangle$. Both (x_o, y_o) and (x_d, y_d) are geospatial positions, respectively, representing the origin and destination of a trip. In detail, the OD pair represents a trip starting at (x_o, y_o) and ending at location (x_d, y_d).

As shown in Figure 1, to construct the spatial networks, in this research, the study area is segmented into disjoint grids, and each grid g_i is set as a node. Trips between two nodes indicate the existence of an edge or a linkage. After extracting mobility flows from the travel trace dataset, the volume of mobility originating from g_i and ending in g_j is set as the weight w_{ij} from n_i to n_j. Thus a weighted and directed network is constructed.

As shown in the Figure 1, some nodes indicate much stronger connections among them than with other nodes. By dividing the network into densely connected subnetworks, the urban area is partitioned into intensely interactive sub-regions. In network science, community detection methods can partition an entire network into tightly connected subnetworks, called communities, and reveal the network's clustering characteristics.

A community, also called a cluster or a module, is typically regarded as a group of vertices which probably share common properties or play similar roles within the network, and the metric of modularity is always to estimate the community detection results [39–41]. When applied to weighted and directed networks, the modularity Q is defined as [42]

$$Q = \sum_{i=1}^{m} \frac{w_{ij}}{w} - \frac{w_i^{\text{in}} w_i^{\text{out}}}{w}. \tag{1}$$

Here w_{ij} is the total weight of links starting and ending in module i, w_i^{in} and w_i^{out} are the total in- and out-weight of links in module i, and w is the total weight of all links in the network.

To optimize Q, the vast majority of searching strategies take one of the following steps to evolve starting the network partitions: merging two communities, splitting a community into two, and moving nodes between distinct communities. We employ a high-quality modularity based community detection algorithm that adopts all the three strategies, called Combo [43] as the adopted community detection technique.

In the spatial networks, partitioned regions are set as nodes, and the number of OD pairs is set as the weight on the directed edge. To explain how the spatial communities are generated by the mobility flows in the network, the ultimate proof of the hidden reason is to match the mobility patterns to POIs distributed among regions. We match POIs of the studied area to nodes, in accordance with the geolocation using the process of map matching. We get the

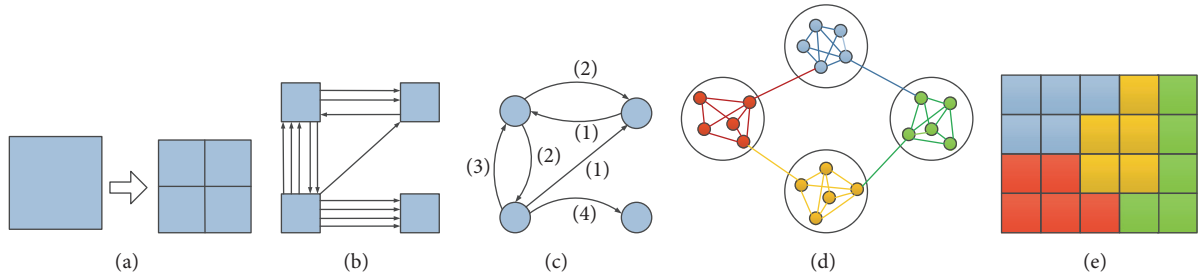


FIGURE 1: Illustration of the network and communities (this is the prototype proposed by Liu et al., 2015). To construct a network based on mobility flows, the study area is divided into small regions (a) with each small regions corresponding to a node in the network. A directed edge or linkage existed between two nodes if there are mobility flows from one node to the other. The weight of an edge equals the volume of mobility flows represented in (b, c). Graphic (d) provides an illustration of the communities detected from a network, which is divided into four parts (depicted by four circles) in which the subnetworks had relatively dense connections. The community detection result corresponds to closely connected subregions (e).

POI features of each node in the network, denoted as $xi = (x^{(1)}, x^{(2)}, \ldots, x^{(M)})$, where M is the number of POI category (particularly equal to 17 in our case studies), and $x^{(j)}$ is the number of POI category j in node i. After applying community detection algorithms to networks, the nodes are partitioned into disjoint sets (spatial communities). Nodes in the same community share the same value y of classification label Y. Then the community label Y is set as the dependent variable. Suppose that the value set of Y is $\{1, 2, \ldots, K\}$, then the multinomial logistic regression is defined as

$$P(Y = k \mid x) = \frac{\exp(w_k \cdot x + b)}{1 + \sum_{k=1}^{K-1} \exp(w_k \cdot x + b)}, \quad k = 1, 2, \ldots, K-1 \quad (2)$$

$$P(Y = K \mid x) = \frac{1}{1 + \sum_{k=1}^{K-1} \exp(w_k \cdot x)},$$

where $w = (w^{(1)}, w^{(2)}, \ldots, w^{(M)})$ and b are parameters of the model. Given the testing set $D = \{(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)\}$, let D^k denote the samples labelled with k, and $\theta = (w, b)$, then the MLE (maximum likelihood estimation) is applied to calculate the parameters:

$$l(\theta_k) = \log P(D_k \mid \theta_k) = \sum_{x \in D_k} \log P(x \mid \theta_k)$$
$$\widehat{\theta}_k = \arg\max_{\theta_k} l(\theta_k). \quad (3)$$

We adopt the stepwise strategy to select POI categories for the logistic regression, and the fitness metric of the R-square guarantees that none redundant dependents are selected. It means that we choose categories of POIs that make sense for distinguishing the spatial communities.

2.2. Significant POIs Identification. For the problems of multiclassification based on logistic regression, one class is always set as the reference class as shown by (2). To identify the categories of POIs that affect the spatial communities, in this paper, each community is set as the reference class in turn, and then the statistic frequency of the significant POIs is set as one element C_{kj} of the feature vector of a community C_k. As shown in (4) and (5), each element of the feature vector for a community C_{kj} represents the frequency of the jth significant category of POIs in community C_k. Then the feature value of a community is calculated by its norm multiplied by the entropy. This guarantees that we selected communities of more significant POI categories and more diverse of the significance frequency.

$$F_k = -|C_k| \sum_{j=1}^{N_k} p(C_{kj}) \log_2 p(C_{kj}) \quad (4)$$

$$C_{kj} = \text{feq}\left(\text{sig}\left(j^{\text{POI}}\right)\right). \quad (5)$$

Then we identify significant POIs by (5), where top n percentage of communities of the largest F_k is selected as the candidate set. Then we identify the most significant categories as one element in the ultimate significant POI set.

$$S^{\text{POI}} = \bigcup_{k=K\times n\%}^{k=\max_k F_k} \text{POI}\left(\max_j C_{kj}\right). \quad (6)$$

For example, suppose that we got four communities, and the significance frequency of each POI for a community is shown in Table 1.

$|C_k|$ selects communities that contain larger significance frequency of the total significant POI categories. And the entropy tends to select spatial community candidates that consist of more frequency variation. Specific to C_2 and C_3, $N_2 = 3$, and $N_3 = 1$, though $|C_3| > |C_2|$, the entropy of C_3 is smaller than that of C_2, meaning smaller difference between significance frequency of POIs. When n is set as 50, we select two communities to identify significant POIs that feature the spatial communities. C_1 and C_2 are selected, and then POIs of traffic facility and enterprise are identified as the ultimate significant ones that make sense for forming this community snapshot.

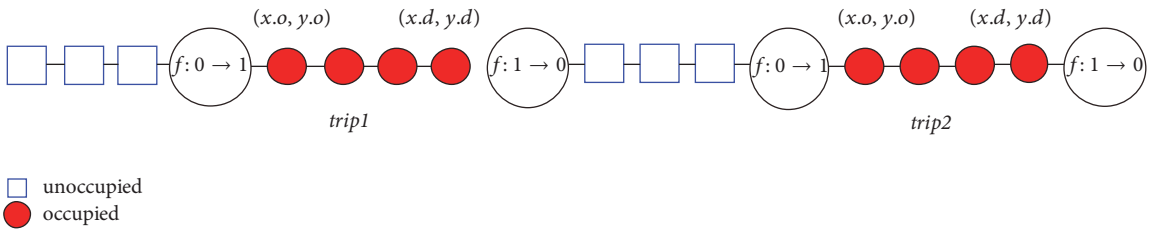
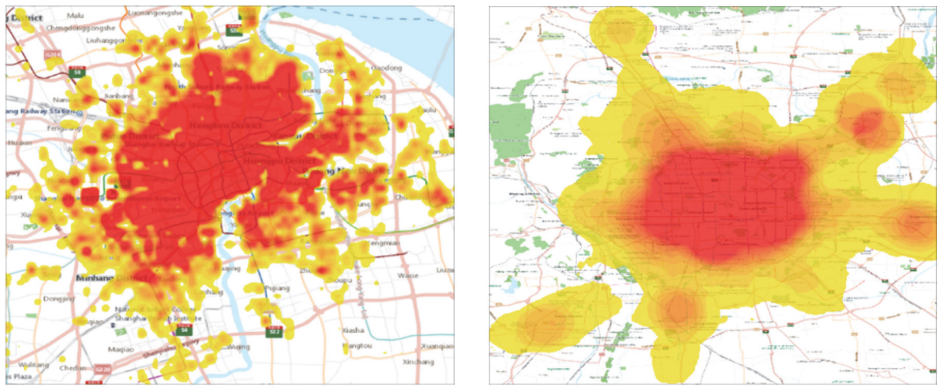


Figure 2: Mobility extraction from taxi trajectories. The occupation state changing from unoccupied to occupied or from occupied to unoccupied is adopted to extract the origination and destination of an urban movement.



(a) (b)

Figure 3: The extracted points for Shanghai and Beijing are, respectively, depicted in (a) and (b).

Table 1: Illustration of the significant POIs identification.

POI	Frequency of a POI category			
	C_1	C_2	C_3	C_4
Traffic Facility	4	4	2	0
Enterprise	3	2	2	1
Hospital	3	1	2	1
Real estate	2	1	2	0
$\lvert\mathbf{C_k}\rvert$	6.16	4.69	5.29	1.73
Entropy	1.5	1.5	0	1
$\mathbf{F_k}$	9.24	7.03	0	1.73

3. Experiments

3.1. Datasets. Taking the spatial networks of Beijing and Shanghai as case studies, datasets of taxi GPS trajectories of both cities are collected. As shown in Figure 2, taxi trajectories are used to extract mobility flows. A taxi trajectory is a sequence of GPS points pertaining to the taxi's sampling location over time. Each point consists of a tuple $\langle (x, y), f \rangle$ with location (x, y), and the taxi's occupancy status f, where (x, y) is a pair of spatial coordinates representing latitude and longitude. $f = 1$ means that the taxi is occupied by passengers; otherwise $f = 0$. The flag f bound to each trajectory position is essential for judging the taxi occupation state, which is used to extract the origin and destination points (OD) of a trip. All other GPS points between a pair of OD $= \langle (x_o, y_o), (x_d, y_d) \rangle$ own the same occupation state $f = 1$. The extracted points for Shanghai and Beijing are, respectively, shown in Figure 3.

Table 2: Studied area and OD number for the network of Shanghai and Beijing.

City	Longitude	Latitude	Node	OD
Shanghai	120.45–122.10	30.00–31.93	2926	38,0640
Beijing	116.0–116.8	39.65–40.25	3995	186,2799

To construct the spatial networks, we first divide the spatial area into grids of 1 km by 1 km using the open street map (OSM) (http://www.openstreetmap.org/copyright), then each grid is set as a node in the network. We extract mobility flows between any nodes by matching origin points or destination points to grids using the OSM. The mobility flow volume originating from grid i to grid j is set as the weight on the directed edge. Disregarding grids visited by none OD pairs, it remains 2926 nodes for the spatial network of Shanghai and 3995 nodes for Beijing. Then datasets of mobility flows are as shown in Table 2.

We use the Baidu APIs (Liu et al., 2015) to collect the POIs in two cities. Seventeen categories of POIs are collected as shown in Figure 4. As the category of POIs will be set as the independent variables in the relation estimation model in this study, we label them as $X^{(i)}$, as shown in Table 3. Each category of POIs is set as a specific dimension of the independent variable. And the number of each POI category

Table 3: Seventeen categories of POIs.

Number	POI category	Variable	Beijing	Shanghai
1	Beauty	$X^{(1)}$	55,696	51,689
2	Traffic facility	$X^{(2)}$	96,356	123,124
3	Entertain	$X^{(3)}$	140,445	183,402
4	Enterprise	$X^{(4)}$	128,188	178,562
5	Hospital	$X^{(5)}$	12,924	9,680
6	Real estate	$X^{(6)}$	300,214	233,777
7	Government	$X^{(7)}$	25,556	20,660
8	Education	$X^{(8)}$	40,381	39,343
9	Culture	$X^{(9)}$	3,971	3,723
10	Scenic spot	$X^{(10)}$	56,996	48,463
11	Auto service	$X^{(11)}$	50,898	55,479
12	Living service	$X^{(12)}$	158,121	149,576
13	Food	$X^{(13)}$	86,301	82,021
14	Shopping	$X^{(14)}$	208,245	230,715
15	Spots	$X^{(15)}$	11,026	9,561
16	Hotel	$X^{(16)}$	8,501	3,704
17	Finance	$X^{(17)}$	22,139	23,386

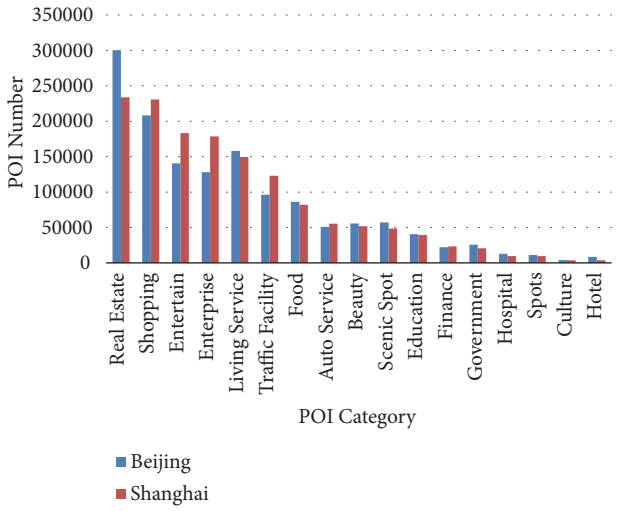


Figure 4: Categories of POIs for Shanghai and Beijing.

of the two cities is as shown in Table 3 and Figure 4; totally we collect 1,446,865 POIs for the network of Shanghai and 1,405,954 POIs for the network of Beijing.

3.2. Results

3.2.1. Relationship between Spatial Communities and POIs. The spatial communities are affected by the travel distance. Thus we add the distance threshold (DT) into the spatial community detecting process. As shown in Figure 5, for the networks of Shanghai, the edge number and the mobility flow reaches 90% as the distance threshold gradually increase to 20 km and 14 km; similar to networks of Beijing, the critical distance is 25 km and 9 km.

The modularity of community detection results for two cities is, respectively, shown in Figure 6, together with the stepwise logistic regression results, R-square. It can be found that the modularity is changing with the distance threshold (DT) increasing.

Larger DT means that more edges and more mobility flows are added to the networks so that it gets smaller modularity for the networks of Shanghai and Beijing. It also shows that the modularity tends to be convergent with the distance threshold tending to be larger. And the modularity variation trend presents quite similar for both networks.

As shown in Figures 7(a) and 7(b), even nodes in the suburban area are connected to the spatially close communities. And the distance increased by 1 km takes little variation (16 km to 17 km) to the spatial community detection results.

Note that the mobility flow density of Beijing network is 466, while it is just 130 for the network of Shanghai. The modularity got for the spatial networks of Beijing is generally larger than that of Shanghai as in Figure 6.

The results of regression fitness (R-square) for both networks also tend to be convergent. The median value of adjusted $R2$ obtained is, respectively, 0.3 for the Shanghai networks and 0.48 for the Beijing networks. This verifies that the spatial communities are tensely correlated to the POI features. Then the quantitative correlation between the modularity and the R-square is presented as shown in Figure 8. It shows that the adjusted $R2$ presents to be positively and linearly correlated to the regression of the modularity. This indicates that the spatial communities are correlated to POIs, and the spatial communities can be explained by POIs.

3.2.2. Identified POIs. The community detection results got without distance threshold limitation for Shanghai and Beijing are depicted in Figure 9.

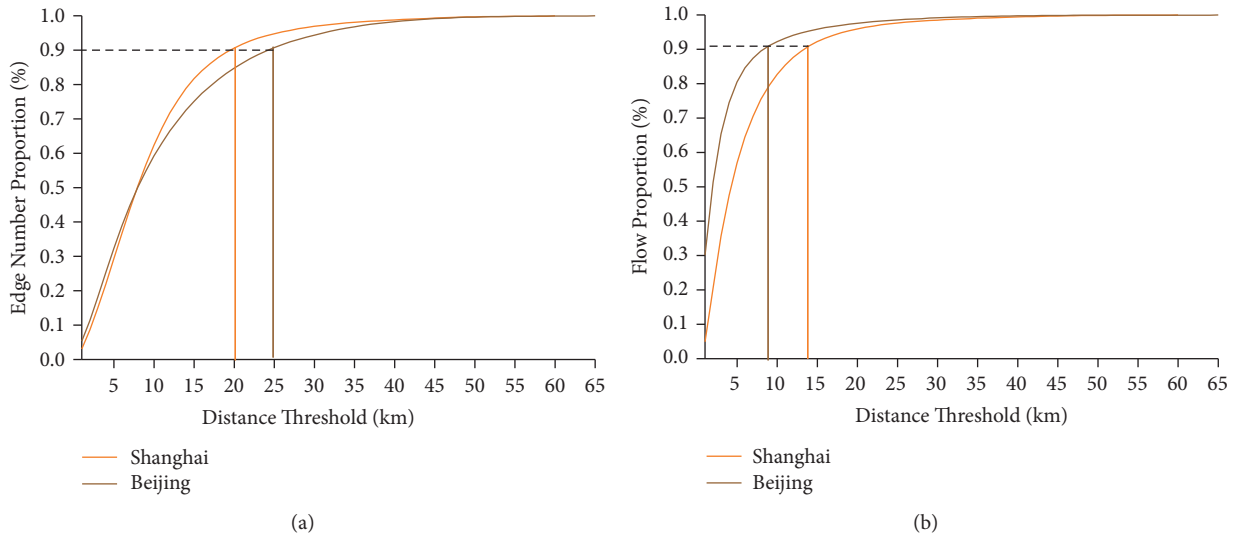


Figure 5: Variation of edge and flow with distance threshold.

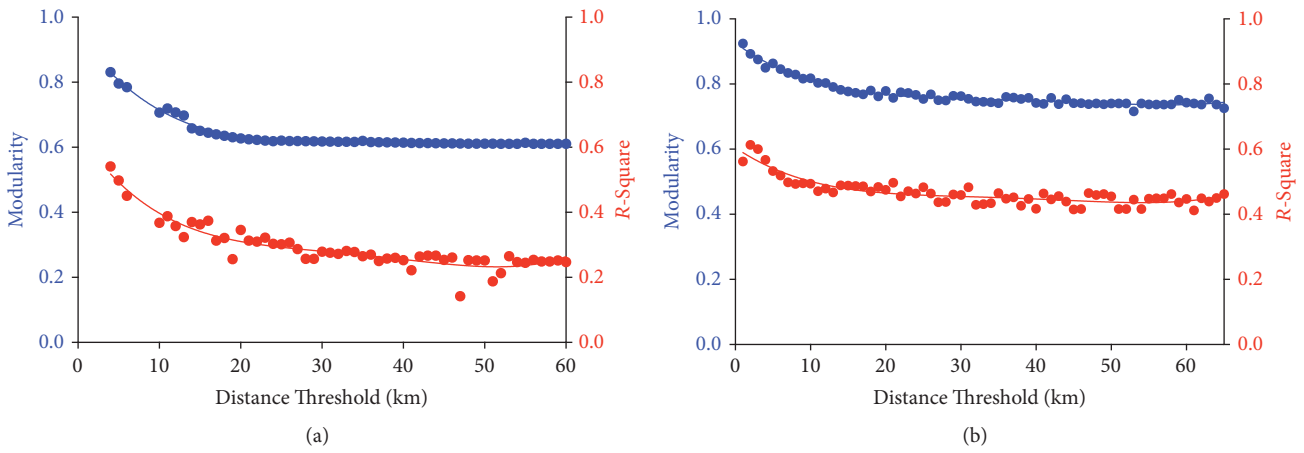


Figure 6: Modularity and R-square in the Shanghai networks (a) and the Beijing networks (b).

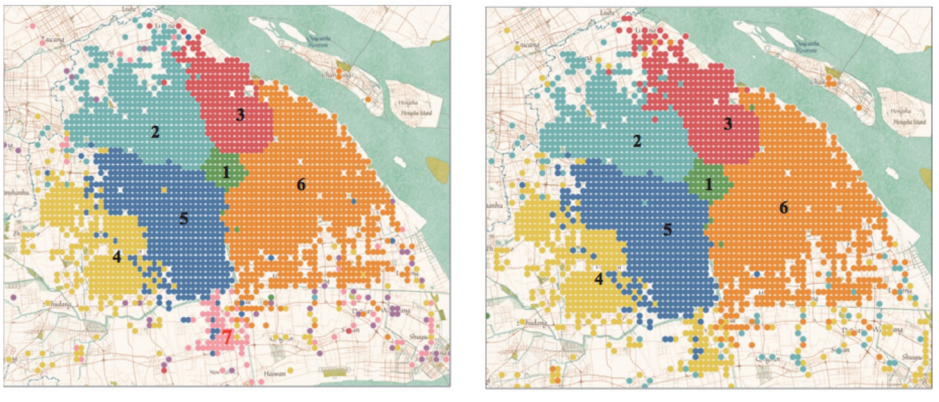


Figure 7: Communities detected for the network of Shanghai with DT = 16 km in (a) and DT = 17 in (b).

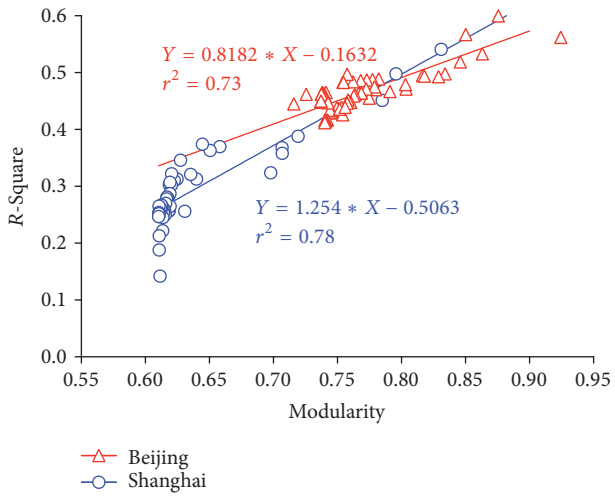


FIGURE 8: Scatter plot of the modularity and the R-square shows the positive and linear relationship between two metrics.

TABLE 4: Identified significant POIs.

