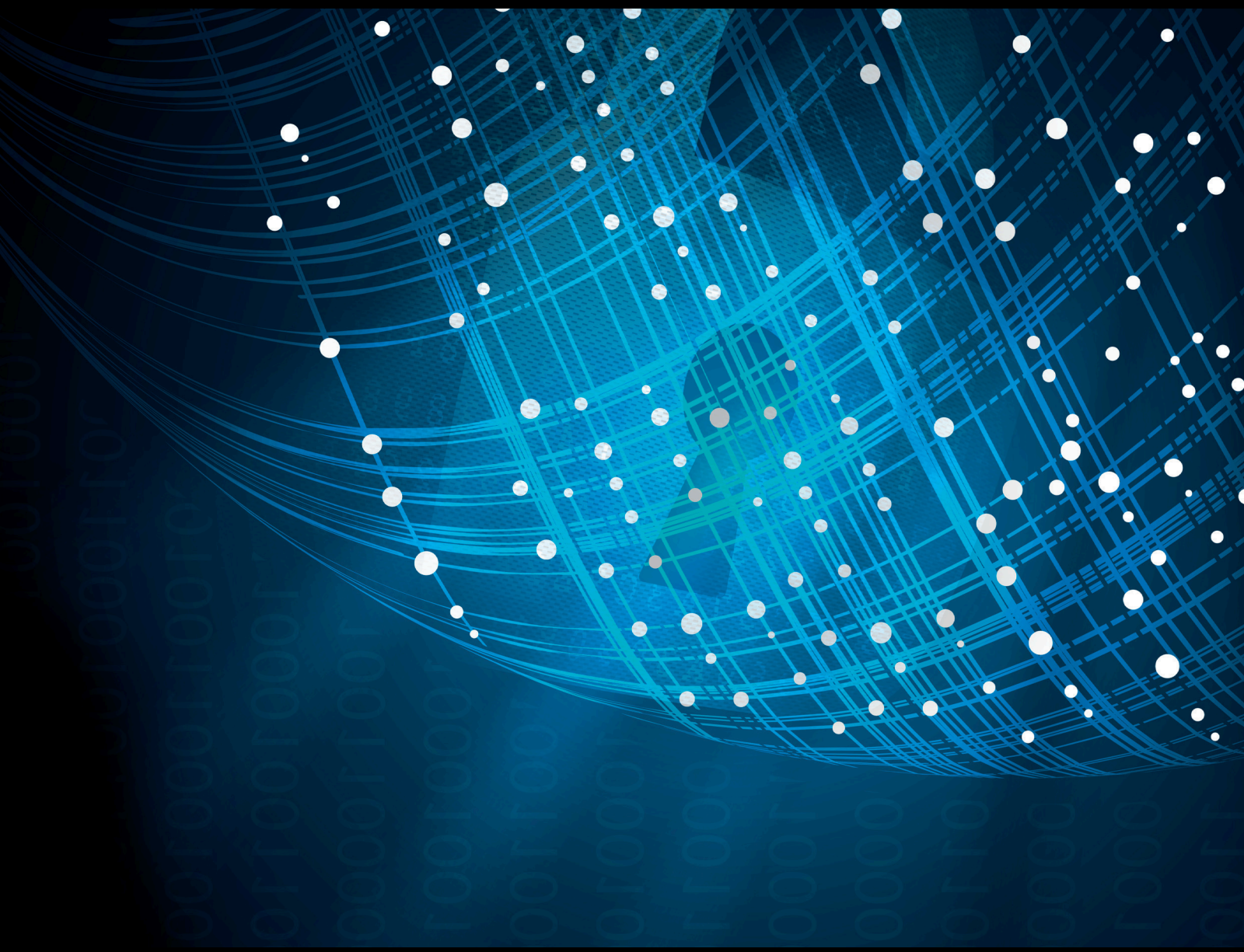
Security and Communication Networks

Security Hardened and Privacy Preserved Vehicle-to-Everything (V2X) Communication

Lead Guest Editor: Azeem Irshad
Guest Editors: Muhammad Shafiq, Shehzad Ashraf Chaudhry, and Muhammad Usman



Security Hardened and Privacy Preserved Vehicle-to-Everything (V2X) Communication