POI	Beijing using [38]	Beijing using our model	Shanghai using our model
Shopping	√		√
Traffic facility		√	√
Living service	√	√	
Food		√	
Government	√	√	√
Enterprise	√		√
Finance			√
Hotel		√	√
Education	√	√	

As shown in Figure 9, we get seven spatial communities for the spatial network of Shanghai and thirteen spatial communities for Beijing. It can be observed that both cities are polycentric.

Then each community is set as the reference class in turn to conduct the stepwise regression method, and we use the R-square as the metric for estimating the regression results. The significance of the variables is adopted to identify the POI categories that are tensely correlated to the spatial communities. Note that some POIs are identified as shared categories for both cities. When n is set as 50, which means that we select half number of communities of the largest value of F, significant frequency of each POI category in a community is as shown in Figures 10 and 11. And the identified POIs for both cities are as shown in Table 4.

To verify the effectiveness of the proposed method, the identified POIs using the method proposed in [38] are listed in Table 4. It can be found that the reference method also identified the POI categories of living service, government, and education in Beijing, which certifies the effectiveness of the proposed significant POI identification method. Compared with findings in [38], which partitions time of day into three time intervals (morning, evening, and night), the proposed identification method finds that the traffic facilities play an important role on shaping the community pattern in urban transport networks for both Beijing and Shanghai. These findings fit in with the actual situation in life, as traffic facilities satisfy the daily commuting needs. This further verifies the effectiveness of the proposed model.

The significant POIs for generating spatial communities in the network of Shanghai contain shopping, enterprise, traffic facility, government, finance, and hotel, while it contains the living service, traffic facility, food, government, and hotel for the network of Beijing. It is found that the POIs of traffic facility, government, and hotel are identified as the common significant POIs to distinguish the communities in both networks.

4. Discussion

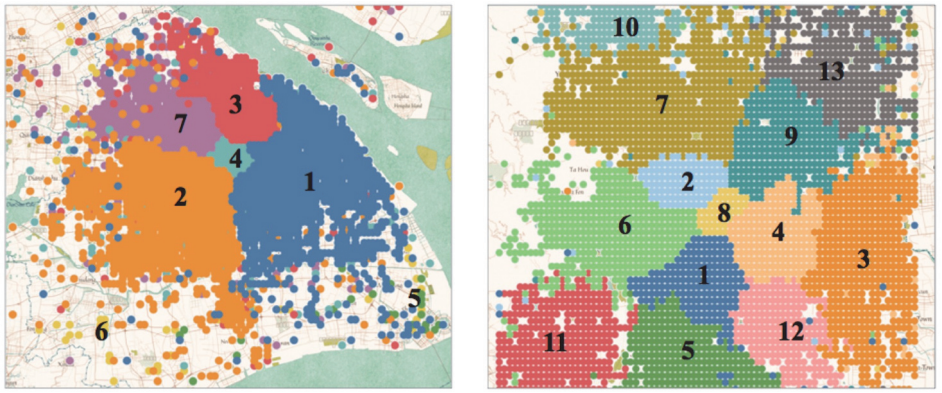
Understanding the spatial patterns and finding the driving factors of the urban mobility flows help planners to evaluate the urban construction plan. To study the drivers of communities of mobility flows, we propose to estimate the relationship between spatial communities and POIs.

Using the taxi systems of Shanghai and Beijing as case studies, the experimental results show that the communities in spatial networks generated by mobility flow linearly correlate to the POIs. To further recognize the specific factors that drive the spatial communities of mobility flows, the stepwise logistic regression is used, and it is found that the POIs of governments, hotels, and the traffic facilities are common features that play an important role on distinguishing communities for both cities.

From the socioeconomical perspective, the locations of governments in a city attract various types of facilities and improve the economic development of the surrounded area, which is reflected by the spatial communities of mobility flows. Similarly, hotels are always located in the area of numerous facilities. A small number of hotels can be a better representative for the regional features that attract mobility flows [44]. Traffic facilities play a role in forming community pattern of mobility flows [45–48], which may be because that these facilities satisfy the essential need for daily traveling and life. Note that mobility flows used in this paper are merely extracted from taxicabs. Thus, another reason for the significance of these categories of POIs may be that citizens are more likely to choose taxicabs due to the arbitrary option of travel departure time. Possibly, taxicabs are also popularly preferred as the transfer tool for the public transport system, such as train station, subway station, or bus station. After all, most commuters more likely choose buses or the subway, and travelers less take taxis for a long trip, especially in the two metropolises in China.

The computational complexity of the relationship and POI identification model is mainly reflected in the community detection process, which justified an upper bound to the execution time of $O(N^2 \log(C))$, where N is the number of nodes and C the number of communities in the network.

This study has some limitations. Mobility flows are only extracted from the taxi trajectories, and other spatial



(a) (b)

Figure 9: Spatial communities for Shanghai (a) and Beijing (b).

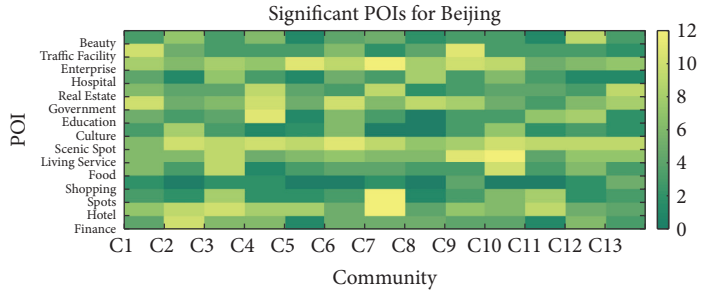


Figure 10: The significance frequency of each POI category of 13 communities for the network of Beijing is shown.

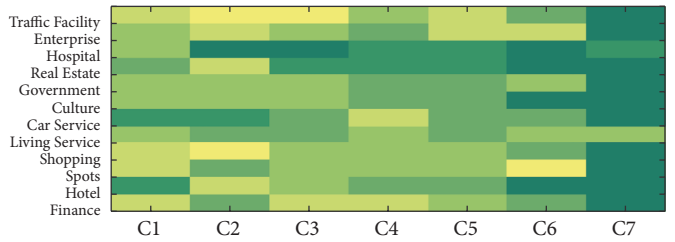


Figure 11: The significance frequency of each category in the seven spatial communities for the network of Shanghai is depicted.

community patterns may be found with various data sources. However, the same analysis of methods could be used. In this study, we focus on the spatial communities generated by the taxi systems; future studies could consider the similarity and differences of the spatial communities in other public transport systems. Another limitation is that we just adopt the number of each category of POIs as the influencing factors, disregarding the scale of each POI, which should be considered in future works.

5. Conclusion

This paper proposes a model for estimating the relationship between the spatial communities of mobility flows and the urban POIs, thus to identify the categories of POIs that drive mobility flows to network communities.

Taking the mobility flows in Beijing and Shanghai as case studies, we find that the spatial communities can be explained by the POIs. Specifically, it is found that POIs of traffic facilities, government, and hotel are of great significance for dominating the spatial communities in both cities. It implies that experts could monitor the spatial distribution of urban mobility flows by observing the distribution of POIs, and urban planners could influence the spatial communities of mobility flows by changing the locations of these categories of POIs or adding new POIs of these categories.

In the future, we will further study the formation mechanism of the spatial communities of mobility flows. Meanwhile, we are going to employ other mobility data sources,

such as cell-tower traces, and check-ins in location-based services.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61772230, 61702215, and 61571444); Jilin Province Science and Technology Development Program (20160204021GX); Guangdong Natural Science Foundation (2016A030310072); Science Research Cultivation Project of Shenzhen Institute of Information Technology (ZY201718); MOE (Ministry of Education in China) Project of Humanities and Social Sciences (17YJCZH261); Special Innovation Project of Guangdong Education Department (2017GKTSCX063); Natural Science Foundation of Jilin Province (20180101332JC); and Science-Technology Research Foundation of Jilin Province (2016104).

References

[1] M. Barthélemy, "Spatial networks," *Physics Reports*, vol. 499, no. 1-3, pp. 1–101, 2011.

[2] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and POIs," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pp. 186–194, August 2012.

[3] Z. Liu, T. Li, P. Li, C. Jia, and J. Li, "Verifiable searchable encryption with aggregate keys for data sharing system," *Future Generation Computer Systems*, vol. 78, pp. 778–788, 2018.

[4] Z. Liu, Z. Wu, T. Li, J. Li, and C. Shen, "GMM and CNN hybrid method for short utterance speaker recognition," *IEEE Transactions on Industrial Informatics*, pp. 1-1, 2018.

[5] B. Li, Y. Huang, Z. Liu, J. Li, Z. Tian, and S. Yiu, "HybridORAM: Practical oblivious cloud storage with constant bandwidth," *Information Sciences*, 2018.

[6] Z. Liu, X. Chen, J. Yang, C. Jia, and I. You, "New order preserving encryption model for outsourced databases in cloud environments," *Journal of Network and Computer Applications*, vol. 59, pp. 198–207, 2016.

[7] X. Zhang, Y. Tan, C. Liang, Y. Li, and J. Li, "A Covert Channel Over VoLTE via Adjusting Silence Periods," *IEEE Access*, vol. 6, pp. 9292–9302, 2018.

[8] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, "Significant permission identification for machine learning based android malware detection," *IEEE Transactions on Industrial Informatics*, pp. 1-1, 2018.

[9] L. Fan, X. Lei, N. Yang, T. Q. Duong, and G. K. Karagiannidis, "Secure Multiple Amplify-and-Forward Relaying with Cochannel Interference," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 8, pp. 1494–1505, 2016.

[10] L. Fan, X. Lei, N. Yang, T. Q. Duong, and G. K. Karagiannidis, "Secrecy Cooperative Networks with Outdated Relay Selection over Correlated Fading Channels," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7599–7603, 2017.

[11] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, vol. 6, pp. 20632–20640, 2018.

[12] W. Lin, Z. Wu, L. Lin, A. Wen, and J. Li, "An ensemble random forest algorithm for insurance big data analysis," *IEEE Access*, vol. 5, pp. 16568–16575, 2017.

[13] Y. Li, G. Wang, L. Nie, Q. Wang, and W. Tan, "Distance metric optimization driven convolutional neural network for age invariant face recognition," *Pattern Recognition*, vol. 75, pp. 51–62, 2018.

[14] X. Huang, Y. Zhao, C. Ma, J. Yang, X. Ye, and C. Zhang, "TrajGraph: A Graph-Based Visual Analytics Approach to Studying Urban Network Centralities Using Taxi Trajectory Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 160–169, 2016.

[15] X. Liu, L. Gong, L. Wu, and Y. Liu, "Inferring trip purposes and uncovering travel patterns from taxi trajectory data," *Cartography and Geographic Information Science*, vol. 43, no. 2, pp. 103–114, 2016.

[16] X. Liu, L. Gong, Y. Gong, and Y. Liu, "Revealing travel patterns and city structure with taxi trip data," *Journal of Transport Geography*, vol. 43, pp. 78–90, 2015.

[17] L. Alexander, S. Jiang, M. Murga, and M. C. González, "Origin-destination trips by purpose and time of day inferred from mobile phone data," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 240–250, 2015.

[18] S. Grauwin, S. Sobolevsky, S. Moritz, I. Gódor, and C. Ratti, "Towards a comparative science of cities: Using mobile traffic records in New York, London, and Hong Kong," *Computational Approaches for Urban Environments*, pp. 363–387, 2015.

[19] Z. Du, B. Yang, and J. Liu, "Understanding the spatial and temporal activity patterns of subway mobility flows," https://arxiv.org/abs/1702.02456.

[20] P. Li, J. Li, Z. Huang et al., "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.

[21] X. Liu, K.-K. R. Choo, and R. H. Deng, "Efficient and privacy-preserving outsourced calculation of rational numbers," *IEEE Transactions on Dependable & Secure Computing*, vol. 15, no. 1, pp. 27–39, 2018.

[22] X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2401–2414, 2016.

[23] X. Liu, R. H. Deng, W. DIng, R. Lu, and B. Qin, "Privacy-preserving outsourced calculation on floating point numbers," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2513–2527, 2016.

[24] Z. Liu, Y. Huang, J. Li, X. Cheng, and C. Shen, "DivORAM: Towards a practical oblivious RAM with variable block size," *Information Sciences*, vol. 447, pp. 1–11, 2018.

[25] F. Simini, M. C. González, A. Maritan, and A.-L. Barabási, "A universal model for mobility and migration patterns," *Nature*, vol. 484, no. 7392, pp. 96–100, 2012.

[26] Y. Ren, M. Ercsey-Ravasz, P. Wang, M. C. González, and Z. Toroczkai, "Predicting commuter flows in spatial networks using a radiation model based on temporal ranges," *Nature Communications*, vol. 5, article no. 5347, 2014.

[27] C. Zhong, S. M. Arisona, X. Huang, M. Batty, and G. Schmitt, "Detecting the dynamics of urban structure through spatial

network analysis," *International Journal of Geographical Information Science*, vol. 28, no. 11, pp. 2178–2199, 2014.

[28] M. E. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[29] S. Jiang, J. Ferreira Jr., and M. C. Gonzalez, "Discovering urban spatial-temporal structure from human activity patterns," in *Proceedings of the International Workshop on Urban Computing, UrbComp 2012 - Held in Conjunctionwith KDD 2012*, pp. 95–102, China, August 2012.

[30] M. Zaltz Austwick, O. O'Brien, E. Strano, and M. Viana, "The Structure of Spatial Networks and Communities in Bicycle Sharing Systems," *PLoS ONE*, vol. 8, no. 9, Article ID e74685, 2013.

[31] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 3, article 38, 2014.

[32] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '11)*, pp. 89–98, ACM, September 2011.

[33] S. Sarkar, S. Chawla, S. Ahmad et al., "Effective Urban Structure Inference from Traffic Flow Dynamics," *IEEE Transactions on Big Data*, vol. 3, no. 2, pp. 181–193, 2017.

[34] C. Zhong, E. Manley, S. Müller Arisona, M. Batty, and G. Schmitt, "Measuring variability of mobility patterns from multiday smart-card data," *Journal of Computational Science*, vol. 9, pp. 125–130, 2015.

[35] J. C. Ying, W. C. Lee, and V. S. Tseng, "Mining geographic-temporal-semantic patterns in trajectories for location prediction," *Acm Transactions on Intelligent Systems & Technology*, vol. 5, no. 1, pp. 1–33, 2014.

[36] C. Zhang, Z. Du, M. D. Parmar, and Y. Bai, "Pocket-switch-network based services optimization in crowdsourced delivery systems," *Computers and Electrical Engineering*, vol. 62, pp. 53–63, 2017.

[37] C. Zhang et al., "Particle swarm optimization algorithm based on ontology model to support cloud computing applications," *Journal of Ambient Intelligence & Humanized Computing*, vol. 7, no. 5, pp. 1–6, 2015.

[38] N. Mou, J. Li, L. Zhang, W. Liu, and Y. Xu, "Spatio-temporal characteristics of resident trip based on POI and od data of float car in Beijing," in *Proceedings of the ISPRS Geospatial Week 2017*, pp. 99–105, China, September 2017.

[39] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 76, no. 3, Article ID 036106, 2007.

[40] K. Cooper and M. Barahona, "Role-based similarity in directed networks," *Physics*, 2012, http://arxiv.org/abs/1021.2726.

[41] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," *The European Physical Journal Special Topics*, vol. 178, no. 1, pp. 13–23, 2009.

[42] Q. Cai, L. Ma, M. Gong, and D. Tian, "A survey on network community detection based on evolutionary computation," *International Journal of Bio-Inspired Computation*, vol. 8, no. 2, pp. 84–98, 2016.

[43] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[44] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.

[45] S. Sobolevsky, R. Campari, A. Belyi et al., "General optimization technique for high-quality community detection in complex networks," *Physical Review E Statistical Nonlinear & Soft Matter Physics*, vol. 90, no. 1, Article ID 012811, p. 10, 2014.

[46] T. Chou, C. Hsu, and M. Chen, "A fuzzy multi-criteria decision model for international tourist hotels location selection," *International Journal of Hospitality Management*, vol. 27, no. 2, pp. 293–301, 2008.

[47] Q. Y. Meng, J. Chen, and Y. C. Guo, "The analysis of the relationship between urban pubic traffic facilities distribution and population distribution-a case study of Shanghao," *Northwest Population Journal*, vol. 5, 2005.

[48] Y. Ohsawa, "Location-Allocation Models of Some Traffic Facilities," *Geographical Analysis*, vol. 21, no. 2, pp. 134–146, 1989.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5401890, 9 pages
https://doi.org/10.1155/2018/5401890

Research Article

An Efficient Identity-Based Proxy Blind Signature for Semioffline Services

Hongfei Zhu,[1] **Yu-an Tan**,[1] **Liehuang Zhu**,[1] **Quanxin Zhang**,[1] **and Yuanzhang Li**[1,2]

[1]*School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China*
[2]*Research Center of Massive Language Information Processing and Cloud Computing Application, Beijing 100081, China*

Correspondence should be addressed to Yuanzhang Li; popular@bit.edu.cn

Received 6 February 2018; Accepted 12 March 2018; Published 9 May 2018

Academic Editor: Ximeng Liu

Fog computing extends the cloud computing to the network edge and allows deploying a new type of semioffline services, which can provide real-time transactions between two entities, while the central cloud server is offline and network edge devices are online. For an e-payment system and e-voting with such feature, proxy blind signature is a cornerstone to protect users' privacy. However, the signature based on number theorem, such as hard mathematical problems on factoring problem, discrete logarithm problem, and bilinear pairings, cannot defeat quantum computers attack. Meanwhile, these schemes need to depend on complex public key infrastructure. Thus, we construct an identity-based proxy blind signature scheme based on number theorem research unit lattice, which can defeat quantum computers attack and does not need to depend on public key infrastructure. The security of the proposed scheme is dependent on Ring-Small Integer Solution problem over number theorem research unit lattice. The proposed scheme meets the properties of blind signature and proxy signature. Then we compare the proposed scheme with other existing proxy blind signature schemes; the result shows that the proposed scheme outperforms ZM scheme except in proxy signer's signature size and can be more secure than TA scheme and MMHP scheme.

1. Introduction

Fog computing was initially introduced by Cisco, which can overcome cloud computing's disadvantages, such as non-real-time service and long delay [1–3]. More specifically, fog computing adds a new layer between cloud server and terminal user [4, 5]; that is, fog servers can be access point, base station, router, or mobile equipment [6–9]. Thus, the semioffline e-payment system can be deployed by utilizing the advantages of fog computing model [10, 11].

In order to defend user's privacy in offline e-payment system, blind signature (BS) is crucial for that it never permits signer to sign on a plaintext before knowing its content [12]. Therefore, BS can protect user privacy during the transactions [13] instead of encrypting the data and searching on the ciphertext [14]. However, this system is deployed in real environment; it will use distributed architecture [15]. The original signer should authorize an agent to sign for himself. Then a proxy signature (PS) should be used in e-payment system, since proxy signer can satisfy this requirement [16]. Combining those two types of schemes together, a new proxy blind signature (PBS) was proposed, which meets the properties of those two signature schemes. After that, many PBS schemes were constructed by scholars.

However, most of the PBS schemes are based on number theory, such as discrete logarithm problem (DLP) and bilinear pairings. These schemes are considered to be insecure to resist the quantum computer attack. Therefore, the e-payment and e-voting systems in the cloud still face the threat from quantum computer attack [17]. Meanwhile, these schemes need to rely on complex public key infrastructure (PKI) [18, 19]. In conclusion, these schemes based on number theorem cannot defeat the quantum computers attack according to the recent research results.

Therefore, the lattice-based PBS schemes become one alternative solution, since they are sufficient enough and able to resist quantum computer attack [20, 21]. Besides, if lattice-based PBS schemes can combine with identity-based cryptography (called IDPBS), they can overcome the shortcomings of traditional PBS schemes, such as relying on

complex PKI [22]. Meanwhile, they can transfer less data than biological recognition methods during the transactions [23, 24].

Zhu et al. presented a new lattice-based BS [20], which can be secure enough for cloud services. However, this scheme has to be combined with proxy signature in practice. Combining BS scheme and IDPS scheme, we initially present an IDPBS on number theorem research unit lattice (IDPBS-NTRU), which can defeat quantum computer attack.

(1) Inspired by [25], a new IDPBS-NTRU scheme is proposed based on NTRU lattice, which can make semioffline e-payment and e-voting systems deployed in fog computing model secure enough to resist quantum computer attack.

(2) The proposed IDPBS-NTRU scheme is proven to be secure. That is, the proposed scheme is correct, blind, unforgeable, verifiable, strong identifiable, strong undeniable, and key-dependent.

(3) The proposed IDPBS-NTRU is compared with the existing IDPBS schemes in terms of performances. The result shows that it outperforms the ZM scheme except in proxy signer's signing key size, and it is more secure than TA and MMHP schemes.

The paper is introduced as follows. Section 2 introduces the background knowledge about NTRU lattice and main key technology. Section 3 introduces the security model for IDPBS. Section 4 shows that the proposed IDPBS is proven to be secure and it is compared with other IDPBS schemes in terms of performances. At last, Section 5 draws the conclusions.

2. Related Works

IDPBS Schemes Based on DLP. In 2011, Beura et al. proposed a new proxy blind signature based on DLP; their scheme satisfies the properties of blind signature and proxy signature. This scheme is more secure and efficient than factoring signature schemes [26]. To improve the efficiency, Tan et al. introduced a couple of PBS schemes; both of them were constructed on Schnorr blind signature. However, Sun et al. pointed that both of them were not unforgeable and unlinkable [27]. However, in 2014, Wang and Liao proved that the schemes proposed by Oo et al. and Beura et al. did not satisfy unlinkability [28]. In 2013, Tan proposed a PBS based on DLP, which did not depend on PKI [29] and was proven to be secure in the random oracle [30]. However, most of these schemes are dependent on PKI and are not strictly proven to be secure.

IDPBS Schemes Based on Bilinear Pairings. In 2003, Zhang et al. proposed a new proxy blind signature based on bilinear pairings, which satisfies distinguishability, verifiability, strong nonforgeability, strong identifiability, strong nondeniability, and prevention of misuse. Meanwhile, this scheme did not depend on public key infrastructure (PKI) [31]. Later, Li et al. introduced a new PBS, which was also constructed on bilinear pairings; it was independent of PKI [32]. However, these schemes are inefficient and are not proven to be secure.

IDPBS Schemes Based on Lattice. In 2014, Zhang and Ma initially proposed a proxy blind signature on lattice; it does not need to depend on PKI; its security is based on short integer solution problem. However, this scheme is still inefficient. [33].

3. Preliminaries

In the beginning, we will define the denotations that will be used all over the paper in Denotations.