Security and Communication Networks

Security Hardened and Privacy Preserved Vehicle-to-Everything (V2X) Communication

Lead Guest Editor: Azeem Irshad
Guest Editors: Muhammad Shafiq, Shehzad Ashraf Chaudhry, and Muhammad Usman

Contents

Hindawi
Security and Communication Networks
Volume 2022, Article ID 9865621, 4 pages
https://doi.org/10.1155/2022/9865621

Editorial

Security Hardened and Privacy Preserved Vehicle-to-Everything (V2X) Communication

Azeem Irshad,[1] **Muhammad Shafiq**,[2] **Shehzad Ashraf Chaudhry**,[3] **and Muhammad Usman**[4]

[1]*Department of Computer Science and Software Engineering, International Islamic University Islamabad, Islamabad, Pakistan*
[2]*Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Republic of Korea*
[3]*Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey*
[4]*Faculty of Computing Engineering and Science, University of South Wales, Pontypridd CF37 1DL, UK*

Correspondence should be addressed to Muhammad Shafiq; shafiq@ynu.ac.kr

Received 19 March 2022; Accepted 19 March 2022; Published 12 April 2022

Vehicle-to-everything (V2X) communications have recently gained concentration of researchers for both, academia as well as industry. In the V2X system, the information is communicated from vehicle sensors to other vehicles, infrastructure, pedestrians, and mobile network cloud through high-bandwidth reliable links [1–4]. The technology may greatly improve the driver's awareness of imminent hazards, thereby reducing the severity of accidents, fatalities, or possible collisions with other vehicles. The V2X technology brings efficiency through creating warning alerts for drivers, imparting the information of alternative routes for avoiding possible traffic congestions and pinpointing available parking spaces. Such critical situations might become problematic if the security and privacy of V2X communication system is compromised [5–8]. Thus, V2X vehicles along with efficiency, reliability, and safety parameters require more secure and robust communication protocols to meet the upcoming security challenges. Moreover, the wireless nature of the system might become challenging in affording secure and ubiquitous connectivity to the V2X network [9–13]. This is crucial to create a fail-safe infrastructure of modern traffic scenario regarding smart cities since security and privacy issues are quite prevalent in our daily lives.

This special issue encompasses 13 research articles focusing on security and privacy of vehicular networks. The details of these articles are summarized as follows.

The authors in a research article titled "Implementation of Blockchain Consensus Algorithm on Embedded Architecture" presented study of the feasibility as well as the gain realized by using an architecture adopted at Ethereum PoW on FPGAs [14]. The concept of finding optimized solutions adapted to the specific constraints of blockchain-based applications such as execution time, number of required nodes, and suitable data security algorithms are heavily researched in the literature. The paper also presents the implementation of an embedded-blockchain approach. This system presents a hybrid implementation of ethereum nodes on Raspberry Pi on one side and of PoW consensus on FPGA. This may prove to be a significant proposal for future implementations since it provides the possibility to set up an ASIC to accelerate the POW execution.

The authors in a research article titled "New Efficient and Secured Authentication Protocol for Remote Healthcare Systems in Cloud-IoT" proposed new authentication scheme for health care systems cloud-IoT [15]. Before presenting the proposed work, the authors demonstrate the vulnerabilities and security issues of previous proposed studies including Sharma and Kalra's scheme. The authors discover few weaknesses in the Sharma and Kalra's protocol along with password guessing and smart card stolen attacks. The simulation tests as performed under Scyther tool confirm that the lightweight proposed protocol satisfies up-to-date security requirements. The formal and informal analysis also

validates the findings of the obtained results in the performance evaluation.

The authors in a research article titled "Chaotic Reversible Watermarking Method Based on IWT with Tamper Detection for Transferring Electronic Health Record" presented a reversible and lightweight watermarking method for IoT-based healthcare systems employing integer wavelet transform (IWT) and chaotic maps, which is capable of tamper detection [16]. In this study, the authors demonstrated a secure and lightweight watermarking method having imperceptibility and reversibility impacts, with least possible attacks in IoT-based healthcare systems. As per the results, the proposed scheme took advantage of IWT and reduces greatly the computational complexity as compared to other related techniques. Besides, the scheme supports tamper detection and reversibility and is also provably resistant to several signal processing threats.