3.1. NTRU Lattice and Rejection Sampling on Lattice

Definition 1 (NTRU lattice). The notations are defined as f, $g \in \mathbf{R}$ and $h = gf^{-1} \bmod q$; after that, the NTRU lattice can be defined as $\mathscr{L}_{h,q} = \{u, v \in \mathbf{R}^2 : u + vh = 0 \bmod q\}$. That is, $\mathscr{L}_{h,q}$ is on behalf of a $\mathbb{R}^{2N}$ full-rank lattice whose basis is $\begin{pmatrix} -\mathbf{T}_N(h) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{pmatrix}$, $\mathbf{I}_N$ denotes a unit matrix, $\mathbf{O}_N$ denotes a null matrix, and $\mathbf{T}_N(h)$ denotes an anticirculant matrix $\begin{pmatrix} h_0 & h_1 & \cdots & h_{N-1} \\ -h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \vdots & \vdots \\ -h_1 & -h_2 & \cdots & h_0 \end{pmatrix}$.

Definition 2 ($R\text{-}SIS^{\kappa}_{q,1,2,\beta}$ on NTRU lattice). Small f and g can be sampled from $D_{\mathbb{Z}^N,\sigma}$ ($f, g \bmod q \in \mathbf{R}_q^{\times}$); then $\mathscr{L}_{h,q} = (h, 1) \in \mathbf{R}_q^{1\times 2}$ and $h = gf^{-1}$ can be obtained by using Algorithm 3 in [25]. Therefore, R-SIS on NTRU means finding $\mathbf{z_1}, \mathbf{z_2}$ satisfying $\mathscr{L}_{h,q}(\mathbf{z}_1, \mathbf{z}_2)^T = \mathbf{0} \bmod q$ and $\|(\mathbf{z_1}, \mathbf{z_2})\| \le \beta$.

Theorem 3 (rejection sampling theorem). *V denotes one subset of $\mathbb{Z}^m$, the norms of V's elements are less than constant T, $\sigma = \omega(T\sqrt{\log m}) \in \mathbb{R}$, ($M$ is invariable), and $h : V \to \mathbb{R}$ is a probability distribution. Two algorithms are as follows: One is*

$$\begin{gathered} \mathbf{v} \longleftarrow h; \\ \mathbf{w} \longleftarrow D^N_{\mathbf{v},\sigma}; \\ \textit{get } (\mathbf{w}, \mathbf{v}) \textit{ with probability } \min\left(\frac{D^N_{\sigma}(\mathbf{w})}{MD^N_{\mathbf{v},\sigma}(\mathbf{w})}, 1\right). \end{gathered} \tag{1}$$

The other is

$$\begin{gathered} \mathbf{v} \longleftarrow h; \\ \mathbf{w} \longleftarrow D^N_{\sigma}; \\ \textit{get } (\mathbf{w}, \mathbf{v}) \textit{ with probability } \frac{1}{M}. \end{gathered} \tag{2}$$

Then the distribution of first algorithm will not exceed the second one's statistical distance $2^{-\omega(\log N)}/M$. Moreover, The first one will export something with probability at least $(1 - 2^{-\omega(\log N)})/M$.

3.2. The Definitions of IDPBS Scheme. An IDPBS consists of seven algorithms $(ST_\varepsilon, KE_\varepsilon, DG_\varepsilon, BS_\varepsilon, PS_\varepsilon, UB_\varepsilon, PV_\varepsilon)$

$Setup: ST_\varepsilon(1^N) \to (params, mk)$
$Key\ \ Extraction: KE_\varepsilon(params, mk, id_i) \to sk_{id_i}, i = \mathcal{O}, \mathcal{P}$
$Delegation\ \ Generation:$
$DG_\varepsilon(id_\mathcal{O}, sk_{id_\mathcal{O}}, \omega) \to \psi$
$\xrightarrow{\psi}$
proxy signer $\mathcal{P}$
Blind (user):
$BS_\varepsilon(m, f) \to m'$
$\xrightarrow{m'}$
Proxy Signature (proxy signer):
$PS_\varepsilon(m', id_P, sk_P, id_O, \psi) \to \sigma'$
$\xleftarrow{\sigma'}$
Unblind (user):
$UB(\sigma', f) \to \sigma$
Proxy Verify:
$PV_\varepsilon(id_\mathcal{O}, id_\mathcal{P}, \omega, m, \psi, \sigma) \to true/false$

Algorithm 1: General IDPBS scheme.

[12, 36, 37]. TTP will execute $ST_\varepsilon(1^N)$ to produce public parameters and keys [29, 38, 39]. The formal definition is presented as follows (Algorithm 1):

(i) $ST_\varepsilon(1^n)$ outputs $params$ and $mk = (msk, mpk)$.

(ii) $KE_\varepsilon(params, msk, id_i)$ outputs sk_{id_i} for $\mathcal{O}$ and $\mathcal{P}$ ($i = \mathcal{O}$ or $\mathcal{P}$).

(iii) $DG_\varepsilon(id_O, sk_{id_O}, w)$ outputs ψ for $\mathcal{P}$.

(iv) Proxy blind signature: $\mathcal{U}$ interacts with $\mathcal{P}$ according to the following protocol:

(1) $BS(m, f)$: $\mathcal{U}$ blinds m to m' by using f and then sends m' to $\mathcal{P}$.

(2) $PS_\varepsilon(m', id_P, sk_P, id_O, \psi)$: $\mathcal{P}$ signs on m' using sk_{id_P} and sends the signature σ' to $\mathcal{U}$.

(3) $UB(\sigma', f)$: $\mathcal{U}$ unblinds σ' by using f and outputs the blind signature σ.

(v) $PV_\varepsilon(id_O, id_P, w, m, \psi, \sigma)$: if ψ, σ are valid, the algorithm outputs true. Otherwise it outputs false.

An IDPBS scheme should meet the following six properties. The details can be seen in [20, 33, 40–42].

(1) Blindness. P^* are denoted as an adversary who can control the proxy signer. P^* chooses two messages m_0 and m_1. Then a random bit $i \in \{0, 1\}$ is chosen in the game. m_0 and m_1 are randomly denoted as m_i and m_{1-i}. These two messages are, respectively, used as two user's inputs. After that, P^* will adaptively and parallelly interact with two honest users according to the signature protocol. Finally, two users output σ_i and σ_{1-i} respectively. Then σ_i and σ_{1-i} ordered by m_i and m_{1-i} are delivered to P^*; after that, P^* will output $i' \in \{0, 1\}$.

(2) One More Unforgeability. $\mathcal{P}$ can generate a legal proxy σ instead of $\mathcal{O}$. However, $\mathcal{O}$ and all the other entities fail to generate a legal signature. The game is presented as follows [33]: Adv_{U^*}, the advantage of U^*, is denoted as success probability in Algorithm 3. If no adversary can win Algorithm 3 at minimum with negligible probability η in time t, then it satisfies one more unforgeability [31].

(3) Verifiability. $\mathcal{V}$ can check whether σ is delegated by $\mathcal{O}$.

(4) Strong Identifiability. Any $\mathcal{V}$ can determine $\mathcal{P}$'s identity once he receives the proxy signature tuple.

(5) Strong Undeniability. $\mathcal{P}$ cannot refuse to admit it once he creates the proxy signature σ.

(6) Key Dependence. $\mathcal{P}$ can sign on a message if and only if he has the authorization from $\mathcal{O}$.

4. Proposed IDPBS-NTRU Scheme

Here, we introduce a novel IDPBS-NTRU $\varepsilon = (ST_\varepsilon, KE_\varepsilon, DG_\varepsilon, BS_\varepsilon, PS_\varepsilon, UB_\varepsilon, PV_\varepsilon)$, which can be seen in Algorithm 4. The details are as follows.

(1) $ST_\varepsilon(1^N)$. $q = \mathrm{Poly}(N)$, $\varepsilon \in (0, \ln N/\ln q)$, and $s = \widetilde{\Omega}(N^{3/2}\sigma)$. If $N > 2, \sigma = N\sqrt{(\ln(8Nq)}q^{1/2+\varepsilon}, q^{1/2-\varepsilon} = \widetilde{\Omega}(n^{7/2})$. If $N = 2, \sigma = N\sqrt{\ln(8Nq)}q^{1/2+\varepsilon}, q^{1/2-\varepsilon} = \widetilde{\Omega}(N^3)$. $mk = (msk, mpk)$ can be obtained as below [25].

The algorithm takes samples f and g from $D_{\mathbb{Z}^N, s}$. Here, f, $g \bmod q \notin \mathbf{R}_q^\times$, $\|f\|, \|g\| \le \sigma\sqrt{N}$, and $\langle f, g\rangle \in \mathbf{R}$. After that, the algorithm can get $F_1, G_1 \in \mathbf{R}$ according to the equation $fG_1 - gF_1 = 1$. Given $F_q = qF_1$ and $G_q = qG_1$, (F_q, G_q) can be obtained from $(f, g), (xf, xg), \ldots, (x^{N-1}f, x^{N-1}g)$ according to Babai algorithm [43]. Then there exists $(F, G) = (F_q, G_q) - k(f, g)$. If $\|(F, G)\| \le N\sigma$, then the algorithm outputs system parameters $paras = (q, \varepsilon, s)$; the master private-key **msk** and master public-key mpk are as follows:

$$\mathbf{msk} = \mathbf{B} = \begin{pmatrix} \mathbf{T}(f) & \mathbf{T}(g) \\ \mathbf{T}(F) & \mathbf{T}(G) \end{pmatrix}, \quad mpk = h = gf^{-1} \in \mathbf{R}_q^{\times}. \tag{3}$$

(2) $KE_\varepsilon(paras, id_i, \boldsymbol{msk})$. The algorithm executes (4) to get an N-dimension matrix $\mathbf{t}$; then the algorithm executes (6) and outputs sk_{id_i} according to corresponding id_i $(i = \mathcal{O}, \mathcal{P})$ [25].

$$\mathbf{t} \longleftarrow H_1(id_i), \tag{4}$$

$$sk_{id_i} = \left(s_{i_1}, s_{i_2}\right) \longleftarrow [(t,0) - \text{Gausssian}(\mathbf{msk}, \sigma, (t,0))], \quad s_{i_1} + s_{i_2} h = t. \tag{5}$$

(3) DG_ε. ω is denoted as a warrant. $\mathcal{O}$ will execute this algorithm to generate a valid delegation.

(i) The algorithm chooses $\mathbf{y}_1, \mathbf{y}_2 \in D_{\mathbb{Z}^N,s}$ at random.

(ii) The algorithm executes (6) to get an N-dimension matrix $\mathbf{u}$.

(iii) The algorithm executes (7) and (8) to generate valid delegations $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$. Here, the algorithm uses the rejection sampling theorem to keep the delegation independent on $\mathcal{O}$'s secret keys $\mathbf{s}_{\mathcal{O}_1}$ and $\mathbf{s}_{\mathcal{O}_2}$.

(iv) $\mathcal{O}$ sends $(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \mathbf{u}, \omega)$ to $\mathcal{P}$.

$$\mathbf{u} = H_2(\mathbf{y}_1 + h\mathbf{y}_2, \omega), \tag{6}$$

$$\boldsymbol{\sigma}_1 = \mathbf{y}_1 + \mathbf{s}_{\mathcal{O}_1}\mathbf{u}, \tag{7}$$

$$\boldsymbol{\sigma}_2 = \mathbf{y}_2 + \mathbf{s}_{\mathcal{O}_2}\mathbf{u}. \tag{8}$$

(4) BS_ε. m is a plaintext. $\mathcal{U}$ will execute this algorithm to generate a blind message, which needs to be signed by $\mathcal{P}$.

(i) The algorithm will randomly select $\mathbf{y}_3, \mathbf{y}_4, \boldsymbol{\alpha}, \boldsymbol{\gamma} \in D_{\mathbb{Z}^N,s}$.

(ii) The algorithm executes (9) to get an N-dimension $\mathbf{e}$.

(iii) The algorithm executes (10) to blind $\mathbf{e}$.

(iv) $\mathcal{U}$ sends $(\mathbf{y}_3, \mathbf{y}_4, \mathbf{e}^*)$ to a proxy signer $\mathcal{P}$.

$$\mathbf{e} = H_3(\mathbf{y}_3 + h\mathbf{y}_4 + h\boldsymbol{\gamma} + \boldsymbol{\alpha} - \boldsymbol{\alpha}H(id), m), \tag{9}$$

$$\mathbf{e}^* = \mathbf{e} - \boldsymbol{\alpha}. \tag{10}$$

(5) PS_ε. The proxy signer $\mathcal{P}$ will execute this algorithm to sign on the blinded message.

(i) The algorithm validates whether (11) and (12) are true. If either is false, $\mathcal{P}$ aborts the algorithm. Otherwise, it continues.

(ii) The algorithm will execute (13) and (14). Here, the rejection sample theory is used to keep the proxy signatures $\boldsymbol{\sigma}_3$ and $\boldsymbol{\sigma}_4$ independent on $\mathcal{P}$'s secret keys $\mathbf{s}_{\mathcal{P}_1}$ and $\mathbf{s}_{\mathcal{P}_2}$.

(iii) The algorithm outputs the tuple $(m, \omega, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \mathbf{u}, \boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4, \mathbf{e})$.

$$\|(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)\| \le 2s\sqrt{2N}, \tag{11}$$

$$H_2(h\boldsymbol{\sigma}_2 + \boldsymbol{\sigma}_1 - H_1(id_{\mathcal{O}})\mathbf{u}, \omega) = \mathbf{u}, \tag{12}$$

$$\boldsymbol{\sigma}_3 = \mathbf{y}_3 + s_{\mathcal{P}_1}\mathbf{e}, \tag{13}$$

$$\boldsymbol{\sigma}_4 = \mathbf{y}_4 + s_{\mathcal{P}_2}\mathbf{e}. \tag{14}$$

(6) UB_ε. $\mathcal{U}$ will execute the algorithm to unblind the proxy signature.

(i) The algorithm executes (15) to unblind the proxy signature tuple.

(ii) $\mathcal{U}$ outputs the signature tuple $(\boldsymbol{\sigma}_1, \mathbf{m}, \boldsymbol{\sigma}_2, \boldsymbol{\omega}, \mathbf{u}, \boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4, \mathbf{e})$.

$$\begin{aligned} \boldsymbol{\sigma}_3 &= \boldsymbol{\sigma}_3^* + \boldsymbol{\alpha}, \\ \boldsymbol{\sigma}_4 &= \boldsymbol{\sigma}_4^* + \boldsymbol{\gamma}. \end{aligned} \tag{15}$$

(7) $PV_\varepsilon(\boldsymbol{\sigma}_1, m, \boldsymbol{\sigma}_2, \omega, \mathbf{u}, \boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4, \mathbf{e}, id_{\mathcal{O}}, id_{\mathcal{P}})$. $\mathcal{V}$ will execute this algorithm to validate whether the signature tuple satisfies (16). If all the equations mentioned above are true, the signatures are valid. Otherwise, they are invalid.

$$\begin{aligned} &\|(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)\| \le 2s\sqrt{2N}, \\ &H_2(h\boldsymbol{\sigma}_2 + \boldsymbol{\sigma}_1 - H_1(id_{\mathcal{O}})u, w) = \mathbf{u}, \\ &\|(\boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4)\| \le 2s\sqrt{2N}, \\ &H_3(h\boldsymbol{\sigma}_4 + \boldsymbol{\sigma}_3 - H_1(id_{\mathcal{P}})\mathbf{e}, m) = \mathbf{e}. \end{aligned} \tag{16}$$

5. Security and Performance Comparison

5.1. Security

(1)

Theorem 4 (correctness). *The IDPBS-NTRU scheme is correct.*

Proof. According to the construction of our IDPBS scheme, we can get

$$\begin{aligned} &h * \boldsymbol{\sigma}_2 + \boldsymbol{\sigma}_1 - H(id_{\mathcal{O}}) * \mathbf{u} \\ &= h(\mathbf{y}_2 + \mathbf{s}_{\mathcal{O}_2}\mathbf{u}) + \mathbf{y}_1 + \mathbf{s}_{\mathcal{O}_1}\mathbf{u} - H(id_{\mathcal{O}}) * \mathbf{u} \\ &= \mathbf{y}_1 + h\mathbf{y}_2. \end{aligned} \tag{17}$$

Therefore, $H_2(h * \boldsymbol{\sigma}_2 + \boldsymbol{\sigma}_1 - H(id_{\mathcal{O}}) * \mathbf{u}, m) = \mathbf{u}$.

$$\begin{aligned} &h\boldsymbol{\sigma}_4 + \boldsymbol{\sigma}_3 - H_1(id_{\mathcal{P}})\mathbf{e} \\ &= h(\boldsymbol{\sigma}_4^* + \boldsymbol{\gamma}) + \boldsymbol{\sigma}_3^* + \boldsymbol{\alpha} - H(id_{\mathcal{P}}) * \mathbf{e} \\ &= h(\mathbf{y}_4 + s_{\mathcal{P}_2}\mathbf{e}^* + \boldsymbol{\gamma}) + \mathbf{y}_3 + \boldsymbol{\alpha} + s_{\mathcal{P}_1}\mathbf{e}^* - H(id_{\mathcal{P}}) \\ &\quad * \mathbf{e} = \mathbf{y}_3 + h\mathbf{y}_4 + h\boldsymbol{\gamma} + \boldsymbol{\alpha} - \boldsymbol{\alpha}H_1(id_{\mathcal{P}}). \end{aligned} \tag{18}$$

Thus, $H_3(h\boldsymbol{\sigma}_4 + \boldsymbol{\sigma}_3 - H_1(id_{\mathcal{P}})\mathbf{e}, m) = \mathbf{e}$. □

$i \in_{\$} \{0,1\}$
$(params, msk) \leftarrow ST(1^n)$
$sk_{id_b} \leftarrow EX(params, id_b, msk), b = \mathcal{O}$ or $\mathcal{P}$
$\psi \leftarrow DG_\varepsilon(id_{\mathcal{O}}, sk_{id_{\mathcal{O}}}, \omega)$
$(m_0, m_1, \omega, \psi, state_{find}) \leftarrow_{\$} \mathcal{P}^*(find, sk_{id_{\mathcal{P}}}, id_{\mathcal{P}}, id_{\mathcal{O}}, \psi)$
$state_{issue} \leftarrow_{\$} \mathcal{P}^{*<.,\mathcal{U}(id_{\mathcal{P}},m_i)^1>,<.,\mathcal{U}(id_{\mathcal{P}},m_{1-i})^1>}(issue, state_{find})$
σ_i, σ_{1-i} are respectively $\mathcal{U}(id_{\mathcal{P}}, m_i), \mathcal{U}(id_{\mathcal{P}}, m_{1-i})$'s output
if $\sigma_0 \neq$ fail and $\sigma_1 \neq fail$ **then**
  $i' \leftarrow_{\$} \mathcal{P}^*(guess, \sigma_0, \sigma_1, state_{issue})$
else
  $i' \leftarrow_{\$} \mathcal{P}^*(guess, fail, fail, state_{issue})$
end if
return true iff $i' = i$

Algorithm 2: $\text{Expt}^{bd}_{\mathcal{S}^*}(n)$.

$(params, msk) \leftarrow ST(1^n)$
$sk_{id_b} \leftarrow EX(params, id_b, msk), b = \mathcal{O}$ or $\mathcal{P}$
$\psi \leftarrow DG_\varepsilon(id_{\mathcal{O}}, sk_{id_{\mathcal{O}}}, \omega)$
$\{(m_1, \sigma_1), ..., (m_k, \sigma_k)\} \leftarrow_{\$} \mathcal{U}^{*h(.),<\mathcal{S}(sk_{id_{\mathcal{P}}}),>^\infty}(id_{\mathcal{P}})$
l is the successful interaction number between $\mathcal{U}^*$ and $\mathcal{P}$
return true iff
  $m_i \neq m_j$ for $1 \leq i < j \leq k$ and
  $VF(m_i, \sigma_i, id) = 1$ and
  $l + 1 = k$

Algorithm 3: $\text{Expt}^{omf}_{\mathcal{U}^*}(n)$.

(2)

Theorem 5 (blindness). *The IDPBS-NTRU scheme satisfies blindness.*

Proof. As shown in Algorithm 2, A random bit $i \in \{0,1\}$ which is kept secret from P^*. Then P^* chooses two messages m_0 and m_1. m_0 and m_1 are randomly denoted as m_i and m_{1-i}. m_i and m_{1-i} are the inputs of two honest users, respectively. P^* adaptively and parallelly interacts with two honest users, respectively. Finally, these two honest users output σ_i and σ_{1-i}, respectively. The sequence σ_i and σ_{1-i} ordered by m_i and m_{1-i} will be sent to P^*. P^* will output a bit $i' \in \{0,1\}$.

In the process of signature protocol, all intermediate results do not depend on m; thus it is enough to analyze $\mathbf{e}^*$, $\mathbf{y}_3$, $\mathbf{y}_4$, $\boldsymbol{\sigma}_1$, $\boldsymbol{\sigma}_2$, $\mathbf{u}$, $\boldsymbol{\omega}$, $\boldsymbol{\sigma}_3^*$, and $\boldsymbol{\sigma}_4^*$.

To $\mathbf{e}^*$, the statistical distance is presented as follows:

$$\Delta\left(\mathbf{e}_i^*, \mathbf{e}_{1-i}^*\right) = \frac{1}{2} \sum_{\mathbf{e}^{*'} \in D_{\mathbb{Z}^N, s}} \left| \Pr\left(\mathbf{e}_i^* = \mathbf{e}^{*'}\right) - \Pr\left(\mathbf{e}_{1-i}^* = \mathbf{e}^{*'}\right) \right|. \quad (19)$$

□

Since α is chosen at random, next we obtain the equations $\Pr(\mathbf{e}_i^* = \mathbf{e}^{*'}) = 1/2$ and $\Pr(\mathbf{e}_{1-i}^* = \mathbf{e}^{*'}) = 1/2$. Therefore, we obtain $\Delta(\mathbf{e}_i^*, \mathbf{e}_{1-i}^*) = 0$.

In the same way, we can obtain $\Delta(\mathbf{y}_3^i, \mathbf{y}_3^{1-i}) = 0$, $\Delta(\mathbf{y}_4^i, \mathbf{y}_4^{1-i}) = 0$, $\Delta(\boldsymbol{\sigma}_1^{i*}, \boldsymbol{\sigma}_1^{1-i*}) = 0$, $\Delta(\boldsymbol{\sigma}_2^{i*}, \boldsymbol{\sigma}_2^{1-i*}) = 0$, $\Delta(\mathbf{u}^{i*}, \mathbf{u}^{1-i*}) = 0$, $\Delta(\boldsymbol{\omega}^{i*}, \boldsymbol{\omega}^{1-i*}) = 0$, $\Delta(\boldsymbol{\sigma}_3^{i*}, \boldsymbol{\sigma}_3^{1-i*}) = 0$, and $\Delta(\boldsymbol{\sigma}_4^{i*}, \boldsymbol{\sigma}_4^{1-i*}) = 0$. Therefore, $\mathcal{S}^*$ cannot distinguish m among $\mathbf{e}^*$, $\mathbf{y}_3$, $\mathbf{y}_4$, $\boldsymbol{\sigma}_1$, $\boldsymbol{\sigma}_2$, $\mathbf{u}$, $\boldsymbol{\omega}$, $\boldsymbol{\sigma}_3^*$, and $\boldsymbol{\sigma}_4^*$; that is, $\mathcal{S}^*$ can win this experiment with probability 1/2. Thus, the theorem is proven.

We denote δ_1, δ_2, δ_3, δ_4, δ_5, and δ_6 as the cost functions of H_1, H_2, H_3 hash oracle, extract Oracle, DG oracle, and signature oracle, respectively, η as a nonnegligible probability, Θ as a polynomial time algorithm, and Γ as a polynomial time forger.

(3)

Theorem 6 (one more unforgeability). *If Γ is able to generate a legal IDPBS signature with η in t successfully, after at most $\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6$ times queries, respectively, to H_1, H_2, H_3 hash, extract, DG, and blind signature oracles, then R-SIS$^{\kappa}_{q,1,2,\beta}$ can be solved by Θ with probability at least $\eta' = ((1 - 2^{-\omega(\log N)})\eta)^2$ in time $t' = t + \tau_1^{\tau_4+\tau_6}(\tau_1\xi_1 + \tau_6\xi_6 + \tau_4\xi_4) + \tau_2^{\tau_5}(\tau_2\xi_2 + \tau_5\xi_5)$.*

Proof. We suppose that Γ is able to generate an IDPBS signature successfully with η; we are able to construct Θ to calculate *R-SIS*'s solution. The interaction environment can be simulated as follows.

Setup. Θ selects $h \in \mathbf{R}_q^{\times}$ and H_1, H_2, and H_3 at random. Next Θ calculates and delivers $paras = \{h, H_1, H_2, \epsilon, q, s\}$ to Γ.

Queries on H_1 Oracle. To reply to H_1 oracle's query, Θ creates one null list L_1. Once Θ obtains one id_i, Θ will query H_1. If there is a t_i consistent with the query, Θ will return t_i. Otherwise, Θ will select a random t_i. At last, Θ will return t_i to Γ and save (id_i, t_i) to L_1.

Queries on H_2 Oracle. To reply to H_2's queries, Θ creates a list L_2; is initialized null. When Θ receives an $(\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, w_i)$, Θ will query H_2. If there is one corresponding value t_i, Θ will return $\mathbf{u}_i$. Else Θ will choose one $\mathbf{u}_i$ at random. Finally, Θ will return $\mathbf{u}_i$ to Γ and save $(\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, w_i, \mathbf{u}_i)$ to L_2.

KE Oracle Queries. When Θ queries a private key related to one id_i, Γ will recover the corresponding (id_i, t_i) from L_1. Next Γ will run $sk_{id_i} = (s_{i_1}, s_{i_2}) \leftarrow [(t, 0) - Gausssian(sk, \sigma, (t, 0))]$; Γ will return id_i, t_i, sk_{id_i} to Θ and save the tuple to L_3.

DG Oracle Queries. When Θ requests the delegation queries, Θ will verify whether id_i has been queried for H_1 or KE_ε oracle. If it has been queried, Θ will obtain (id_i, t_i, sk_{id_i}) from L_3. Else Θ will simulate KE_ε oracle and get a new private key. Next Θ will execute $\boldsymbol{\sigma}_{i_1} = \mathbf{y}_{i_1} + s_{\mathcal{O}_{i_1}}\mathbf{u}_i$ and $\boldsymbol{\sigma}_{i_2} = \mathbf{y}_{i_2} + s_{\mathcal{O}_{i_2}}\mathbf{u}_i$ to get a legal delegation signature $(w_i, \mathbf{u}_i, \boldsymbol{\sigma}_{i_1}, \boldsymbol{\sigma}_{i_2})$ and save $(\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, m_i, \boldsymbol{\sigma}_{i_1}, \boldsymbol{\sigma}_{i_2})$ to L_4.

$Setup: ST_\varepsilon(1^N) \to (paras, mk)$
$Key\ Extraction: t \leftarrow H_1(id_i), sk_{id_i} = (s_{i_1}, s_{i_2}) \leftarrow [(t,0) - Gausssian(\mathbf{msk}, \sigma, (t,0))]$
$Delegation\ Generation\ (original\ signer\mathcal{O})$:
$\mathbf{y}_1, \mathbf{y}_2 \in \mathrm{D}_{\mathbb{Z}^N,s}$
$\mathbf{u} = H_2(\mathbf{y}_1 + h\mathbf{y}_2, \omega)$
$\boldsymbol{\sigma}_1 = \mathbf{y}_1 + s_{\mathcal{O}_1}\mathbf{u}$
$\boldsymbol{\sigma}_2 = \mathbf{y}_2 + s_{\mathcal{O}_2}\mathbf{u}$

$\xrightarrow{\omega, \mathbf{u}, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2}$

proxy signer $\mathcal{P}$

Blind (user $\mathcal{U}$):
$\mathbf{y}_3, \mathbf{y}_4, \boldsymbol{\alpha}, \boldsymbol{\gamma} \in D_{\mathbb{Z}^N,s}$
$\mathbf{e} = H_3(\mathbf{y}_3 + h\mathbf{y}_4 + h\boldsymbol{\gamma} + \boldsymbol{\alpha} - \boldsymbol{\alpha}H(id), m)$
$\mathbf{e}^* = \mathbf{e} - \boldsymbol{\alpha}$

$\xrightarrow{\mathbf{e}^*, \mathbf{y}_3, \mathbf{y}_4}$

Proxy Signature (proxy signer $\mathcal{P}$):
$\boldsymbol{\sigma}_3^* = \mathbf{y}_3 + s_{\mathcal{P}_1}\mathbf{e}^*$
$\boldsymbol{\sigma}_4^* = \mathbf{y}_4 + s_{\mathcal{P}_2}\mathbf{e}^*$

$\xleftarrow{\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \mathbf{u}, \omega, \boldsymbol{\sigma}_3^*, \boldsymbol{\sigma}_4^*}$

Unblind (user $\mathcal{U}$):
$\boldsymbol{\sigma}_3 = \boldsymbol{\sigma}_3^* + \boldsymbol{\alpha}$
$\boldsymbol{\sigma}_4 = \boldsymbol{\sigma}_4^* + \boldsymbol{\gamma}$
$signature \quad tuple(\omega, m, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \mathbf{u}, \boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4, \mathbf{e})$
$Proxy\ Verify\ (verifier\ \mathcal{V})$:
if $\|(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)\| \le 2s\sqrt{2N}$
& $\|(\boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4)\| \le 2s\sqrt{2N}$
& $H_2(h\boldsymbol{\sigma}_2 + \boldsymbol{\sigma}_1 - H_1(id_{\mathcal{O}})\mathbf{u}, \omega) = \mathbf{u}$
& $H_3(h\boldsymbol{\sigma}_4 + \boldsymbol{\sigma}_3 - H_1(id_{\mathcal{P}})\mathbf{e}, m) = \mathbf{e}$
output true, otherwise false

Algorithm 4: IDPBS-NTRU protocol.

Signature Oracle Queries. Γ queries the signing oracle for $\mathbf{y}_{i_3}$, $\mathbf{y}_{i_4}$, α_i, β_i, id_i, and m_i. Θ will verify whether id_i has been queried for H_1 or KE_ε oracle. If it has been queried, Θ will obtain (id_i, t_i, sk_{id_i}) from L_3. Else Θ will simulate the EX_ϵ oracle and get a new private key. Next Θ will execute DG queries and then obtain $(\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, \omega_i, u_i, \boldsymbol{\sigma}_{i_1}, \boldsymbol{\sigma}_{i_2})$ from L_4. Then Θ will execute $\boldsymbol{\sigma}_{i_3} = \mathbf{y}_{i_3} + s_{\mathcal{P}_{i_1}}\mathbf{e}_i$ and $\boldsymbol{\sigma}_{i_4} = \mathbf{y}_{i_4} + s_{\mathcal{P}_{i_2}}\mathbf{e}_i$ to get a valid signature $(m_i, \mathbf{u}_i, \boldsymbol{\sigma}_{i_3}, \boldsymbol{\sigma}_{i_4}, \mathbf{e}_i)$ and save $(\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, \omega_i, u_i, \boldsymbol{\sigma}_{i_1}, \boldsymbol{\sigma}_{i_2}, \mathbf{y}_{i_3}, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \mathbf{y}_{i_4}, m_i, \boldsymbol{\sigma}_{i_3}, \boldsymbol{\sigma}_{i_4}, \mathbf{e}_i)$ to L_5.

Output. At last, Γ will output one forged signature $(\mathbf{u}_i, \omega_i, \boldsymbol{\sigma}_{i_1}, \boldsymbol{\sigma}_{i_2}, m_i, \mathbf{e}_i, \boldsymbol{\sigma}_{i_3}, \boldsymbol{\sigma}_{i_4})$ on w_i, m_i, $id_{\mathcal{O}_i}$, and $id_{\mathcal{P}_i}$ for the first time. Θ will rewind Γ to the point where w_i and m_i are queried for H_1; next Γ will get a valid tuple $(\mathbf{u}_i', \omega_i', \boldsymbol{\sigma}_{i_1}', \boldsymbol{\sigma}_{i_2}', m_i', \mathbf{e}_i', \boldsymbol{\sigma}_{i_3}', \boldsymbol{\sigma}_{i_4}')$ once again.

Thus, Θ are able to solve $R\text{-}SIS^{\kappa}_{q,1,2,\beta}$ problem. Θ will compute $\boldsymbol{\sigma}_{i_3^*} = \mathbf{y}_{i_3} + s_{O_{i_1}}\mathbf{e}_i$, $\boldsymbol{\sigma}_{i_4} = \mathbf{y}_{i_4} + s_{P_{i_2}}\mathbf{e}_i$, and $\boldsymbol{\sigma}_{i_3} + \boldsymbol{\sigma}_{i_4}h - H(id_{\mathcal{P}_i})\mathbf{e}_i$. Next Θ will verify whether $\boldsymbol{\sigma}_{i_3} + \boldsymbol{\sigma}_{i_4}h - H(id_{\mathcal{P}_i})\mathbf{e}_i = \boldsymbol{\sigma}_{i_3}' + \boldsymbol{\sigma}_{i_4}'h - H(id_{\mathcal{P}_i})\mathbf{e}_i = \mathbf{y}_{i_3} + h\mathbf{y}_{i_4} + h\gamma_i + \alpha_i - \alpha_i H(id_{P_i})$. If $(\boldsymbol{\sigma}_{i_3}, \boldsymbol{\sigma}_{i_4}) \neq (\boldsymbol{\sigma}_{i_3}', \boldsymbol{\sigma}_{i_4}')$, we can obtain $\|(\boldsymbol{\sigma}_{i_3} - \boldsymbol{\sigma}_{i_3}', \boldsymbol{\sigma}_{i_4} - \boldsymbol{\sigma}_{i_4}')\| \le 4s\sqrt{2N}$. After that, $(\boldsymbol{\sigma}_{i_4} - \boldsymbol{\sigma}_{i_4}', \boldsymbol{\sigma}_{i_3} - \boldsymbol{\sigma}_{i_3}')$ is a solution to $R\text{-}SIS^{\kappa}_{q,1,2,\beta}$. Similarly, we can obtain that $(\boldsymbol{\sigma}_{i_2} - \boldsymbol{\sigma}_{i_2}', \boldsymbol{\sigma}_{i_1} - \boldsymbol{\sigma}_{i_1}')$ is a solution to $R\text{-}SIS^{\kappa}_{q,1,2,\beta}$.

After that, we begin to analyze Θ's advantages. As mentioned above, Θ will win this game if Γ has already forged a valid $(\boldsymbol{\sigma}_{i_1}', \omega_i, m_i, \boldsymbol{\sigma}_{i_2}', \mathbf{u}_i', \boldsymbol{\sigma}_{i_3}', \boldsymbol{\sigma}_{i_4}', \mathbf{e}_i')$ and $(\boldsymbol{\sigma}_{i_1}, \boldsymbol{\sigma}_{i_2}) \neq (\boldsymbol{\sigma}_{i_1}', \boldsymbol{\sigma}_{i_2}')$ and $(\boldsymbol{\sigma}_{i_3}, z_{i_4}) \neq (\boldsymbol{\sigma}_{i_3}', \boldsymbol{\sigma}_{i_4}')$. The simulation of the EX_ϵ oracle fails if H_2 causes inconformity. Then Θ is able to solve $R\text{-}SIS^{\kappa}_{q,1,2,\beta}$ with probability at minimum $\eta' = ((1 - 2^{-\omega(\log N)})\eta)^2$ [25]; here, $\beta = 4s\sqrt{2N}$; it is clear that $t' = t + \tau_1^{\tau_4+\tau_6}(\tau_1\xi_1 + \tau_6\xi_6 + \tau_4\xi_4) + \tau_2^{\tau_5}(\tau_2\xi_2 + \tau_5\xi_5)$. Therefore, we can prove the theorem.

(4) Verifiability. Once receiving $(\omega, m, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \mathbf{u}, \boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4, \mathbf{e}, id_{\mathcal{O}}, id_{\mathcal{P}})$, V will execute PV_ε to check whether $\|(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)\| \le 2s\sqrt{2N}$ and $H_2(h\boldsymbol{\sigma}_2 + \boldsymbol{\sigma}_1 - H_1(id_{\mathcal{O}})\mathbf{u}, w) = \mathbf{u}$ are true. If both are true, the proxy signer is delegated by $\mathcal{O}$ to sign on m.

(5) Strong Identifiability. After receiving $(w, m, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, u, \boldsymbol{\sigma}_3, \boldsymbol{\sigma}_4, \mathbf{e}, id_{\mathcal{O}}, id_{\mathcal{P}})$, V can confirm $\mathcal{P}$'s identity in accordance with $id_{\mathcal{P}}$; thus the IDPBS-NTRU scheme satisfies strong identifiability.

(6) Strong Undeniability. $\boldsymbol{\sigma}_3$ and $\boldsymbol{\sigma}_4$ are signed by using $\mathcal{P}$'s secret keys $s_{\mathcal{P}_1}$ and $s_{\mathcal{P}_2}$; they will only be known by $\mathcal{P}$; thus $\mathcal{P}$ cannot refuse his signature once he signed; thus the IDPBS-NTRU scheme satisfies strong undeniability.

Table 1: Performance comparison with other lattice-based IDPBS schemes.

	ZM [33]	TA [34]	MMHP [35]	Ours
Problem	SIS	DLP	DLP	R-SIS
OSS	$m^2 \log(\lambda + 1)$	$3m$	m	$2N \log(12\sigma) + N(\log \lambda + 1)$
OSK	$m^2 \log(\lambda + 1)$	m	m	$2N \log(s\sqrt{N})$
PSS	$2m \log(\lambda + 1)$	m	m	$4N \log(12\sigma) + 2N(\log \lambda + 1)$
PSK	$m^2 \log(\lambda + 1)$	$7m$	m	$2N \log(s\sqrt{N})$

(7) Key Dependence. $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ on warrant ω are signed by $\mathcal{O}$'s secret keys $s_{\mathcal{O}_1}$ and $s_{\mathcal{O}_2}$; they are only known by $\mathcal{O}$,; $\mathcal{P}$ has no legal right to sign on a message before he is authorized by $\mathcal{O}$; thus the IDPBS-NTRU scheme satisfies key dependence. □

5.2. Performance. In this section, we compare the performance of IDPBS-NTRU with other IDPBS schemes. λ is written as security parameter, we denote $\mathcal{O}$'s signature size and signing-key size as OSS and OSK, respectively. $\mathcal{P}$'s signature size and signing key size are denoted as PSS and PSK, respectively. In ZM scheme [33], the parameters satisfy $m \geq 2N \lg q$ and $q > \beta\omega \log n$. In TA [34] and MMHP [35] schemes, the security parameter m is equal to N.

In Table 1, we compute the signature size and signing key size for $\mathcal{O}$ and $\mathcal{P}$. It is clear to draw a conclusion that our proxy signer's OSS, OSK, and PSK are smaller than ZM, TA, and MMHP schemes, our PSS is larger than ZM scheme, our PSS is smaller than TA scheme and MMHP scheme, and our OSS, OSK, PSK, and PSS are larger than TA and MMHP schemes. However, TA scheme and MMHP scheme are based on DLP; they are considered as not secure to resist the quantum computer attack. So our scheme can be more secure than them.

6. Conclusions

In this work, we present an IDPBS-NTRU scheme by using NTRU lattice; this scheme plays an important role in offline e-payment system, which can be deployed in fog computing model. We demonstrate that IDPBS-NTRU is efficient and secure. In addition, our IDPBS-NTRU's OSS, OSK, and PSK are smaller than ZM scheme and safer than TA and MMHP schemes. The proposed scheme is constructed based on NTRU lattice, which has the advantages of NTRU lattice. In the future, we will continue to construct a partial IDPBS scheme based on lattice.

Denotations

$\mathcal{O}$: Original signer
$\mathcal{P}$: Proxy signer
$\mathcal{U}$: A user
$\mathcal{V}$: A certifier
TTP: Trusted third party
$params$: System parameters
mk: Master key
mpk: Master public-key
msk: Master secret-key
id: User's identity
sk_{id}: Secret key related to a user
ω: Warrant
m: A message
m': A blinded message
f: A blind factor
ψ: Delegation
σ: Blind signature
$R = Z[x]/(x^N + 1)$: A ring
$f = \sum_{i=0}^{N-1} f_i x^i$: A polynomial in R
$g = \sum_{i=0}^{N-1} g_i x^i$: A polynomial in R
$\widetilde{\Omega}(\cdot)$: The asymptotic lower bound
$Poly(N)$: A polynomial function related to N
A: An adversary
C: A challenger
c: A constant
N: Security parameter.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (no. U1636213).

References

[1] M. Z. A. Bhuiyan, G. Wang, J. Wu, J. Cao, X. Liu, and T. Wang, "Dependable Structural Health Monitoring Using Wireless Sensor Networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 363–376, 2017.

[2] X. Liu, K. R. Choo, R. H. Deng, R. Lu, and J. Weng, "Efficient and Privacy-Preserving Outsourced Calculation of Rational Numbers," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 27–39, 2018.

[3] X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2401–2414, 2016.

[4] Y. Zhang and Z. Han, "Multi-dimensional Payment Plan in Fog Computing with Moral Hazard," in *Contract Theory for Wireless Networks*, Wireless Networks, pp. 73–88, Springer International Publishing, Cham, 2017.

[5] Y. Xue, Y.-a. Tan, C. Liang, Y. Li, J. Zheng, and Q. Zhang, "RootAgency: A Digital signature-based root privilege management agency for cloud terminal devices," *Information Sciences*, vol. 444, pp. 36–50, 2018.

[6] S. Park and Y. Yoo, "Network Intelligence Based on Network State Information for Connected Vehicles Utilizing Fog Computing," *Mobile Information Systems*, vol. 2017, Article ID 7479267, 9 pages, 2017.

[7] Y. Tan, Y. Xue, C. Liang et al., "A root privilege management scheme with revocable authorization for Android devices," *Journal of Network and Computer Applications*, vol. 107, pp. 69–82, 2018.

[8] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, "Significant Permission Identification for Machine Learning Based Android Malware Detection," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1-1, 2018.

[9] M. Z. Alam Bhuiyan, J. Wu, G. Wang, and J. Cao, "Sensing and Decision Making in Cyber-Physical Systems: The Case of Structural Event Monitoring," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2103–2114, 2016.

[10] X. Chen, X. Huang, J. Li, J. Ma, W. Lou, and D. S. Wong, "New algorithms for secure outsourcing of large-scale systems of linear equations," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 69–78, 2014.

[11] Z. Guan, J. Li, L. Zhu, Z. Zhang, X. Du, and M. Guizani, "Toward Delay-Tolerant Flexible Data Access Control for Smart Grid With Renewable Energy Resources," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3216–3225, 2017.

[12] R. Zhu, B. Zhang, J. Mao, Q. Zhang, and Y.-A. Tan, "A methodology for determining the image base of ARM-based industrial control system firmware," *International Journal of Critical Infrastructure Protection*, vol. 16, pp. 26–35, 2017.

[13] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private Naive Bayes learning over multiple data sources," *Information Sciences*, vol. 444, pp. 89–104, 2018.

[14] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, and C. Gao, "Dynamic Fully Homomorphic encryption-based Merkle Tree for lightweight streaming authenticated data structures," *Journal of Network and Computer Applications*, vol. 107, pp. 113–124, 2018.

[15] X. Yu, Y. Tan, C. Zhang et al., "A High-Performance Hierarchical Snapshot Scheme for Hybrid Storage Systems," *Journal of Electronics*, vol. 27, no. 1, pp. 76–85, 2018.

[16] Q. Lin, J. Li, Z. Huang, W. Chen, and J. Shen, "A short linearly homomorphic proxy signature scheme," *IEEE Access*, vol. 6, pp. 12966–12972, 2018.

[17] Z. Guan, J. Li, L. Wu, Y. Zhang, J. Wu, and X. Du, "Achieving Efficient and Secure Data Acquisition for Cloud-Supported Internet of Things in Smart Grid," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1934–1944, 2017.

[18] Y. Xue, Y.-A. Tan, C. Liang, C. Zhang, and J. Zheng, "An optimized data hiding scheme for Deflate codes," *Soft Computing*, pp. 1–11, 2017.

[19] Z. Sun, Q. Zhang, Y. Li, and Y. Tan, "DPPDL: a Dynamic Partial-Parallel Data Layout for Green Video Surveillance Storage," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 1, pp. 193–205, 2018.

[20] H. Zhu, Y.-A. Tan, X. Zhang, L. Zhu, C. Zhang, and J. Zheng, "A round-optimal lattice-based blind signature scheme for cloud services," *Future Generation Computer Systems*, vol. 73, pp. 106–114, 2017.

[21] X. Zhang, Y. Tan, C. Liang, Y. Li, and J. Li, "A Covert Channel Over VoLTE via Adjusting Silence Periods," *IEEE Access*, vol. 6, pp. 9292–9302, 2018.

[22] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, vol. PP, no. 99, pp. 1-1, 2018.

[23] Y. Li, G. Wang, L. Nie, Q. Wang, and W. Tan, "Distance metric optimization driven convolutional neural network for age invariant face recognition," *Pattern Recognition*, vol. 75, pp. 51–62, 2018.

[24] Z. Liu, Z. Wu, T. Li, J. Li, and C. Shen, "GMM and CNN Hybrid Method for Short Utterance Speaker Recognition," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1-1, 2018.

[25] J. Xie, Y.-P. Hu, J.-T. Gao, and W. Gao, "Efficient identity-based signature over NTRU lattice," *Frontiers of Information Technology and Electronic Engineering*, vol. 17, no. 2, pp. 135–142, 2016.

[26] S. Beura, M. Behera, and A. K. Tripathy, "Secured proxy blind signature scheme based on dlp with minimum computation cost," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 2, pp. 808–811, 2011.

[27] H. M. Sun, B. T. Hsieh, and S. M. Tseng, "On the security of some proxy blind signature schemes," *The Journal of Systems and Software*, vol. 74, no. 3, pp. 297–302, 2005.

[28] C.-H. Wang and M.-Z. Liao, "Security Analysis and Enhanced Construction on ECDLP-Based Proxy Blind Signature Scheme," *International Journal of e-Education, e-Business, e-Management and e-Learning*, vol. 4, no. 1, pp. 47–51, 2014.