The research article titled "A Provably Secure Authentication and Key Exchange Protocol in Vehicular Ad Hoc Networks" presents a novel and secure authenticated key agreement scheme to negotiate an agreed session key prior to communicating the confidential information in vehicular ad hoc network (VANET) [17]. In the follow-up of sensing sensitive information, the transmission of information may be affected or tampered due to insecure public wireless channel. Therefore, it becomes critical to secure the transmission. In this context, the proposed protocol achieves the objective by supporting mutual authenticity among the three participating entities including RSU, user, and cloud server. Finally, the formal security analysis depicts that the protocol is workable, efficient, and secure.

The research article titled "Security in Vehicular Ad hoc Networks: Challenges and Countermeasures" discusses the characteristics and all the possible security limitations including attacks and threats at different protocol layers of the VANETs architecture [18]. Moreover, the paper also surveys different countermeasures. This paper surveys VANET security challenges such as DoS, Sybil, impersonation, replay, and other attacks. Furthermore, it presents the possible countermeasures. The survey may serve as a useful reference for future intelligent transport systems.

The research article titled "The Snowden Phone: A Comparative Survey of Secure Instant Messaging Mobile Applications" presents a comprehensive survey on security and privacy analysis of recent and popular instant messaging applications [19]. In this paper, the authors discussed all the necessary prerequisites that a reader needs to do for securing the messaging applications as well as analyzing the mobile applications that help in implementing end-to-end secure messaging. They define the key characteristics of a secure and privacy-preserving communication protocol for instant messaging apps and then perform an analysis on the most popular ones. Furthermore, the authors perform a comparison on the end-to-end encryption protocols. After the analysis, the authors recommend some possible security improvements for all applications under analysis that provide quite interesting highlights. They use different testing scenarios to study the security and usability characteristics of secure mobile applications and provide suggestions for improvements.

The research article titled "A Reliable Network Intrusion Detection Approach using Decision Tree with Enhanced Data Quality" presents a reliable network intrusion detection approach based on decision tree classifier and engineering feature techniques [20]. In the present research paper, authors proposed a new reliable network intrusion detection approach based on decision tree with improved data quality. The authors employed network data preprocessing and entropy decision-based feature selection to enhance quality of training for building decision tree classifier to boost the quality of intrusion detection. The proposed paper is new and significant because it is based on machine learning algorithm. The experimental study depicts that the contributed approach presents many advantages and shows high accuracy in comparison with other state-of-the-art models.

The research article titled "Internet of Things Security: Challenges and Key Issues" aims to study the key issues including security threats in state-of-the-art IoT-based authentication schemes [21]. Mostly, the paper highlights the current challenges as posed to the induction of IoT devices in the precarious domain. The authors emphasized on securing real and virtual worlds based on IoT technology resulting in secure energy, water management, construction, industry, environment, telecommunications, healthcare, surveillance-based sectors, etc. Mostly, IoT-based networks are prone to Denial of Service (DoS) attack, replay attack, and insider attacks. The authors emphasized on countering the mentioned threats by employing one-time password, elliptic-curve cryptography (ECC), ID-based authentication, and certificate-based authentication solutions.

The research article titled "Adaptive Fault-Tolerant System and Optimal Power Allocation for Smart Vehicles in Smart Cities using Controller Area Network (CAN)"aims to analyze the increased energy consumptions and transmission collisions resulting in loss of data packets in a CAN-based smart vehicle system [22]. The authors try to find the fault-tolerant capability through probabilistic automatic repeat request (PARQ) and also probabilistic automatic repeat request (PARQ) with fault impact (PARQ-FI) and also provide the optimal power allocation in CAN sensor nodes for enhancing the performance of the system. The simulation results depict an increase packet delivery ratio of the proposed scheme. The promising findings of the proposed system may prove to be a significant reference for future smart cities.