[29] J. Zheng, Y.-a. Tan, Q. Zhang, X. Zhang, L. Zhu, and Q. Zhang, "Cross-cluster asymmetric group key agreement for wireless sensor networks," *Science China Information Sciences*, vol. 61, no. 4, pp. 048103:1–048103:3, 2018.

[30] Z. Tan, "Efficient pairing-free provably secure identity-based proxy blind signature scheme," *Security and Communication Networks*, vol. 6, no. 5, pp. 593–601, 2013.

[31] F. Zhang, R. Safavi-Naini, and C.-Y. Lin, "New proxy signature, proxy blind signature and proxy ring signature schemes from bilinear pairing," in *Cryptology ePrint Archive, Report 2003/104*, 2003, https://eprint.iacr.org/2003/104.

[32] P. Li, J. Li, Z. Huang et al., "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.

[33] L. Zhang and Y. Ma, "A lattice-based identity-based proxy blind signature scheme in the standard model," *Mathematical Problems in Engineering*, vol. 2014, Article ID 307637, 6 pages, 2014.

[34] N. Tahat and E. E. Abdallah, "A proxy partially blind signature approach using elliptic curve cryptosystem," *International Journal of Mathematics in Operational Research*, vol. 8, no. 1, pp. 87–95, 2016.

[35] S. Mohapatra, S. Mohanty, A. Hota, and S. Pattanayak, "Data Security Enhancement in Cloud Computing using Proxy Blind Signature," *International Journal of Computer Applications*, vol. 161, no. 2, pp. 27–31, 2017.

[36] K. Gu, W. Jia, Y. Deng, and X. Nie, "Secure and efficient multi-proxy signature scheme in the standard model," *Journal of Electronics*, vol. 25, no. 1, pp. 93–99, 2016.

[37] X. Yang, C. Wang, L. Zhang, and J. Qiu, "On-line/off-line threshold proxy re-signatures," *Chinese Journal of Electronics*, vol. 23, no. 2, pp. 248–253, 2014.

[38] X. Zhang, Y.-A. Tan, Y. Xue et al., "Cryptographic key protection against FROST for mobile devices," *Cluster Computing*, vol. 20, no. 3, pp. 2393–2402, 2017.

[39] C. Gao, Q. Cheng, X. Li, and S. Xia, "Cloud-assisted privacy-preserving profile-matching scheme under multiple keys in mobile social network," *Cluster Computing*, pp. 1–9, 2018.

[40] T. Peng, Q. Liu, D. Meng, and G. Wang, "Collaborative trajectory privacy preserving scheme in location-based services," *Information Sciences*, vol. 387, pp. 165–179, 2017.

[41] Z. Chen, L. Peng, C. Gao, B. Yang, Y. Chen, and J. Li, "Flexible neural trees based early stage identification for IP traffic," *Soft Computing*, vol. 21, no. 8, pp. 2035–2046, 2017.

[42] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *Journal of Network and Computer Applications*, vol. 106, pp. 117–123, 2018.

[43] X. J. Du, M. Guizani, Y. Xiao, and H.-H. Chen, "Transactions papers a routing-driven Elliptic Curve Cryptography based key management scheme for Heterogeneous Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1223–1229, 2009.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 4627108, 19 pages
https://doi.org/10.1155/2018/4627108

Research Article

Niffler: A Context-Aware and User-Independent Side-Channel Attack System for Password Inference

Benxiao Tang, Zhibo Wang, Run Wang, Lei Zhao, and Lina Wang

School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Zhibo Wang; wzb.zju@gmail.com

Received 2 March 2018; Accepted 26 March 2018; Published 8 May 2018

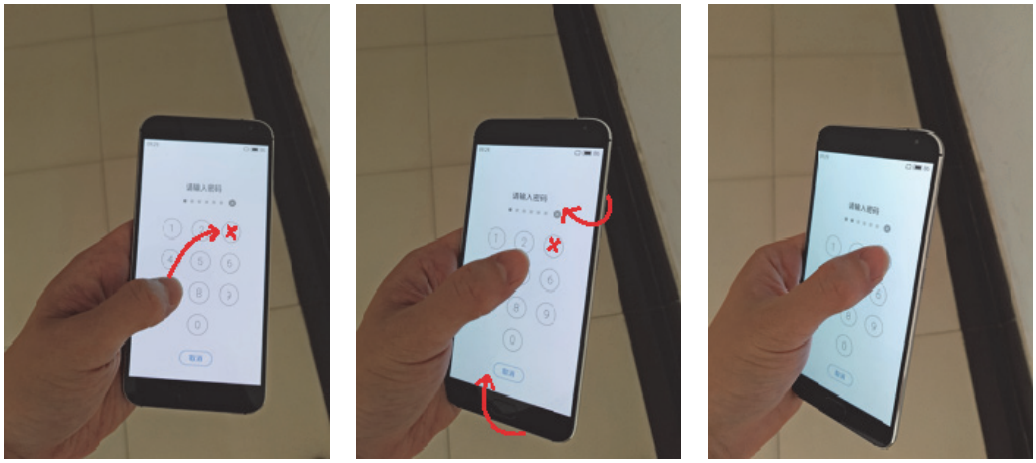
Academic Editor: Ximeng Liu

Digital password lock has been commonly used on mobile devices as the primary authentication method. Researches have demonstrated that sensors embedded on mobile devices can be employed to infer the password. However, existing works focus on either each single keystroke inference or entire password sequence inference, which are user-dependent and require huge efforts to collect the ground truth training data. In this paper, we design a novel side-channel attack system, called Niffler, which leverages the user-independent features of movements of tapping consecutive buttons to infer unlocking passwords on smartphones. We extract angle features to reflect the changing trends and build a multicategory classifier combining the dynamic time warping algorithm to infer the probability of each movement. We further use the Markov model to model the unlocking process and use the sequences with the highest probabilities as the attack candidates. Moreover, the sensor readings of successful attacks will be further fed back to continually improve the accuracy of the classifier. In our experiments, 100,000 samples collected from 25 participants are used to evaluate the performance of Niffler. The results show that Niffler achieves 70% and 85% accuracy with 10 attempts in user-independent and user-dependent environments with few training samples, respectively.

1. Introduction

With the rapid development of embedded systems and mobile computing, mobile devices (e.g., smartphones) become more and more powerful and make our life much more convenient than before. Thousands of Apps have been developed for mobile devices which rely on the embedded sensors (e.g., camera, accelerometer, and compass) to provide specific functions. Unlike certain data, such as photo, which is protected strictly and can be stored in cloud [1, 2], mobile systems provide kinds of interfaces for developers to access sensors with little restrictions. Although providing convenience [3] and protection [4] for developers and mobile users, this leaves vulnerability to attackers who intend to construct side-channel attacks with embedded sensors to recover users' sensitive information, such as the location [5], screen lock password, and credit card numbers [6–8]. These attacks based on motion sensors usually work in background and the perceived risk of motion sensors among users is very low [9]. Considering the popularity and universality of mobile devices, this kind of attacks has become a significant threat to sensitive information leakage.

Although there exist other secure authentication methods (e.g., fingerprint reader, graphical password), digital password still remains a vital place and is commonly used in screen lock and mobile payment. Many works have leveraged motion sensors to launch side-channel attacks to infer password lock. Some works divide the unlocking process into several tap events on keystrokes and explore the process of tapping a single keystroke to infer the lock password [10–13]. They found that tapping different positions on touchscreen causes particular reflections on the readings of accelerometer and gyroscope. Based on this observation, they inferred every single keystroke separately and combined them together to recover users' passwords. In contrast, Aviv et al. refer the process of tapping one entire password as a category to infer the password, instead of individual keystroke [14]. However, it is difficult to build a ground truth training set that contains all possible categories (e.g., there are 10^4 categories for four-digit password). Researches have proved that users have their



(a) (b) (c)

Figure 1: An illustration of grip change for a movement from button "7" to button "3." (a) The user taps button "7" and wants to tap button "3"; (b) the device is tilted to a suitable angle for the user; (c) the user taps button "3" with his thumb.

own unique behavioral patterns when tapping keystrokes on the touch screens [15], and the authentication mechanism [16, 17] exploring motion sensor is based on this theory; therefore, all of the above side channels are sensitive to users. The common assumption that attackers can collect sufficient data in target's device in training stage is difficult in practice. Whether accessing to mobile sensors within app or browser [18], attackers must build a unique model for certain target.

Note that existing attack methods focus on either single keystroke inference or whole password inference, which are user-dependent so that they are not scalable in practice and require a huge effort of collecting the ground truth training set [19]. In this paper, we explore the user-independent feature during the password unlocking process to accurately infer password with little effort and good scalability. We consider a common scenario that a user holds and unlocks its smartphones with one hand (according to the research of Hoober, 49% users hold mobile devices with one hand as they are the right size for that [20]). If a user moves his thumb from one keystroke to another, he needs to change the grip of the smartphone. We observe a fact from the experiments that same movements from one keystroke to another of different users have similar grip changes which can be reflected by similar changing trends of motion sensor readings. As shown in Figure 1, when a user moves his thumb from button "7" to button "3", the device will be tilted to a suitable angle for the thumb. Drawn from the observation and the characteristics of password unlocking process which consists of alternative movement and tap event, we argue that password unlocking is a Markov process and the movements of tapping two consecutive keystrokes are context-aware and user-independent, and the experimental results shown in Section 3 validate our argument.

In this paper, we design a novel side-channel attack system, called Niffler, that leverages accelerometer to infer unlocking passwords on smartphones by exploiting user-independent movements of tapping consecutive buttons. The basic idea of Niffler is to infer every movement (e.g., movement "7-3" in Figure 1) of the password sequence with accelerometer readings and then combine them together to find the most possible candidates for the password. We extract angle features to reflect the changing trends of similar grip changes. With these features, we build a multicategory classifier to infer the probability of each movement of password sequence and employ the Markov model to reconstruct the whole password sequence. The sequences with the highest probabilities will be considered as the possible candidates for the password. In particular, Niffler provides a feedback component that can improve the accuracy of classifier by feeding the successful attack results back to the classifier.

The data of our experiments are collected from twenty-five volunteers over two months. We evaluate the performance of Niffler in both user-dependent and user-independent environments. In user-dependent experiments, with a few training samples, Niffler achieves an average accuracy about 85% for single movement inference with one attempt. The accuracy of the password inference is near 40% with only one attempt and reaches 84% with 10 attempts. In user-independent environments, the accuracy of the password inference is about 75% with 10 attempts.

Our contributions are summarized as follows:

(i) We observe a fact from experiments that movements during password unlocking are context-aware and user-independent. The same movement of different users has similar grip changes which lead to similar changing trends of accelerometer readings.

(ii) We design and implement a side-channel attack system to infer digital password on smartphones. Different from existing works, Niffler infers every movement of the password based on the user-independent feature and employs the Markov model to reconstruct the whole password sequence.

(iii) We build a multicategory classifier that divides all the centroids of the 110 movements into 11 categories to avoid the ambiguity due to similar movements and

adopt the dynamic time warping (DTW) algorithm to infer the probability of each movement, where each category contains 10 centroids of movements with the same starting button.

(iv) We conduct extensive experiments to evaluate the performance of Niffler in both user-dependent and user-independent environments. The experimental results demonstrate that Niffler is an effectively user-independent attack system.

The rest of this paper is organized as follows. We describe the system model in Section 2 and present the observation of internal relation of movements in Section 3. We present the overview and the design of the attack system in Section 4. The experimental results are shown in Section 5. The literatures on password unlock are briefly discussed in Section 6. Finally, Section 7 concludes the paper.

2. System Model

2.1. Digital Lock. Digital lock and pattern lock are still the most popular authentication methods for touch screen devices such as smartphones and tablets even if some new secure authentication methods have arisen. A typical layout of screen lock is shown in Figure 2. A digital password for screen lock is usually 4 or 6 in length where each number can be randomly selected from "0" to "9." Unlike the rules of pattern lock, each number in a digital password can appear more than once, so there will be 10^4 possibilities for a 4-digit password. Given that the Android operating system usually allows at most 20 attempts of entering wrong passwords before locking the device, it is impossible to unlock the device by random guess. In this paper, we focus on the inference of 4-digit password on smartphone, but the attack model can be easily extended to infer 6-digit password.

Generally speaking, Android allows 5 consecutive attempts within a specified time window by default when a user attempts to log in with his/her password. If the user fails to pass the authentication within 5 attempts, he/she is not allowed to try again until a certain time period passes. The interval is usually one minute, and after that another 5 attempts will be given. Most manufacturers allow 20 attempts in total, and the others allow at least 10 attempts.

2.2. Sensor Monitoring. In Android, developers can easily access to sensors in application layer by calling system APIs without claiming any permissions [21]. Furthermore, the sensor data is regarded as public source which is not protected by sandbox. Although users can outsource these data to a cloud server to preserve privacy [22–24], there are many ways for attackers to monitor sensor readings legally. For example, if a user permits the browser to access some harmless information, such as geographic location and sensor readings, the website can record these data stealthily.

Using Java, we designed a simple application which keeps running at background in Android application layer as a monitoring module to record accelerometer readings during user unlocking process. The program is hidden in a pedometer application which needs sensor data to avoid security inspection. Since the W3C specifications allow access to the motion sensors from JavaScript [9], more complex monitoring can be implemented via websites. Otherwise, researches have found approach to sniff and manipulate protected sensors on unrooted Android devices [25].

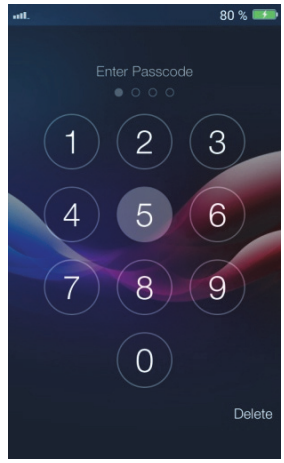


Figure 2: A typical layout of screen lock.

2.3. Scenarios and Assumptions. The main stream smartphones are designed with screen size about 4.7 inches to 5.5 inches which are suitable for single hand holding. We focus on the scenario that users hold mobile devices with one hand only because they are the right size for that, and a user can use the other hand to do other things simultaneously when he/she plays on the smartphone with one hand, according to the research of Hoober [20]. Niffler works for both left hand holding and right-hand holding, but in this paper, we use the scenario that users hold and unlock smartphones with their left-hands only as an example. The readings of accelerometers on smartphones are used to capture grip changes and infer the 4-digit password.

Definition 1 (movement). A movement is defined as the process of moving from one button to another on the screen during password unlocking.

If two movements start from the same keystroke and end at the same keystroke, they are called the *same movement*.

The stream of accelerometer readings can be represented as $\{a_1, a_2, \ldots, a_n\}$, where $a_i = (a_i^x, a_i^y, a_i^z)$ is the reading of accelerometer when the i_{th} sensor event occurs. a_i^x, a_i^y, and a_i^z express the acceleration in the x-axis, y-axis, and z-axis, respectively. The mean interval between every two tap events is about 500 ms and the typical tapping duration is between 100 ms and 200 ms [26], so we sample the accelerometer readings with a frequency of 50 Hz.

3. Context-Aware and User-Independent Movement

In this section, we use experimental results to show that the movements during unlocking process are context-aware and

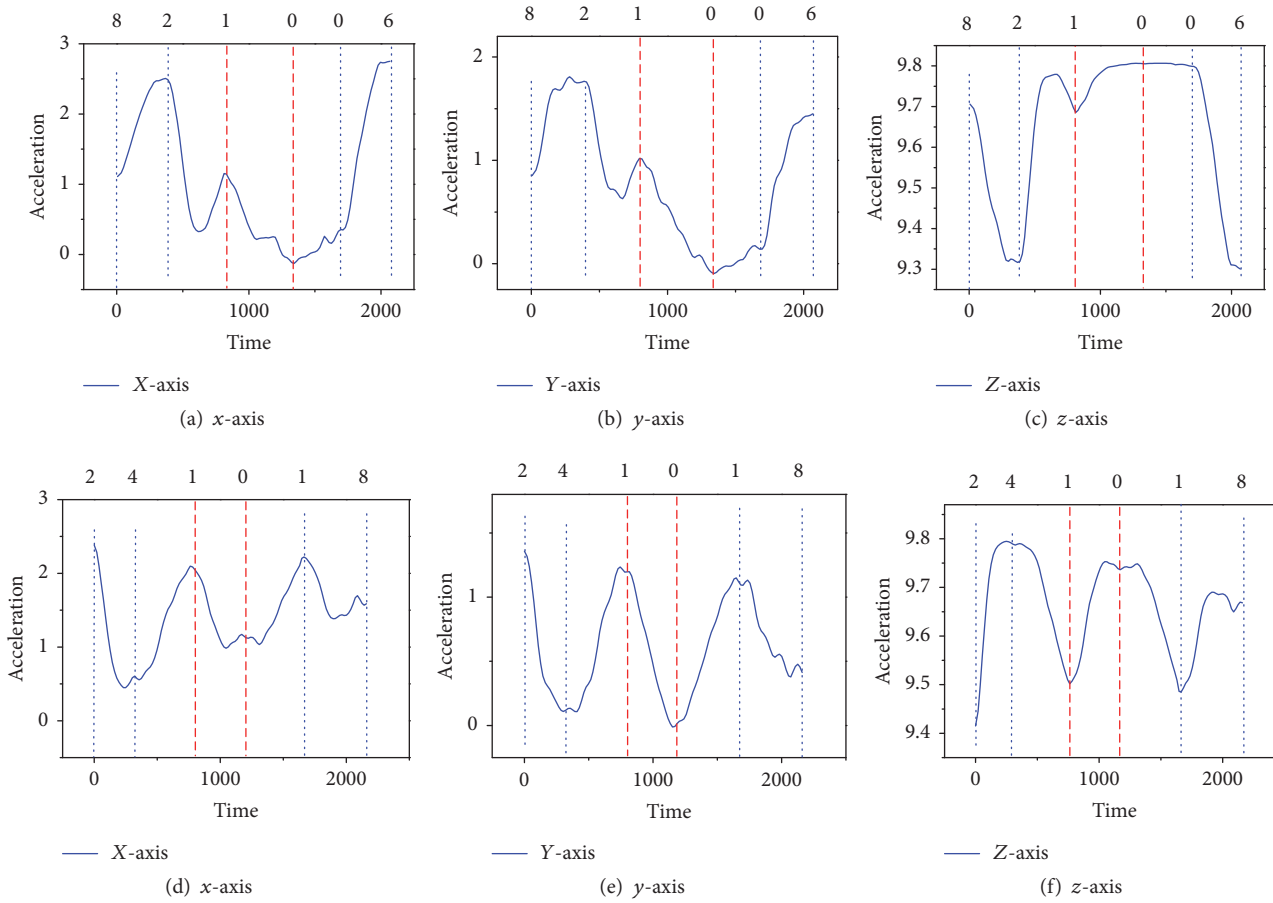


Figure 3: (a–c) Accelerometer readings in three axes of sequence "8-2-1-0-0-6" of user *a*. (d–f) Accelerometer readings in three axes of sequence "2-4-1-0-1-8" of user *b*.

user-independent, which is an ideal observation that can be used to recover passwords. The previous studies of sensor-based side-channel attacks [27] mainly focus on one user's unique features (e.g., pressure, touched size, key hold time, and interkey time) and use these user-dependent features to crack passwords [14] and help authentication [15] of specific target. However, the context-aware and user-independent features of movements can be employed to build a general side-channel attack model that is scalable to users.

Although smartphones with screen size of 4.7 inches to 5.5 inches are suitable for single hand holding, users still need to tilt and rotate smartphones when they are tapping keystrokes [28]. For example, as shown in Figure 1, a user holds his smartphone with his left hand and wants to type a two-keystroke sequence from button "7" to button "3" with his thumb. The right side of this smartphone has been lied down to uplift the bottom left corner when he types button "7." When he moves his thumb from button "7" to button "3" at the top right corner, the other four fingers uplift the right side of the smartphone to make the thumb reach the target position easily. We can see that the grip changes are context-aware that are related to the starting position and ending position of the movement. Due to similar structures of human hands as well as the layouts of virtual keyboards, we argue that the same movement on smartphones of different users leads to similar grip changes, which can be reflected by similar changing trends on accelerometer readings.

We perform extensive experiments and the results validate the correctness of our argument. In the following, we use two examples to better explain the context-awareness and user-independence of movements. The sensor sample frequency is 50 Hz, and we leave out tap events to put more focus on movements.

Figures 3(a)–3(c) show the accelerometer readings in three axes of a sequence "8-2-1-0-0-6" of user *a*, and Figures 3(d)–3(f) show the accelerometer readings in three axes of a sequence "2-4-1-0-1-8" of user *b*. These two users have one same movement "1-0". Although the data are from two different users, we can see that the readings of the same movement "1-0" have the same changing trends in each axis. In contrast, different movements, such as movement "2-1" of user *a* and movement "4-1" of user *b*, show quite different changing trends. These figures also imply the relation between the acceleration variation and the movements. For example, if the movement is from the bottom to the top (e.g., "8-2"), the readings in x-axis and y-axis have increasing trends, while the readings in the z-axis are decreasing.