The research article titled "Designing an Efficient and Secure Message Exchange Protocol for Internet of Vehicles" aims to present a secure message authenticated key exchange protocol for the exchange of information among legitimate participating members of IoV (SMEP-IoV) [23]. Initially, the author reviewed some of the recently presented authentication protocols for securing IoVs. Then, they constructed a symmetric key-oriented authenticated key exchange protocol which can be employed by a vehicle and corresponding RSU to converge on a mutually agreed secret key with the assistance of vehicle server. The presented SMEP-IoV scheme meets the security as well as performance requirements of IoV. To analyze the security on formal basis,

the SMEP-IoV employed BAN logic. According to the demonstrated results, the lightweight SMEP-IoV achieves the desired security properties.

The research article titled "Improved Secure and Lightweight Authentication Scheme for Next-Generation IoT Infrastructure" proposed a secure as well as lightweight authenticated key agreement technique for next-generation IoT infrastructure after reviewing and presenting the weaknesses in Rana et al. [24]. This study suffers from vulnerability to offline password guessing attack and privileged insider threats. The improved scheme solves the drawbacks of reviewed scheme, and its security features are proven under formal as well as informal analysis. They proved the security properties of the scheme using BAN logic as well formal analysis based on the RoR model. Ultimately, they took a comparative analysis of the proposed work and previous related schemes and found that their scheme is not only efficient as far as computational and communicational costs are concerned but also robust regarding the security features. Moreover, the performance evaluation also acknowledges the effectiveness of proposed scheme in terms of time and memory consumption.

The research article titled "AVoD: Advanced Verify-on-Demand for Efficient Authentication against DoS Attacks in V2X Communication" presented a technique for preventing Denial-of-Service (DoS) threats in the interaction of autonomous cooperative driving vehicles by employing security credential management system [25]. The contributed technique minimizes the authentication costs on the basis of classification for similar messages into several categories, while verifying the authenticity for the first message as characterizing the group. This scheme has been duly tested with experiments and demonstrations, while the scheme significantly enhances the speed of processing messages by reducing DoS attacks, attributing to the contributed scheme.

The research article titled "V2X-Based Mobile Localization in 3D Wireless Sensor Network" presented a range-free localization algorithm with respect to sensors in 3D wireless sensor network architecture on the basis of flying anchors [26]. The developed algorithm is quite suitable for localization of the vehicle as it employs the vehicle-to-infrastructure (V2I) based positioning algorithm. It chooses the multilayer C-shaped trajectory for random walk of the mobile anchor-based nodes which is installed with GPS. These anchor nodes keep transmitting beacon signal besides the information of position towards unknown nodes to form a triangle using three further nodes upon receipt of RSSI values. Thereafter, distance is calculated using link quality induction for every mobile anchor node using centroid-based formula for computing localization error. The results of simulation indicate that C-CURVE algorithm demonstrates higher efficiency even in multipath fading. The presented algorithm affords higher accuracy despite the presence of noise, due to the employment of recurring LQI values.

We would like to extend our profound appreciation to all the reviewers and authors for their timely and worthy contributions. Moreover, we would like to thank the Editor-in-Chief of *Security and Communication Networks*, Hindawi, for granting us the privilege to contribute this special issue in the worthy journal. We hope this special issue will provide useful insight to the researchers seeking for the novel prospects to secure vehicular communications.

Conflicts of Interest

The guest editors declare that there are no conflicts of interest regarding the publication of this special issue.

Azeem Irshad
Muhammad Shafiq
Shehzad Ashraf Chaudhry
Muhammad Usman

References

[1] P. Papadimitratos, L. Buttyan, T. Holczer et al., "Secure vehicular communication systems: design and architecture," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100–109, 2008.

[2] Z. MacHardy, A. Khan, K. Obana, and S. Iwashina, "V2X access technologies: regulation, research, and remaining challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1858–1877, 2018.

[3] G. Karagiannis, O. Altintas, E. Ekici et al., "Vehicular networking: a survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 584–616, 2011.

[4] S. Jiang, X. Zhu, and L. Wang, "An efficient anonymous batch authentication scheme based on HMAC for VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2193–2204, 2016.

[5] R. Hussain, F. Hussain, S. Zeadally, and J. Lee, "On the adequacy of 5G security for vehicular ad hoc networks," *IEEE Communications Standards Magazine*, vol. 5, no. 1, pp. 32–39, 2021.

[6] J. Miao, Z. Wang, X. Miao, and L. Xing, "A secure and efficient lightweight vehicle group Authentication protocol in 5G networks," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 4079092, 12 pages, 2021.