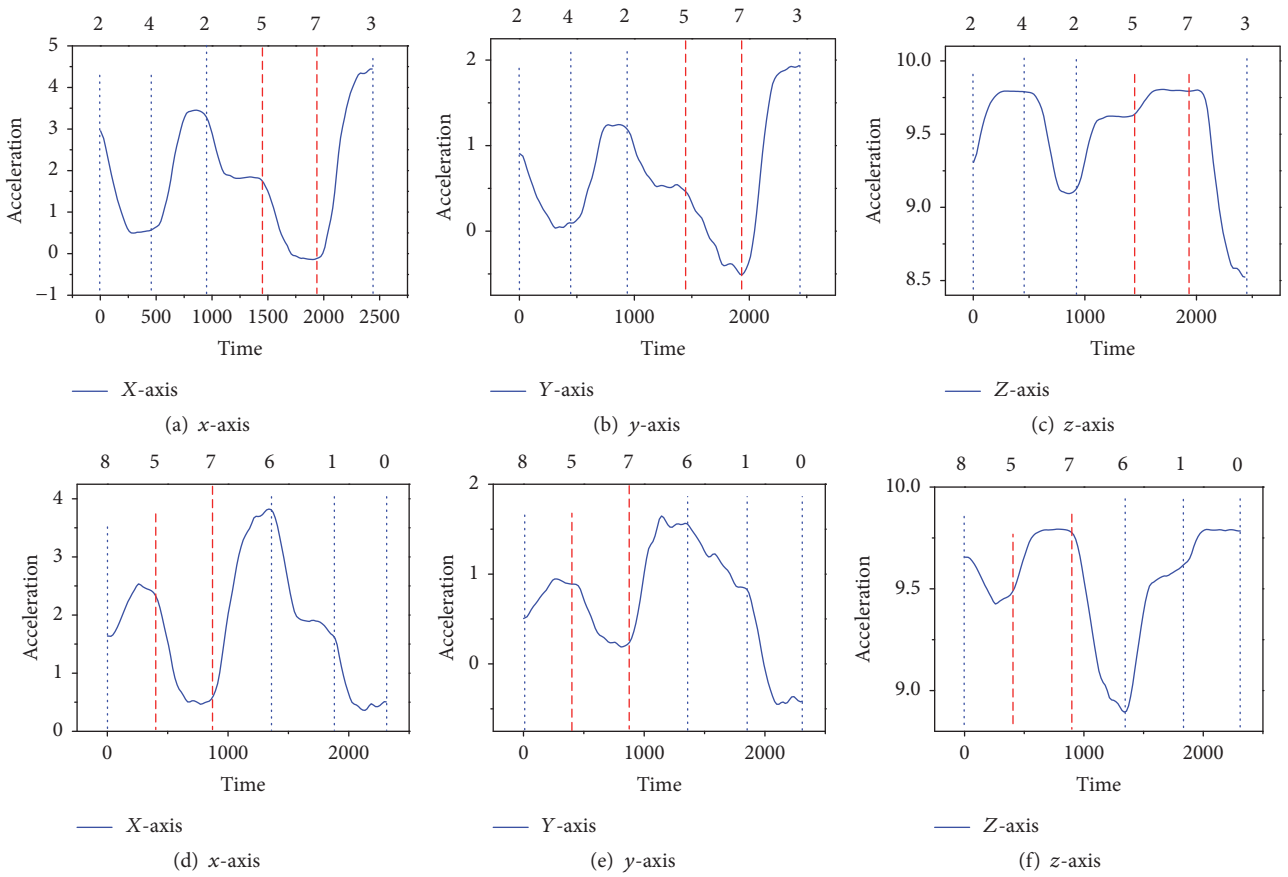


Figure 4: (a–c) Accelerometer readings in three axes of sequence "2-4-2-5-7-3" of user *c*. (d–f) Accelerometer readings in three axes of sequence "8-5-7-6-1-0" of user *d*.

Figure 4 is another example that Figures 4(a)–4(c) show the accelerometer readings in three axes of a sequence "2-4-2-5-7-3" of user *a*, and Figures 4(d)–4(f) show the accelerometer readings in three axes of a sequence "8-5-7-6-1-0" of user *b*. We can see that the same movement "5-7" of two different users also have similar changing trends in each axis.

Figures 3 and 4 validate our argument that the same movement leads to similar grip changes and therefore similar changing trends on accelerometer readings. This implies that the changing trends of accelerometer readings due to movements are user-independent, which in turn can be used to infer movements of a password sequence.

It is worth noting that, besides the same movement, similar movements may also lead to similar changing trends on accelerometer readings. For example, the movement of "8-2" and the movement of "4-1" are similar movements because both of them move upward straightly, which leads to quite similar grip changes and therefore similar changing trends on accelerometer readings in each axis, as shown in Figure 3. This introduces challenges to us when we infer movements from changing trends of accelerometer readings. In the Niffler, we design a multicategory classifier to avoid the ambiguity due to similar movements by putting the movements with the same starting button into one category.

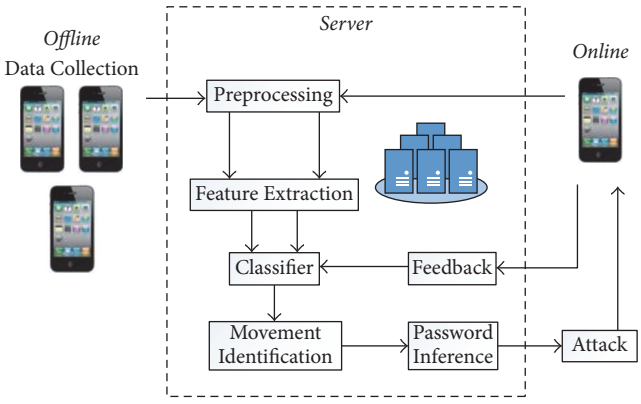


Figure 5: System architecture of Niffler.

4. System Design

In this section, we first give a high-level overview of Niffler, which uses accelerometer readings to infer unlocking passwords, and then describe the design of important components in Niffler.

Figure 5 shows the system architecture of Niffler. All the data collected on smartphones during the unlocking

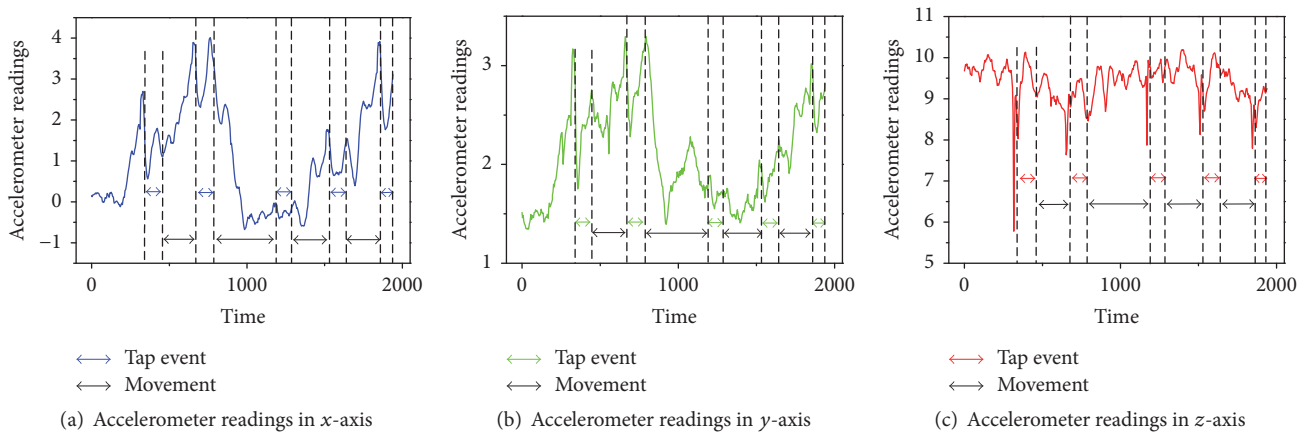


Figure 6: Accelerometer readings of password unlocking process.

process will be uploaded to the server and processed by Niffler. Niffler first preprocesses the collected data to remove noise and segment movements. Since the same movement leads to similar grip changes, angle feature vectors are extracted to represent grip changes of movements. During the offline phase, data of different movements from users are collected to train a classifier which can be further used to infer the movement during online phase. The dynamic time warping (DTW) algorithm is combined with the classifier to infer the possibility and probability of each movement of a password sequence. We further use the Markov model to model the unlocking process with multiple movements and use the sequences with the highest probabilities as the attack candidates. In particular, Niffler provides a feedback component that can improve the accuracy of classifier by feeding the successful attack results back to the classifier.

4.1. Preprocessing. All the raw data collected on smartphones during the unlocking process must be preprocessed at first, which aims to remove noise and segment original data into movements.

Noise Removal. The hand trembles during user tapping introduce noise to accelerometer readings which can affect the password inference. Since the frequency of hand trembles is much higher than that of grip changes, we use a median filter to remove noise. The sampling rate in Niffler is about Hz, so the window size of the median filter is 50 ms and moves with a step size of 20 ms.

Data Extension. The difference among devices' hardware leads to inconsistent sampling rate, even if using the sample app. In order to keep the consistency of frequencies of accelerometer readings, we use cubic spline interpolation to interpolate the vacant values. Data augmentation [29] is also used to leverage limited data in user-independent environment by transforming existing samples of same category to create new ones.

Movement Segmentation. We segment the accelerometer readings into movements in order to further extract features from movements. For the data collected offline, since we know the ground truth of password unlocking, we can easily segment the readings of each password into several movements. Note that, besides the movement between two keystrokes, we also consider a special movement that happens before tapping the first keystroke. Our experiments show that no matter where movements start from in the air, the grip changes are similar if the landing positions are the same. This observation is used for the case that the thumb hangs in the air. Because the average duration between two keystrokes is about 500 ms, we choose the time 300 ms before tapping the first keystroke as the time this movement starts.

Compared to the offline collected data, the segmentation is more difficult for online data, since it is impossible to record the ground truth of tap events legally by using a third-party application due to sandbox. We first identify the start of unlocking process by monitoring the power event. Based on our investigation, users should press the power button to turn on touch screen before they unlock mobiles for most of Android systems. We also use the sum of square root of acceleration in three dimensions to distinguish unlocking process from other activities (i.e., walking or running) [11]. The great background noise will interfere segmentation; thus, Niffler collects data with small standard deviation of acceleration readings.

Since a complete input process consists of tap event and movement in turn, we segment movements by identifying tap event, and the readings between two tapping events will be recognized as a movement. The methods of tap event detection in researches [11, 13] work poorly in our experiment; thus, we employ a new observation to measure the occurrence of tap events. We observe that accelerometer readings have a unique trend in the time of tapping a keystroke, which can be used to estimate the approximate time of tapping a keystroke with a high accuracy. The observation is shown in Figures 6 and 7.

Figure 6 presents accelerometer readings of a complete unlocking process with a higher frequency of 200 Hz to show the detail of tap events and movements. Although the acceleration fluctuations caused by tap events in x-axis (Figure 6(a)) and y-axis (Figure 6(b)) can hardly be identified

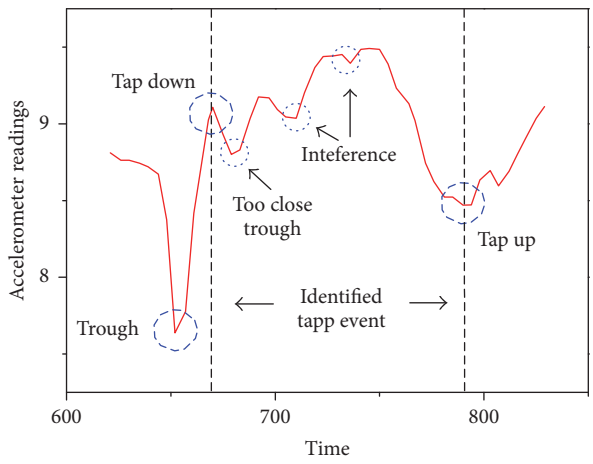


FIGURE 7: Accelerometer readings of a certain tap event in z-axis.

from movements, the tap events cause a unique and obvious change in accelerometer readings in z-axis (Figure 6(c)). At the moment someone taps the screen, he gives a downward (in z-axis) force to the device and this force will cause an instantaneous acceleration which includes gravity. Figure 7 shows accelerometer readings of one of tap events from Figure 6(c). There is a great trough before tap down, and the first peak point following this trough is the start of tap event. When the tap is finished and the user lifts his finger, an opposite direction acceleration is generated. This action causes another obvious trough which is the sign of the end of tap event. Note that if the following trough is too close (less than 15 ms) to the first trough, it will be regarded as noise. In addition, background noise and shaking of hand also cause some small waves within the tap event, and we lower these interferences by noise removal. The readings from the end of one tap event to the start of next tap event are segmented as a movement.

4.2. Feature Extraction. As we mentioned, the same movement of different users leads to similar grip changes and thus similar changing trends on accelerometer readings. With this observation, we use angle features from acceleration to form a feature vector to represent the trends of grip changes. The pitch angle and the roll angle represent the tilt degree of a device in forward-backward direction and the right-left direction, respectively. We ignore the azimuth angle because the experimental results show that there is little azimuth change during password unlocking process. Thus, we use pitch angles and roll angles as the features to represent grip changes.

The pitch angle and roll angle can be calculated through the gravity acceleration because it is a great reference system to calculate the angle of a device. As Figure 8 shows, the approximate pitch angle is expressed by arc-tangent between the components in z-axis and y-axis, and the approximate roll angle is computed by arc-tangent between the components in z-axis and x-axis.

Each accelerometer reading a_i collected from the device is a proper acceleration, which is a combination of gravity acceleration a_{gi} and linear acceleration a_{li}, where $a_{li} = (a_{li}^x, a_{li}^y, a_{li}^z)$ and $a_{gi} = (a_{gi}^x, a_{gi}^y, a_{gi}^z)$. We have to remove the linear acceleration in order to calculate the pitch angle and roll angle from the gravity acceleration. The linear acceleration in a fixed orientation is a transient variation and is sensitive with the force acted on the device, which means the linear acceleration changes with higher frequency than gravity acceleration. Therefore, we can use a discrete-time low-pass filter to remove the linear acceleration, which is shown as follows.

$$a_{gi} = \rho \times \left[a_{g(i-1)} + \left(a_i - a_{i-1}\right)\right], \tag{1}$$

where $\rho \triangleq c/(c + 1/f)$ controls the frequency of passing acceleration. c is a time constant and f is the data sampling rate. ρ ranges from 0 to 1, and the closer to 1, the higher frequency of passing acceleration. In our experiment, the value of ρ is 0.8.

With the gravity acceleration a_{gi}, the pitch angle and roll angle of a_i are $\arctan(a_{gi}^z, a_{gi}^y)$ and $\arctan(a_{gi}^z, a_{gi}^x)$, respectively. a_{gi}^x, a_{gi}^y, and a_{gi}^z are the components of a_{gi} in x-axis, y-axis, and z-axis, respectively.

Let $\mathbf{F} = [\mathbf{F}_p; \mathbf{F}_r]$ denote the feature vector of a movement, where $\mathbf{F}_p$ and $\mathbf{F}_r$ denote the pitch vector and the roll vector of the movement, respectively. The sampling rate for accelerometer readings is 50 Hz. Each movement may take different duration from another, so they have different number of readings. Thus, the sizes of feature vectors of different movements may be different. Suppose a movement has τ accelerometer readings. Its pitch vector and roll vector are represented as follows.

$$\begin{aligned}\mathbf{F}_p &= \left\{\arctan\left(a_{g1}^z, a_{g1}^y\right), \ldots, \arctan\left(a_{g\tau}^z, a_{g\tau}^y\right)\right\},\\ \mathbf{F}_r &= \left\{\arctan\left(a_{g1}^z, a_{g1}^x\right), \ldots, \arctan\left(a_{g\tau}^z, a_{g\tau}^x\right)\right\}.\end{aligned} \tag{2}$$

We further normalize the values of these two vectors to the range of $[0, 255]$ to eliminate the impact of different value ranges while keeping the changing trends of movements.

4.3. Multicategory Classifier Construction. In this section, we describe how to build a multicategory classifier with the collected samples in the offline phase. Note that the classifier can be further improved with a real-time feedback mechanism in the online phase by providing samples of successful attacks to the classifier.

For each button on the keyboard, there are 10 possible movements starting from it. In our system, we also consider the process starting from the air to the first tapped position as a movement. Thus, there are a total of 110 movements. Some ground truth samples of movements are collected and the feature vector for each sample of a movement can be extracted. The sizes of feature vectors are different from each other since movements take different durations. We scale the pitch vector and the roll vector of each feature vector into size of 30 considering that the durations of movements are less than 600 ms. The scaling operation also removes user's individual characteristics in time intervals and keeps

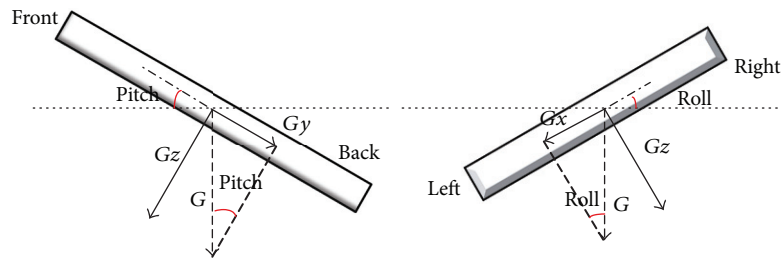


Figure 8: Angle calculation.

the general features between different users. Some researches have shown that the unique behavior of a user will be reflected in time intervals between two nearby keystrokes [15, 30]. Note that the feature vector of a movement is scaled only if it is used for building the classifier.

Let $C(\mathbf{F})$ denote the centroid of feature vectors of a movement, which consists of two vectors $C(\mathbf{F}_p)$ and $C(\mathbf{F}_r)$, where $C(\mathbf{F}_p)$ denotes the centroid of pitch vectors of the movement and $C(\mathbf{F}_r)$ denotes the centroid of roll vectors of the movement. Suppose there are a total of N samples for this movement; the centroid can be calculated as follows.

$$C\left(\mathbf{F}_p\right) = \frac{\sum \mathbf{F}_p}{N}, \qquad C\left(\mathbf{F}_r\right) = \frac{\sum \mathbf{F}_r}{N}. \tag{3}$$

After calculating the centroid for each movement, 110 movements have 110 corresponding centroids in the classifier. Once a new unknown movement comes, the most straightforward way to infer it is to calculate the similarity between its feature vector and each centroid in the classifier and the most similar one will be identified as the true movement. However, this straightforward way is not effective or accurate since there exist many similar movements that results in similar grip changes (e.g., "8-2" and "4-1"). It is difficult to accurately decide the true movement just by calculating the similarity to the centroid of each movement.

In order to reduce the influence of similar movements, we build the classifier in another way. Note that the movements starting from the same button to different buttons usually are not similar movements. Thus, we classify all the centroids into 11 categories according to the starting positions, and each category has 10 centroids for the corresponding 10 movements. Figure 9 shows the 10 centroids for the corresponding movements of the category starting from button "1." The ordinate is the range of normalization. The red solid line and the blue dashed line denote the roll vector and the pitch vector of the centroid, respectively. The centroids within one category are different from each other since they are not similar movements. Thus, if we know the starting position, we can achieve a high accuracy on the movement inference. However, this inference relies on the starting position, which may lead to cumulative errors. To solve this problem, several candidates for each starting position are taken into consideration during movement inference.

4.4. Movement Identification. In the classifier, 110 centroids of movements are divided into 11 categories. For a password sequence with four movements, we infer each movement separately by calculating the similarity between each unknown movement and the centroids of movements in each category.

First Movement Inference. First movement inference is indeed the first tapped position inference. Instead of matching the first unknown movement to all 110 centroids of movements, it will only be matched with 10 centroids at the specific category starting from the air. The smaller matching cost to a centroid, the larger similarity probability it has. Since movement inference severely depends on the starting position, we choose several movements with smallest matching costs rather than only one as the possible candidates to the unknown movement. The ending position of each candidate to the unknown movement will be regarded as a candidate for the first tapped position, which can further decide which category the following unknown movement belongs to.

Following Movements Inference. The following movements inference is similar to the first movement inference. The only difference is that the unknown movement will be matched with centroids at several categories since there may exist several possible candidates for the starting position. Each category is corresponding to one possible candidate for the starting position. Similarly, several movements with smallest matching costs are chosen by the possible candidates to the unknown movement, and the ending position of each candidate is regarded as the starting position to decide which category the next unknown movement belongs to.

Given one unknown movement, suppose μ possible candidates are selected for it. Let φ_i denote the matching cost between the unknown movement and the i_{th} candidate. The similarity probability of the unknown movement to the i_{th} candidate is calculated as $(1/\varphi_i)/(1/\varphi_1 + \cdots + 1/\varphi_\mu)$. That is, the smaller matching cost, the larger similarity probability.

Matching Cost Calculation. Let $\mathbf{U} = [\mathbf{U}_p; \mathbf{U}_r]$ denote the feature vector of an unknown movement consisting of a pitch vector $\mathbf{U}_p$ and a roll vector $\mathbf{U}_r$. We want to calculate the matching cost between $\mathbf{U}$ and each centroid $C(\mathbf{F})$ of movements. Note that the number of elements in $\mathbf{U}_p$ or $\mathbf{U}_r$ may not be 30, since movements take different durations and the sampling rate is fixed with Hz. Let m denote the actual number of readings in the pitch and roll vectors. We do not

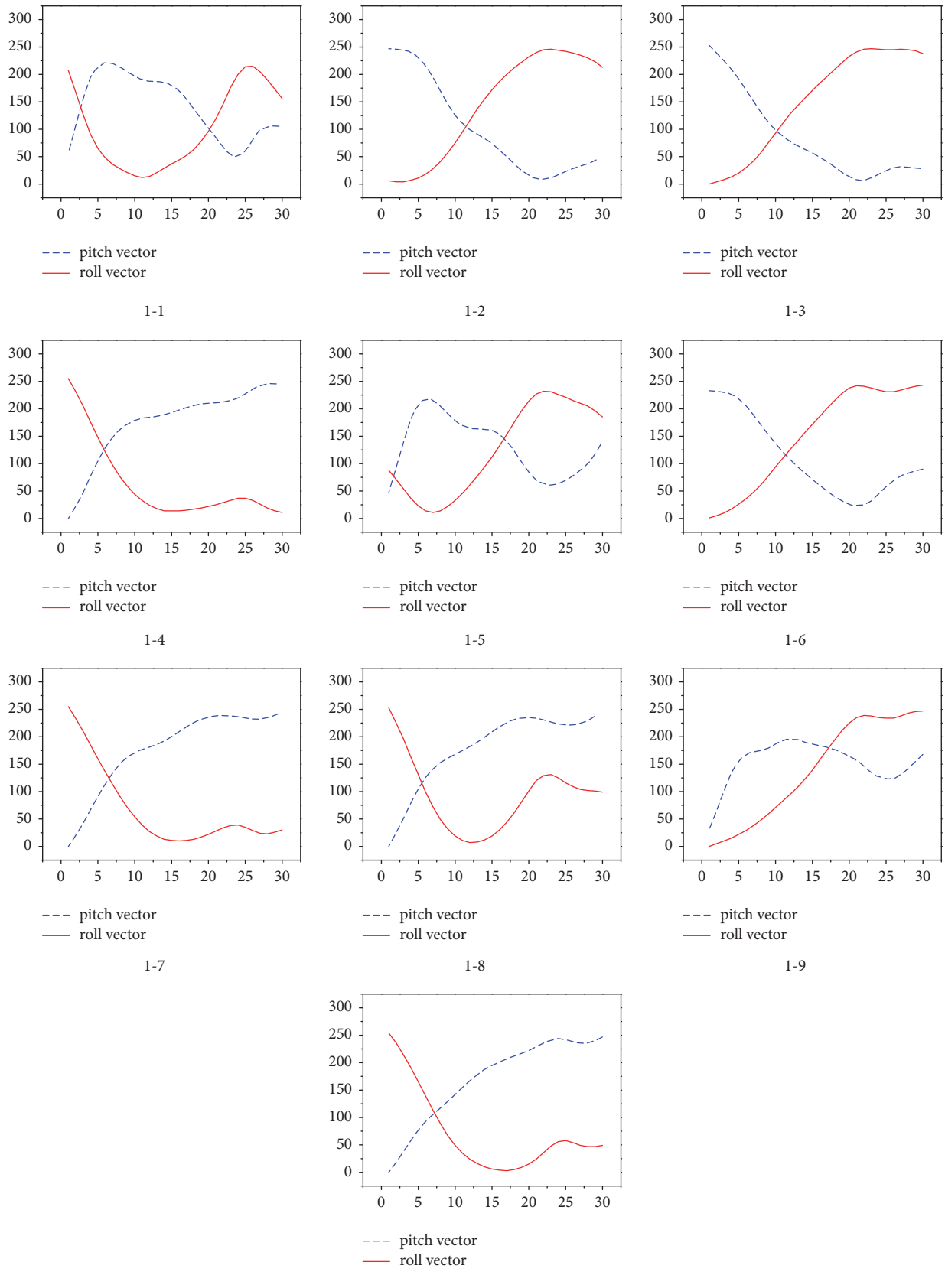


Figure 9: The centroids for the category starting at button "1" and ending at different buttons.

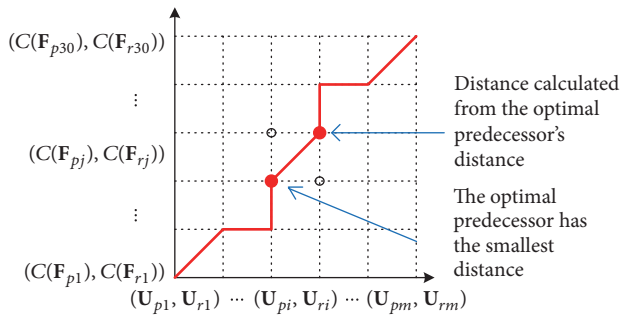


Figure 10: Illustration of matching cost calculation with the DTW algorithm.

scale $\mathbf{U}_p$ or $\mathbf{U}_r$ to the size of 30 elements for keeping data integrity as much as possible.

Figure 10 shows an example of calculating the matching cost of two vectors. The y-axis denotes the values of pitch and roll angles in $C(\mathbf{F})$, and the x-axis denotes the values of pitch and roll angles in $\mathbf{U}$. The matching cost is the length of the shortest path from coordinate (1, 1) to coordinate $(m, 30)$. Euclidean distance is used to calculate the distance from one coordinate to another. Note that the matching cost exhibits suboptimal substructure. Therefore, we adopt the DTW algorithm [16, 31] which employs dynamic programming to calculate the matching cost.

Figure 11 shows an example of movement inference when one candidate position of the starting button is "1." The unknown movement will be matched with all the centroids in category starting from button "1." As shown in Figure 11, the red number on each button is the matching cost and smaller cost means higher similarity. The movement ending at button "7" has the smallest matching cost and is regarded as the most likely candidate. But the buttons in this layout are very close to each other, such that buttons "4," "8," and "0" are neighboring buttons. If we only select the label with the smallest cost as the candidate, the true movement may not be covered. The costs of movements ending in "4" and "0" are also much smaller than the other buttons. Therefore, we select three movements ending at "7," "4," and "0" as the candidates for the unknown movement, and their similarity probability can be calculated accordingly.

4.5. Password Inference. We regard password unlocking as a typical Markov process as the transition probability to the next button is only related to the current button. People tend to use some personal and important information as the password (e.g., birthday) [32]. By taking the side information [32] of users into consideration, we use a Markov chain to model the unlocking process and infer the most possible candidates for each password.

Figure 12 shows the Markov model for the 4-digit password. It contains 5 layers, where layer 0 represents the air and the other layers each correspond to one tapping position on the screen. Since there are 10 numbers ranging from 0 to 9 on the screen, there are 10 possible positions at each layer. As shown in Figure 12, a path from a position at layer 0 to a position at layer 4 represents a sequence of movements for a 4-digit password.

Let ℓ_k^i denote the button i at the kth layer. A movement "$i - j$" from kth layer to the next layer can be represented by (ℓ_k^i, ℓ_{k+1}^j), which is a directed line on the model. Let $T(\ell_k^i, \ell_{k+1}^j)$ denote the transition probability on the directed line, which is equivalent to the similarity probability of the unknown movement to this movement. That is,

$$T\left(\ell_k^i, \ell_{k+1}^j\right) = p\left(\ell_k^i, \ell_{k+1}^j\right), \tag{4}$$

where $p(\ell_k^i, \ell_{k+1}^j)$ denotes the similarity probability between the unknown movement and movement "$i - j$."

The probability of a path is calculated as the multiplication of the transition probability of each movement on the path.

$$P = \prod_{k=0}^{3} T\left(\ell_k^i, \ell_{k+1}^j\right). \tag{5}$$

For example, the probability of a sequence "4-8-2-6" is calculated as $P(4826) = T(\ell_0^4)T(\ell_1^4, \ell_2^8)T(\ell_2^8, \ell_3^2)T(\ell_3^2, \ell_4^6)$, as shown in Figure 12. There are 10^4 possible paths on the graph, but it is not necessary to calculate the probability of each path. Meanwhile, as we mentioned in movement inference, if we only consider one movement at each layer, it may result in cumulative errors due to inaccurate starting position.

In our system, we always use multiple candidates rather than one for each unknown movement to avoid the error due to inaccurate starting position. For an unknown movement, two centroids with the largest similarity probability to it will be selected as the candidates. In particular, we select 3 possible candidates for the first unknown movement considering its ending position is the first tapped position, in order to ensure that the true beginning position is covered. The experimental results in the evaluation section show that the accuracy of the first tapped position inference can reach 97% with three attempts. In this case, there are a total of 24 possible paths left on the graph for a 4-digit password and their probabilities can be calculated accordingly. Finally, the possible paths will be ranked in terms of the probability. We can try the sequence on each path one by one to see whether it is the true password.

5. Evaluation

In this section, we evaluate the performance of Niffler with real experiments in both user-dependent and user-independent environments. The former evaluation will answer the question of *how Niffler performs if an attacker can get sufficient samples from victim's device for training*, and the latter evaluation will answer the question of *how Niffler performs without victims' samples for training, which is the scalability of Niffler*.

In this paper, we compare Niffler with the native random guess method. Although several password inference models have been proposed, it is difficult to compare with them due to the following reasons. First, it is difficult to implement

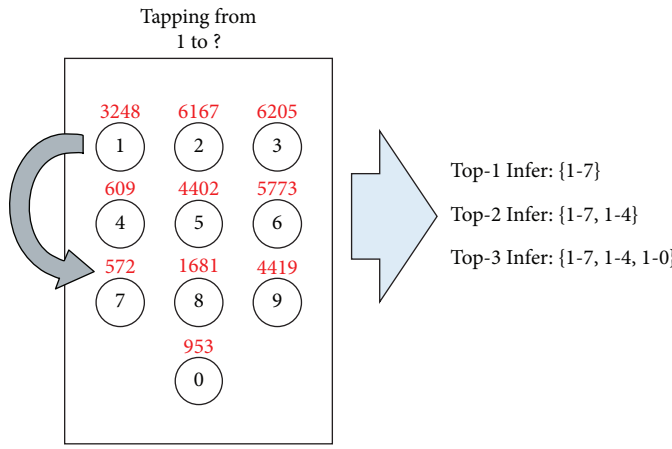


Figure 11: Example of movement inference when the starting position is "1."

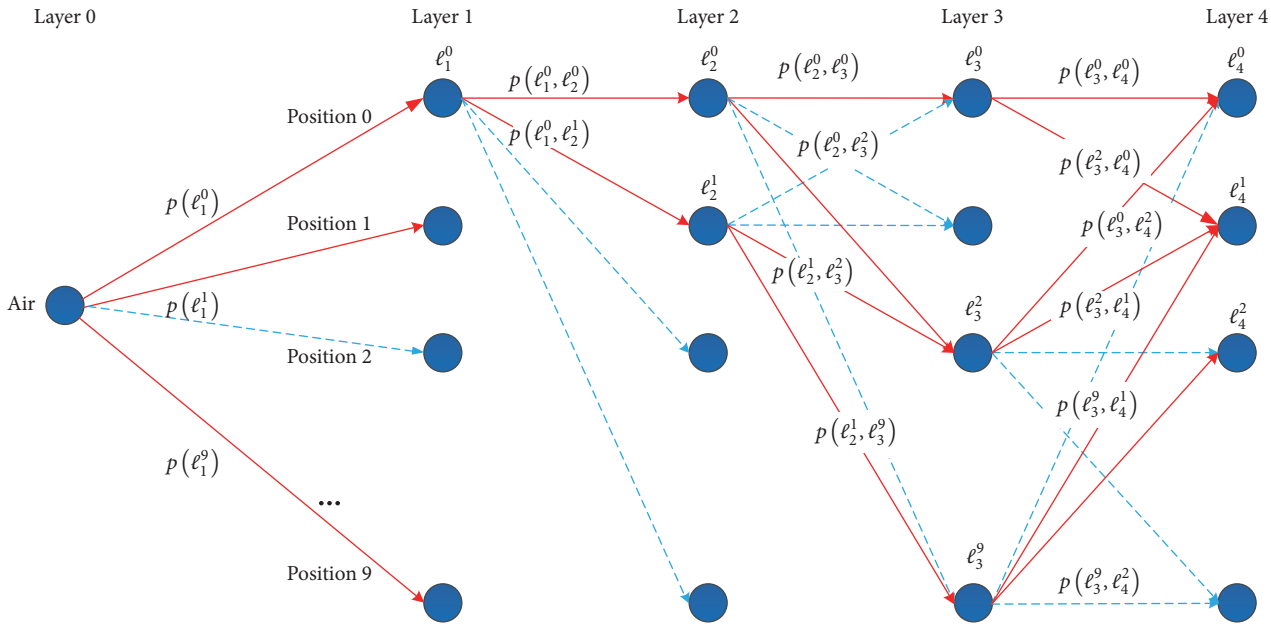


Figure 12: Illustration of password inference with the Markov model.

some of the previous works [10, 14] in real systems as their assumptions are strong. For example, they assume that an attacker can collect sufficient samples from a target user for training purpose, which is actually not realistic for a real attack. In contrast, Niffler is a general side-channel attack model without these assumptions. Second, some other models like [11, 12] demand very high precision in features extraction which are difficult to be extracted in practice due to the large noise in data with high sampling frequency. This also implies that these models are sensitive to noise and are not effective in real attacks.

5.1. Data Collection. We recruit 25 participants to collect samples for unlocking processes. There are 13 males and 12 females, and all of them are between 18 and 30 years old. They can try as many passwords as they want and the ground truth data are recorded as samples. The passwords they used are well designed: some of them are random and some of them are designed with side information. In addition, we also consider the extreme cases, such as "1111" and "9999." The participants are asked to hold and unlock their smartphone with left hand. As shown in Table 1, three types of smartphones with different mainstream screen sizes are used in our experiments. The sampling frequency is about 50 Hz. All participants are asked to keep their body static when they unlock smartphones. Finally, more than 100,000

Table 1: Device list.

Model	Screen size	CPU	Sample rate	Android
Meizu Mx5	5.5 inches	Helio X10	~50 HZ	V5.0
Nexus 4	4.7 inches	APQ8064	~50 HZ	V4.1
OPPO R8	5.2 inches	Cortex-A53	~50 HZ	V4.4

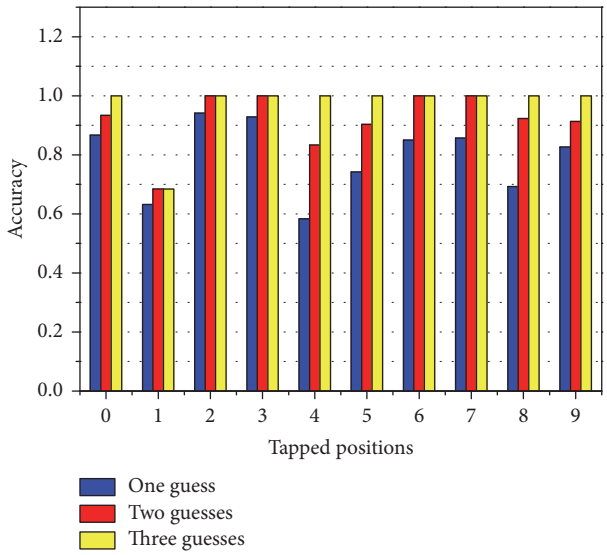


Figure 13: Inference accuracy of first tapped position in user-dependent environment.

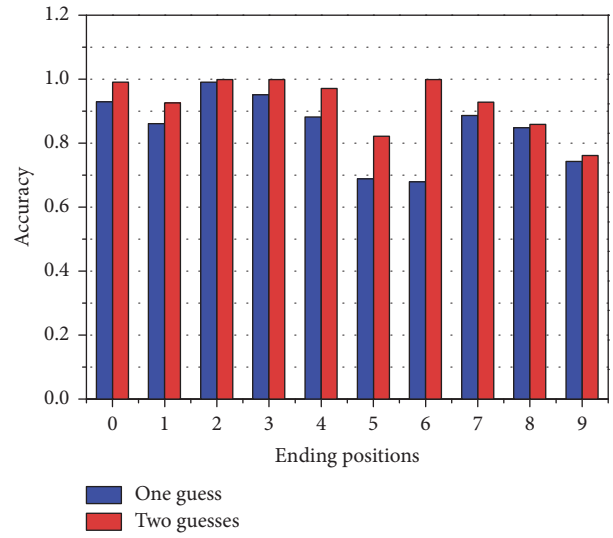


Figure 14: Inference accuracy of category "1" in user-dependent environment.

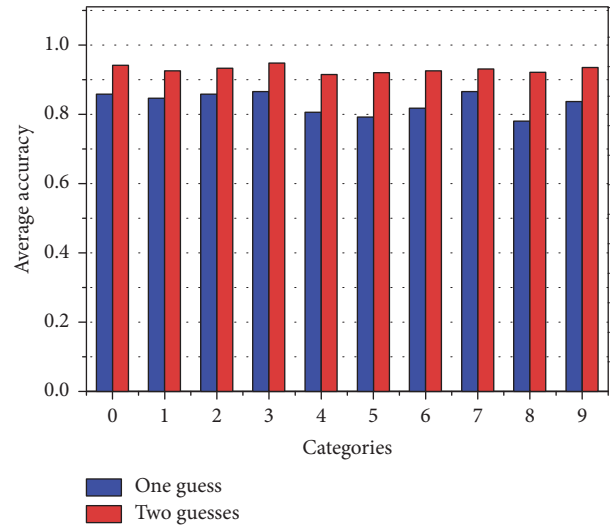


Figure 15: Average accuracy of 10 categories' movement inference in user-dependent environment.

samples were collected over two months, which can be used for training and testing purposes.

5.2. Evaluation in User-Dependent Environments. The data used in this experiment were collected from 15 participants. Each of them provided a dataset containing 3,500 samples for training and another 300 samples for testing. We first evaluate the accuracy of identifying the movements starting in the air, because it is different from other movements and is very important for the following movements inference. We then present the identification of movements of the category "1" as an example. Finally, we show the results of whole password inference.

Figure 13 shows the inference accuracy of the first tapped position inference which is defined as the number of successful inferences divided by the total number of samples. With one guess, the inference accuracy of each position ranges from 58.33% to 94.12% and the average inference accuracy of ten positions is about 79.19% which is much higher than random guess (10%). The inference accuracy for each starting position increases with more number of guesses and can reach 97% with three attempts. This implies that 3 candidates with the largest similarity probability can cover the true first tapped position with almost 100% guarantee, so we use 3 candidates for the first tapped position inference during password inference.

Figure 14 shows the inference accuracy of movements at category "1." The x-axis refers to the ending positions of the movements starting at button "1." With one attempt, the average inference accuracy is about 84.58% which is much higher than TouchLogger 71.5% [12]. With one more attempt, the inference accuracy increases and ranges from 76.15% to 100%.

Figure 15 shows the average inference accuracy of movements in each category, except for the movements starting in the air. The x-axis refers to each category. We can see that the average inference accuracy is more than 80% for all categories with only one attempt, and the average accuracy reaches 90% given one more attempt. This implies that our inference model is robust to different movements and also implies that 2 candidates with the largest similarity probabilities can cover the true movement with more than 90% probability in user-dependent environments.

We present a comparison in terms of the experimental results among our work and typical side-channel attacks exploited motion sensors in Table 2. In our experimental environment, Niffler seams to achieve the highest average accuracy of 84.6% for one key inference on soft keyboards. But, for the inference of a movement depends on the result of last inference, the accumulated error would increase and

Table 2: Comparison with previous works that exploited motion sensors.

Source study	Sensor	Platform	Accuracy
Niffler	Accelerometer	Android	84.6% (72%)
Owusu et al. [8]	Accelerometer	Android	63%
Mehrnezhad et al. [18]	Accelerometer Linear accelerometer Gyroscope Orientation	Android	69%
Mehrnezhad et al. [9]	Accelerometer Orientation	Android	65%
Ping et al. [33]	Accelerometer Gyroscope	Android	75%
Miluzzo et al. [26]	Accelerometer Gyroscope	Android and iOS	72%
Xu et al. [11]	Accelerometer Gyroscope	Android	33.4%

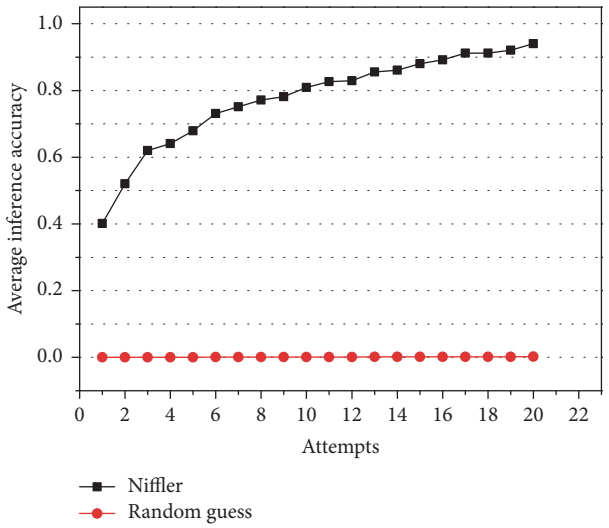


Figure 16: Average inference accuracy over multiple attempts for inferring 4-digit password in user-dependent environment.

Niffler obtains inference accuracy about 72% in effect. Nevertheless, Niffler still performs better than several previous works [8, 9, 11, 18] in user-dependent environment (the data of comparison experiment is collected by our participants, please contact us if you have any questions).

Figures 13–15 show the inference accuracy when the starting position is known to us. We then show the inference of the whole password without knowing the starting position. We ask each one of the five participants to randomly type 25 4-digit passwords. We use Niffler to infer the possible candidates for each password. We try each candidate one by one from the highest probability to the smallest and see whether it is the true password. Once a candidate matches the password, it is a successful inference.

As shown in Figure 16, we can see that the accuracy of password inference increases with more number of attempts. The inference accuracy is about 40% even with only one attempt and can reach 70% with 5 attempts, while random guess only has 0.1%. For a smartphone with Android operation system, the inference accuracy is about 95% with 20 attempts, which means that we can unlock smartphones with 95% successful rate. Therefore, the results indicate that our system based on grip changes achieves a very good attack performance.

5.3. Evaluation in User-Independent Environments. In this section, we evaluate the performance of Niffler in user-independent environments. We collected 50,000 samples in total from 10 participants and mix these data into one dataset to train the classifier. The testing samples are collected from another 5 participants.

Figure 17 shows the inference accuracy for category "1." The x-axis refers to the ending positions of the movements starting at button "1." With one attempt, the average inference accuracy is about 64.26% which is smaller than the average inference accuracy in user-dependent experiments (84.58%). The predication accuracy increases with more attempts, and the accuracy difference between user-dependent and user-independent becomes smaller with more attempts. The average inference accuracy reaches 90% with two attempts, which indicates that 2 candidates with the largest similarity probabilities also can cover the true movement with more than 90% probability in user-independent environments. That is why we only consider 2 possible candidates for each movement during password inference.

Figure 18 shows the inference of the whole password without knowing the starting position in user-independent environments. We can see that the accuracy of password inference increases with more number of attempts. The inference accuracy with one attempt is about 23%, which is smaller than 40% in user-dependent environments, but it is much higher than random guess. The accuracy reaches 55% with 5 attempts, which means there is a half probability we

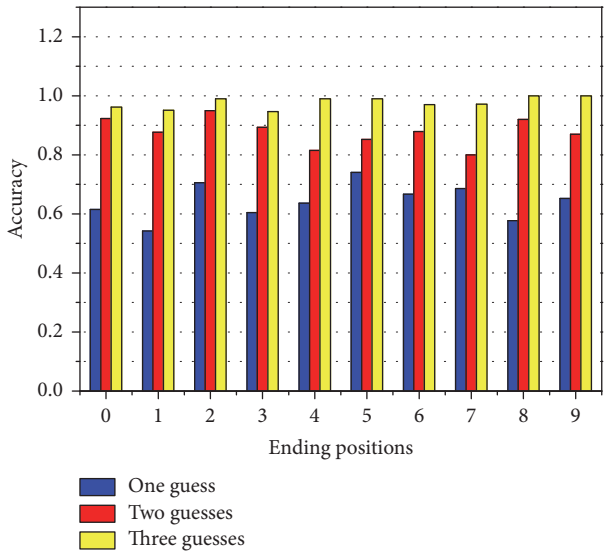


Figure 17: Inference accuracy for category "1" in user-independent environment.

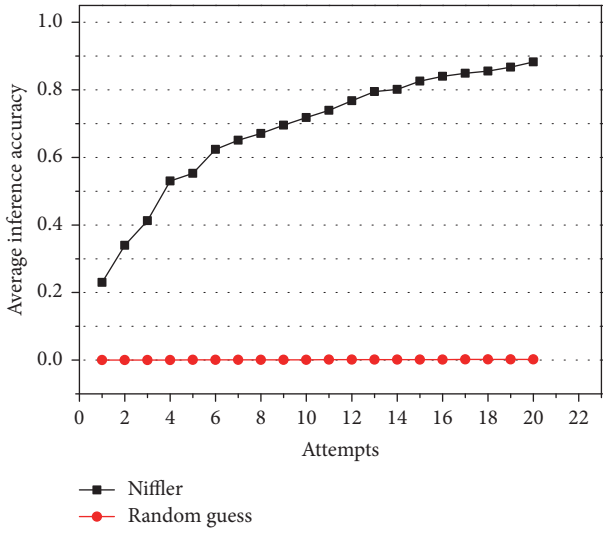


Figure 18: Average inference accuracy over multiple attempts for inferring 4-digit password in user-independent environment.

can unlock screen with 5 attempts. For a smartphone with Android operation system, the inference accuracy is about 74% with 10 attempts, which is close to the accuracy in user-dependent environments. We can conclude that the attack model is more effective in user-dependent environments and can also achieve a high inference accuracy in user-independent environments.

Table 3 shows the results of comparison with some other motion sensor based side channels in user-independent environment. The second column expresses the average inference accuracy for single typed position with one attempt based on our data set. Compared with Table 2, the performances of all of the attacks turn down sharply in user-independent environment, but Niffler still achieves the highest accuracy with 65% even considering accumulated error (42.3%).

Table 3: Comparison with previous works that exploited motion sensors in user-independent environment.