[7] H. U. Rahman, A. Ghani, I. Khan, N. Ahmad, S. Vimal, and M. Bilal, "Improving network efficiency in wireless body area networks using dual forwarder selection technique," *Personal and Ubiquitous Computing*, vol. 26, no. 1, pp. 11–24, 2022.

[8] X. Li, J. Liu, M. S. Obaidat, P. Vijayakumar, Q. Jiang, and R. Amin, "An unlinkable authenticated key agreement with collusion resistant for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7992–8006, 2021.

[9] P. R. Babu, R. Amin, A. G. Reddy, A. K. Das, W. Susilo, and Y. Park, "Robust authentication protocol for dynamic charging system of electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 11338–11351, 2021.

[10] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman, "Securing smart city surveillance: a lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, vol. 8, pp. 43711–43724, 2020.

[11] G. U. Rehman, A. Ghani, M. Zubair, S. A. Ghayyure, and S. Muhammad, "Honesty based democratic scheme to improve community cooperation for Internet of Things based vehicular delay tolerant networks," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, p. e4191, 2021.

[12] L. Feng, A. Ali, M. Iqbal et al., "Dynamic wireless information and power transfer scheme for nano-empowered vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4088–4099, 2020.

[13] A. Ghani, A. Badshah, S. Jan, A. A. Alshdadi, and A. Daud, "Issues and challenges in cloud storage architecture: a survey," vol. 12, 2020, https://arXiv.org/abs/2004.06809.

[14] T. Frikha, F. Chaabane, N. Aouinti, O. Cheikhrouhou, N. Ben Amor, and A. Kerrouche, "Implementation of blockchain consensus algorithm on embedded architecture," *Security and Communication Networks*, vol. 20, no. 21, p. 3268, 2021.

[15] M. Azrour, J. Mabrouki, and R. Chaganti, "New efficient and secured authentication protocol for Remote healthcare systems in cloud-IoT," *Security and Communication Networks*, vol. 2021, Article ID 5546334, 12 pages, 2021.

[16] M. Nazari and A. Maneshi, "Chaotic reversible watermarking method based on IWT with tamper detection for transferring electronic health record," *Security and Communication Networks*, vol. 2021, Article ID 5514944, 15 pages, 2021.

[17] T. Y. Wu, Z. Lee, L. Yang, and C. M. Chen, "A provably secure authentication and key exchange protocol in vehicular ad hoc networks," *Security and Communication Networks*, vol. 2021, Article ID 9944460, 17 pages, 2021.

[18] J. Mahmood, Z. Duan, Y. Yang, Q. Wang, J. Nebhen, and M. N. M. Bhutta, "Security in vehicular ad hoc networks: challenges and Countermeasures," *Security and Communication Networks*, vol. 2021, Article ID 9997771, 20 pages, 2021.

[19] C. Johansen, A. Mujaj, H. Arshad, and J. Noll, "The snowden phone: a comparative survey of secure instant messaging mobile applications," 2018, https://arxiv.org/abs/1807.07952.

[20] A. Guezzaz, S. Benkirane, M. Azrour, and S. Khurram, "A reliable network intrusion detection approach using decision tree with enhanced data quality," *Security and Communication Networks*, vol. 2021, Article ID 1230593, 8 pages, 2021.

[21] M. Azrour, J. Mabrouki, A. Guezzaz, and A. Kanwal, "Internet of things security: challenges and key issues," *Security and Communication Networks*, vol. 2021, Article ID 5533843, 11 pages, 2021.

[22] A. K. Biswal, D. Singh, B. K. Pattanayak, D. Samanta, S. A. Chaudhry, and A. Irshad, "Adaptive fault-tolerant system and optimal power allocation for smart vehicles in smart cities using controller area network," *Security and Communication Networks*, vol. 2021, Article ID 2147958, 13 pages, 2021.

[23] S. A. Chaudhry, "Designing an efficient and secure message exchange protocol for Internet of vehicles," *Security and Communication Networks*, vol. 2021, Article ID 5554318, 9 pages, 2021.

[24] C. M. Chen and S. Liu, "Improved secure and lightweight Authentication scheme for next-generation IoT infrastructure," *Security and Communication Networks*, vol. 2021, Article ID 6537678, 13 pages, 2021.