Source study	Accuracy
Niffler	65% (42.3%)
Owusu et al. [8]	40.9%
Mehrnezhad et al. [18]	30.2%
Mehrnezhad et al. [9]	34.8%
Ping et al. [33]	35.9%
Miluzzo et al. [26]	31.4%
Xu et al. [11]	29.6%

Summary. We conduct experiments to evaluate the performance of Niffler in both user-dependent and user-independent environments. The results show that Niffler achieves better performance in user-dependent environments since the testing data and the training date are from the same source. However, Niffler also achieves much better performance in user-independent environments than other side channels. It can achieve about 90% inference accuracy for movement inference with three attempts and 55% inference accuracy for the whole password sequence with 5 attempts. The inference accuracy increases with more attempts and can reach 74% with 10 attempts. We can see that our attack model based on grip changes can achieve a good inference accuracy and is user-independent.

5.4. Evaluation on Influence Factors

The Size of Training Samples. In the experiments above, each centroid is trained with about 100 samples. One question is: *How would our model perform as the number of training samples increases?* That is, we are interested in the performance of the attack model with more training samples. We first construct the classifier with 1,000 samples for each and increase the number of training samples to double size and triple size. We compare the average accuracy with one attempt, two attempts, and three attempts, respectively. All of the samples were collected from the participants with similar hand size and holding habit.

Figure 19 shows the inference accuracy under different sizes of training samples for one, two, and three attempts, respectively. The y-axis is the average inference accuracy and the x-axis is the ID of categories. As shown in Figure 19(a), the inference accuracy increases for most categories with more training samples. However, this is not true for some categories (e.g., categories 2, 3, 4, and 6), which may be because the added samples contain too many abnormal samples that influence the inference. We can also observe that the inference accuracy is better with more attempts, and most of them are larger than 90%. Therefore, the results suggest that the performance would be better with more training samples. The results suggest that Niffler performs better with more samples for training. Meanwhile, it can achieve a good attack performance even with limited number of samples.

Extreme Case. We also evaluate the performance of Niffler for extreme cases. In particular, ten special cases are evaluated

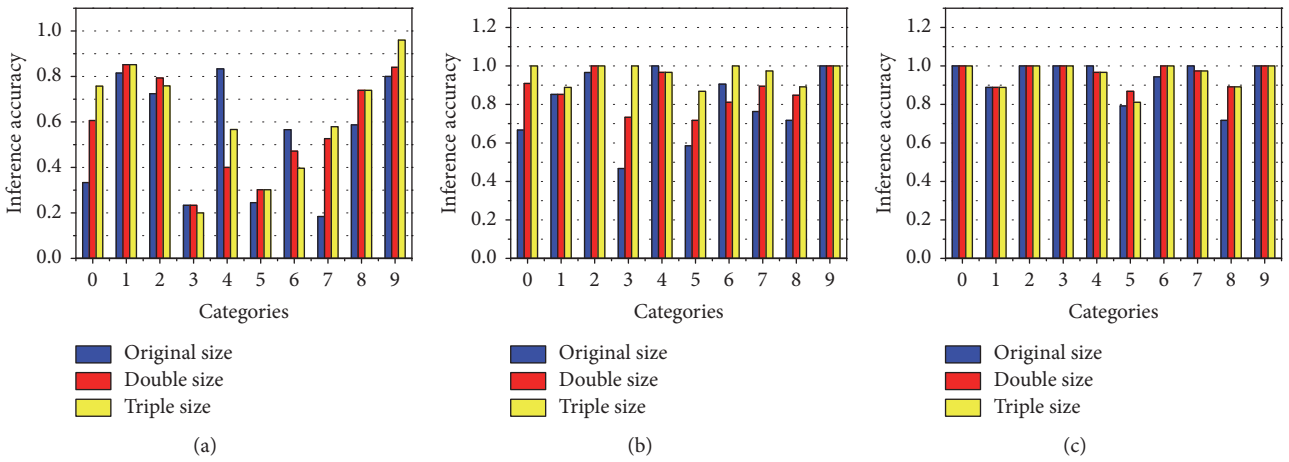


Figure 19: The inference accuracy under different sizes of training samples for (a) one attempt, (b) two attempts, and (c) three attempts, respectively.

Table 4: The impact of hand size on movement identification.

Approximate ratio	Average accuracy
0.93	77.2%
0.96	82.4%
0.98	91.1%
1.01	88.3%
1.04	84.3%
1.07	81.1%

from "0000" to "9999." We consider two scenarios: the first tapped position is known to us and we infer the rest 3 positions, and none of the passwords are known to us and we infer the whole password. Figure 20 shows the average accuracy for the two scenarios with 10 attempts.

We can observe that the accuracy of password inference is better if the first tapped position is known to us, which is about 25% higher. The accuracy of password inference is larger than 90% if the first tapped position is known, while it is about 65% if not. Therefore, the performance of Niffler highly depends on the inference of the first tapped position for extreme cases. It is worth noting that Niffler performs better for general passwords than passwords in extreme cases, which can be observed from Figures 18 and 20. This is because the passwords in extreme cases are too extreme that it is difficult to leverage other methods, such as side information or screening, to improve the inference accuracy.

Hand Size. The basic idea of Niffler is based on the grip change of a smartphone during user holding and tapping. The size of hands might affect the performance of Niffler. In our experiments, we also evaluate the performance of Niffler under different sizes of hands. Table 4 shows the results of the impact of hand size on movement identification. The right column is the average movement identification accuracy with two attempts, and the left column is the approximate ratio between the size of testing sample providers' hands and that of training sample providers' hands. We found that

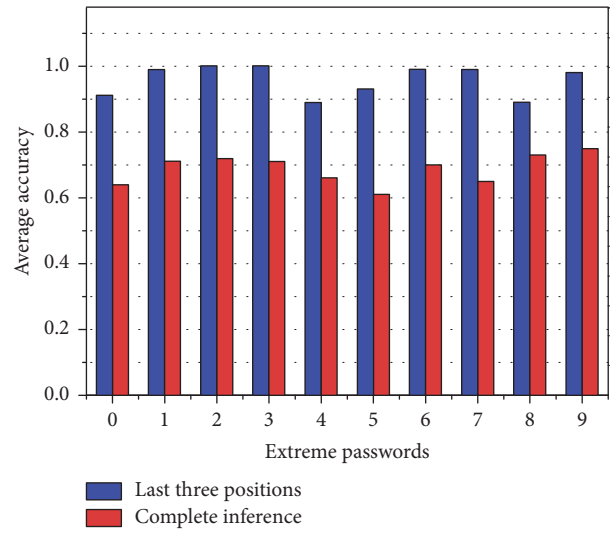


Figure 20: The average accuracy for extreme passwords within 10 attempts.

users with similar hand size produce similar grip changes for the same movement, but the difference of grip changes among different sizes of hands is not obvious. Therefore, Niffler is a general attack model that is robust to the sizes of hands.

5.5. Evaluation on Power Consumption. We also evaluate the power consumption of the monitoring module of Niffler. Power consumption is an important factor [34] for successful attack since a high power-consuming service will be more possible to attract the user's attention and be killed [35]. It is difficult to accurately measure the consumption of the monitoring module. In our experiments, we compare the power consumption of the monitoring module of Niffler with several common applications and services including a preinstalled music player, QQ, a system service, a system

Table 5: Power consumption comparison when the monitoring module is only in monitoring state.

Application	Run time	Consumption
Music player	1 h	14.0%
QQ	1 h	7.2%
Monitoring module of Niffler	1 h	5.3%
System service	1 h	7.2%
System interface	1 h	5.6%
Input method	1 h	1.3%

Table 6: Power consumption comparison when the monitoring module is listening to accelerometer readings.

Application	Run time	Consumption
Music player	1 h	21.3%
QQ	1 h	5.3%
Monitoring module of Niffler	1 h	13.6%
System service	1 h	4.0%
System interface	1 h	3.2%
Input method	1 h	1.0%

interface, and SOGOU input method. Three participants were asked to do the same experiments on the same Meizu Mx5 mobile phone. For each participant, we first asked him to use the phone for 1 hour. During the 1 hour, all of the tested applications or services were kept running in the background. Specifically, the music player was in active state but did not play any music, and the monitoring module of Niffler did nothing except waiting for the unlocking process. After that, each participant was asked to use the phone for 1 hour. During this 1 hour, the music player started to play music for another 1 hour, and the monitoring module registered *Sensor Listener* and kept listening to accelerometer readings.

Table 5 shows the comparison of power consumption of these services for the first 1 hour, and Table 6 shows the comparison for the other 1 hour. As shown in Table 5, we can see that the monitoring module of Niffler has almost the same power consumption as QQ and is much less than the music player which did not play music. As shown in Table 6, the power consumption of Niffler when it is listening to accelerometer readings is about half of that of music player in playing mode but is also double that of QQ. Note that the monitoring module was kept listening to accelerometer readings for 1 hour, but in real system it only lasts for a few seconds. Therefore, we can conclude that Niffler is not power-consuming which would not significantly attract users' attentions.

6. Related Work

6.1. Side-Channel Attacks. Many kinds of side channels have been used in mobile attacks. Free floating (FF) windows in Android have been used to infer a user's input by Ying et al. [30]. The attackers can pass sandbox mechanism and get time intervals between two nearby keystrokes with a well-designed FF windows application to guess input sequences. Aviv et al. cracked graphical password about a recent user's input by analyzing oily residues on touch screen [36]. According to their methods, an attacker must approach the victim's device before oily residues being destroyed. It seems to be impractical in real attacks. Similarly, fingerprint is used to construct side-channel attacks [27]. They need the support of additional hardware, such as fingerprint powder, to dust the touch screen to reveal fingerprints left from tapping fingers. Zhang et al. proposed WiPass, a system that breaks Android pattern lock at a high success rate by analyzing how the finger movement affects the WiFi signal while drawing the pattern lock [37]. Ambient sensor can also be used for keylogging attacks. Simon and Anderson [38] presented an attack that exploits an ambient sensor to infer user's PIN input. Narain et al. [39] and Gupta et al. [40] showed that tap sounds recorded via stereo-microphones can be used to infer typed text on the touchscreen.

Vision information has also been studied to construct side-channel attacks recently. Raguram et al. [41] used the reflection of objects to recover the typed information. However, it needs a clear vision of the content displayed on the touch screen to realize accurate recovery. Yue et al. [42] proposed a method to break digit-based passwords by analyzing the shadow formation around the fingertip. Similar work in [43] reconstructed patterns by using reflection even when the adversary is around a corner. Shukla et al. [44] exploited video-based side-channel model to decode PIN entry process which relies on the spatiotemporal dynamics of the hands during typing. It is the first work to reconstruct typed text from a video recording of the hands without using any information about the keys being pressed or the content displayed on the screen. Ye et al. proposed a vision-based attack to crack pattern based lock [45]. They exploited the geometric information exposed by fingertip movement to identify patterns without knowing the console geometry, such as the screen size. Hansch et al. [46] demonstrated that the front camera can be used to capture the screen reflections in the eyeballs, which allows inferring user input.

6.2. Motion Sensor-Based Side-Channel Attacks. With the development of embedded systems and mobile devices, sensors have been widely exploited to detect the motion information [47] and infer passwords of touch-enabled screen devices, including accelerometer, orientation sensor, gyroscopic, and other motion sensors [10, 48]. Cai and Chen proposed *TouchLogger* [12] which uses the gyroscope and the orientation sensor as a side-channel attack to infer the location user tapped. They pointed out that keystroke vibration on touch screens is highly correlated to the keys being typed. This discovery became one of most important bases of sensor-based attacks. *Taplogger* [11], a side-channel model constructed with the gyroscope and the accelerometer data, used similar theory to infer tapped sequences. Both of them focus on the relationship between sensor readings and each tap event, which are too sensitive to data source. Subsequent publications [10, 26] also considered the combination of the accelerometer and the gyroscope in order to improve the

performance as well as to infer even longer text inputs [33]. Besides text password, motion sensors are also used to infer pattern password. Andriotis et al. [49] established a scheme, which combines a behavior-based attack and a physical attack on graphical lock screen methods.

Aviv et al. [14] referred to research object as the process of tapping one PIN, instead of individual digits. Their work can reach high accuracy. However, it is almost impossible to build the large corpus containing all possible categories. *ACCessory* [8] proposed by Owusu et al. is similar to above work. The authors demonstrated that the accelerometer can be used as a side-channel attack to infer short sequences of touches on screen and employed standard machine learning techniques to infer input passwords. Our work differs from these pervious methods in that we focus on the movement between nearby keystrokes. Unlike individual tap events or entire process of tapping PINs, we observe that movements contain general characteristics among different users. Using this observation, our side-channel model, Niffler, can be easily implemented without facing previous limitations that training data and inferring data must be collected from the same source.

Noor et al. [50] outlined a machine learning approach modeling the grip changes to predict the resulting touchdown point while the finger is still in the air. They detected physical grip and grip change over time through adding extra hardware (additional sensor arrays on the back or side of the device). Ulteriorly, Negulescu and McGrenere [6] showed that internal motion sensors can achieve similar results in the air predictions of touch points. Similarly, we employ internal sensors to reflect grip changes during user tapping password. Our work differs from Noor et al. and Negulescu and McGrenere in that we discover the general features in grip changes to infer movements between nearby keystrokes and use them to infer the tapped sequences, for example, PIN.

7. Conclusions

In this paper, we design and implement a new sensor-based side-channel attack system, called Niffler, to accurately infer the unlocking passwords on smartphones. Our system is based on the observation that movements during the unlocking process are context-aware and user-independent. We collected 100,000 samples from 25 volunteers over two months for training and testing purposes. Angle feature vectors are extracted to build a multicategory classifier which can further use the DTW algorithm to calculate the similarity between an unknown movement and the centroids in each category. The Markov model is used to model the unlocking process and the paths with the largest probabilities are selected as the candidates for the true password. We conduct experiments in both user-dependent and user-independent environments. The results show that Niffler is a user-independent system and achieves a high accuracy on password inference.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. U1536204, 61502352, 61672394, 61373169, and 61672393), National High Technology Research and Development Program of China (Grant no. 2015AA016004), Natural Science Foundation of Hubei Province and Jiangsu Province (Grants nos. 2017CFB503 and BK20150383), and Fundamental Research Funds for the Central Universities (Grants nos. 2042015kf0016 and 413000035).

References

[1] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, "A Hybrid Cloud Approach for Secure Authorized Deduplication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1206–1216, 2015.

[2] X. Liu, R. Lu, J. Ma, L. Chen, and B. Qin, "Privacy-Preserving Patient-Centric Clinical Decision Support System on Naïve Bayesian Classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 2, pp. 655–668, 2016.

[3] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, "Friendbook: a semantic-based friend recommendation system for social networks," *IEEE Transactions on Mobile Computing*, 2014.

[4] 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.

[5] Z. Wang, R. Tan, J. Hu et al., "Heterogeneous incentive mechanism for time-sensitive and location-dependent crowdsensing networks with random arrivals," *Computer Networks*, vol. 131, no. 2, pp. 96–109, 2018.

[6] M. Negulescu and J. McGrenere, "Grip change as an information side channel for mobile touch interaction," in *Proceedings of the 33rd Annual CHI Conference on Human Factors in Computing Systems, CHI 2015*, pp. 1519–1522, Republic of Korea, April 2015.

[7] A. Nahapetian, "Side-channel attacks on mobile and wearable systems," in *Proceedings of the 13th IEEE Annual Consumer Communications and Networking Conference, CCNC 2016*, pp. 243–247, USA, January 2016.

[8] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "ACCessory: password inference using accelerometers on smartphones," in *Proceedings of the Proceeding of the 13th Workshop on Mobile Computing Systems and Applications (HotMobile '12)*, no. 9, New York, NY, USA, February 2012.

[9] M. Mehrnezhad, E. Toreini, S. F. Shahandashti, and F. Hao, "Stealing PINs via mobile sensors: actual risk versus user perception," *International Journal of Information Security*, pp. 1–23, 2017.

[10] L. Cai and H. Chen, "On the practicality of motion based keystroke inference attack," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7344, pp. 273–290, 2012.

[11] Z. Xu, K. Bai, and S. Zhu, "TapLogger: inferring user inputs on smartphone touchscreens using on-board motion sensors," in *Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 113–124, Tucson, Ariz, USA, April 2012.

[12] L. Cai and H. Chen, "Touchlogger: Inferring keystrokes on touch screen from smartphone motion," *HotSec*, vol. 11, p. 9, 2011.

[13] C. Shen, S. Pei, Z. Yang, and X. Guan, "Input extraction via motion-sensor behavior analysis on smartphones," *Computers & Security*, vol. 53, pp. 143–155, 2015.

[14] A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith, "Practicality of accelerometer side channels on smartphones," in *Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC '12)*, pp. 41–50, ACM, Orlando, Fla, USA, December 2012.

[15] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: user verification on smartphones via tapping behaviors," in *Proceedings of the IEEE 22nd International Conference on Network Protocols (ICNP '14)*, pp. 221–232, Raleigh, NC, USA, October 2014.

[16] A. de Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and i know it's you!: implicit authentication based on touch screen patterns," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, pp. 987–996, ACM, May 2012.

[17] A. Das, N. Borisov, and M. Caesar, "Tracking Mobile Web Users Through Motion Sensors: Attacks and Defenses," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA.

[18] M. Mehrnezhad, E. Toreini, S. F. Shahandashti, and F. Hao, "TouchSignatures: Identification of user touch actions and PINs based on mobile sensor data via JavaScript," *Journal of Information Security and Applications*, vol. 26, pp. 23–38, 2016.

[19] R. Spreitzer, V. Moonsamy, T. Korak, and S. Mangard, "Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices," *IEEE Communications Surveys & Tutorials*, 2017.

[20] S. Hoober, *How do users really hold mobile devices*, vol. 18, Uxmatters, 2013, http://www.uxmatter.com.

[21] J. Song, "The design of bottom layer sensor interfaces based on android os," in *Proceedings of the International Proceedings of Computer Science and Information Technology*, vol. 58, p. 54, 2012.

[22] X. Liu, K. R. Choo, R. H. Deng, R. Lu, and J. Weng, "Efficient and Privacy-Preserving Outsourced Calculation of Rational Numbers," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 27–39, 2018.

[23] X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2401–2414, 2016.

[24] Z. Huang, S. Liu, X. Mao, K. Chen, and J. Li, "Insight of the protection for data security under selective opening attacks," *Information Sciences*, vol. 412-413, pp. 223–241, 2017.

[25] M. Mohamed, B. Shrestha, and N. Saxena, "SMASheD: Sniffing and Manipulating Android Sensor Data for Offensive Purposes," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 901–913, 2017.

[26] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, "Tapprints: your finger taps have fingerprints," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*, pp. 323–336, Ambleside, UK, June 2012.

[27] Y. Zhang, P. Xia, J. Luo, Z. Ling, B. Liu, and X. Fu, "Fingerprint attack against touch-enabled devices," in *Proceedings of the 2nd ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '12)*, pp. 57–68, Raleigh, NC, USA, October 2012.

[28] K.-E. Kim, W. Chang, S.-J. Cho et al., "Hand grip pattern recognition for mobile user interfaces," in *Proceedings of the AAAI*, vol. 21, p. 1789, 2006.

[29] J. Heaton, "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618," *Genetic Programming and Evolvable Machines*, pp. 1–3, 2017.

[30] L. Ying, Y. Gu, Y. Chengy, P. Su, Y. Lu, and D. Feng, "Attacks and defence on android free floating windows," in *Proceedings of the 11th ACM Asia Conference on Computer and Communications Security, ASIA CCS 2016*, pp. 759–770, China, June 2016.

[31] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.

[32] SplashData, "Announcing our worst passwords of 2015," 2015, https://www.teamsid.com/worst-passwords-2015/.

[33] D. Ping, X. Sun, and B. Mao, "TextLogger: Inferring longer inputs on touch screen using motion sensors," in *Proceedings of the 8th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2015*, USA, June 2015.

[34] W. Guo, J. Li, G. Chen, Y. Niu, and C. Chen, "A PSO-Optimized Real-Time Fault-Tolerant Task Allocation Algorithm in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3236–3249, 2015.

[35] Y. Liu, C. Xu, and S. C. Cheung, "Where has my battery gone? Finding sensor related energy black holes in smartphone applications," in *Proceedings of the 11th IEEE International Conference on Pervasive Computing and Communications, PerCom 2013*, pp. 2–10, USA, March 2013.

[36] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," *WOOT*, vol. 10, pp. 1–7, 2010.

[37] J. Zhang, X. Zheng, Z. Tang et al., "Privacy leakage in mobile sensing: your unlock passwords can be leaked through wireless hotspot functionality," *Mobile Information Systems*, vol. 2016, Article ID 8793025, 14 pages, 2016.

[38] L. Simon and R. Anderson, "PIN skimmer: Inferring PINs through the camera and microphone," in *Proceedings of the 3rd ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, SPSM 2013, Held in Association with the 20th ACM Conference on Computer and Communications Security, CCS 2013*, pp. 67–78, Germany, November 2013.

[39] S. Narain, A. Sanatinia, and G. Noubir, "Single-stroke language-agnostic keylogging using stereo-microphones and domain specific machine learning," in *Proceedings of the 7th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2014*, pp. 201–212, UK, July 2014.

[40] H. Gupta, S. Sural, V. Atluri, and J. Vaidya, "Deciphering text from touchscreen key taps," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9766, pp. 3–18, 2016.

[41] R. Raguram, A. M. White, D. Goswami, F. Monrose, and J.-M. Frahm, "iSpy: Automatic reconstruction of typed input from compromising reflections," in *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS '11)*, pp. 527–536, Chicago, Ill, USA, October 2011.

[42] Q. Yue, Z. Ling, B. Liu, X. Fu, and W. Zhao, "Blind recognition of touched keys: Attack and countermeasures," https://arxiv.org/abs/1403.4829.

[43] Y. Xu, J. Heinly, A. M. White, F. Monrose, and J.-M. Frahm, "Seeing double: Reconstructing obscured typed input from repeated compromising reflections," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS 2013*, pp. 1063–1074, Germany, November 2013.

[44] D. Shukla, R. Kumar, V. V. Phoha, and A. Serwadda, "Beware, your hands reveal your secrets!," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 904–917, ACM, Scottsdale, Ariz, USA, November 2014.

[45] G. Ye, Z. Tang, D. Fang et al., "Cracking Android Pattern Lock in Five Attempts," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, USA.

[46] R. Hansch, T. Fiebig, and J. Krissler, "Security impact of high resolution smartphone cameras," in *Proceedings of the 23rd USENIX Security Symposium*, 2014.

[47] Y. J. Lee, "Detection of Movement and Shake Information using Android Sensor," in *Proceedings of the Multimedia 2015*, pp. 52–56.

[48] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp)iPhone: Decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS'11*, pp. 551–562, USA, October 2011.

[49] P. Andriotis, T. Tryfonas, G. Oikonomou, and C. Yildiz, "A pilot study on the security of pattern screen-lock methods and soft side channel attacks," in *Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '13)*, pp. 1–6, April 2013.

[50] M. F. M. Noor, A. Ramsay, S. Hughes, S. Rogers, J. Williamson, and R. Murray-Smith, "28 frames later: Predicting screen touches from back-of-device grip changes," in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems, CHI 2014*, pp. 2005–2008, Canada, May 2014.