[25] K. Taehyoung, J. Cheongmin, and H. Manpyo, "AVoD: Advanced Verify-on-Demand for efficient authentication against DoS attacks in V2X communication," *Security and Communication Networks*, vol. 2021, Article ID 2890132, 9 pages, 2021.

[26] I. Javed, X. Tang, K. Shaukat et al., "V2X-Based mobile localization in 3D wireless sensor network," *Security and Communication Networks*, vol. 2021, Article ID 6677896, 13 pages, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 2890132, 9 pages
https://doi.org/10.1155/2021/2890132

Research Article

AVoD: Advanced Verify-on-Demand for Efficient Authentication against DoS Attacks in V2X Communication

Taehyoung Ko,[1] **Cheongmin Ji**,[1] **and Manpyo Hong**[2]

[1]*Department of Computer Engineering, Ajou University, Suwon 16499, Republic of Korea*
[2]*Department of Cyber Security, Ajou University, Suwon 16499, Republic of Korea*

Correspondence should be addressed to Manpyo Hong; mphong@ajou.ac.kr

Received 21 June 2021; Revised 17 October 2021; Accepted 10 November 2021; Published 1 December 2021

Academic Editor: Azeem Irshad

Owing to the development of information and communication technology (ICT), autonomous cooperative vehicles are being developed. Autonomous cooperative driving combines vehicle-to-everything (V2X) communication technology in existing autonomous driving and provides safe driving by sharing information between communication entities. However, security factors should be considered during communication. Security Credential Management System (SCMS) has been proposed as one of these elements, but it is vulnerable to denial-of-service (DoS) attacks due to message authentication costs. In congested situations, the number of messages exchanged between vehicles becomes very large. However, the performance of the on-board unit (OBU) is not sufficient to handle huge number of messages, which can lead to a DoS attack. Therefore, a technique to prevent DoS attacks on autonomous cooperative driving vehicles using SCMS has been proposed in this paper. The proposed technique reduces authentication costs by classifying similar messages into multiple categories and authenticating only the first message represented in the group for a unit time. The effectiveness of this technique has been demonstrated by comparing the time it takes to verify huge number of message signatures in each method.

1. Introduction

With the recent development of information technology, autonomous driving technology has been actively researched in the automotive field. Research costs in the automotive field are increasing every year, and the proportion of software and computer services is also increasing. In addition, patents for self-driving cars have been increasing since 2015 [1,2], and IT companies such as Google and Apple, as well as automobile manufacturers, such as Mercedes and BMW, have been developing self-driving cars [3]. The advantage of autonomous driving is that there are fewer errors caused by humans because of minimal human intervention, compared to manual driving. In particular, according to a survey [4] conducted by the National Highway Traffic Safety Administration (NHTSA), 94% of car accidents were caused by drivers. Therefore, it is possible to perform safe driving using autonomous driving technology in which the driver is a system and not a human being, when compared to manual driving.

However, because autonomous driving alone is not sufficient to perform safe driving, autonomous cooperative driving using vehicle-to-everything (V2X) communication technology is required. Autonomous cooperative driving is not judged by only one vehicle but communicates with nearby vehicles or roadside fixed V2X communication units (road side unit (RSU)) to make judgments for safer driving. However, security factors must be considered, because V2X communication is used to communicate with other elements. In fact, white hackers infiltrated vehicles from the outside and showed examples of attacks that manipulated various functions in the vehicle [5,6]. In this regard, measures for message integrity verification, authentication, and personal protection should be implemented. As a result, security system and standards have been established.

The U.S. Department of Transportation (USDOT) is currently developing, applying, and implementing a V2X security authentication system called the Security Credential Management System (SCMS) [7] to enhance the security of autonomous cooperative driving. The SCMS is a PKI-based

message authentication system. Each participant performing V2X communication using SCMS can trust shared information through authentication. There are three design objectives for the SCMS. The first is to ensure accuracy and reliability of the information exchanged, the second is to protect the privacy of the driver, and the third is to support the identification of devices revoked through malfunctioning device identification and certificate revocation list (CRL) distribution. Therefore, the SCMS provides a security infrastructure for issuing and managing security certificates. Each entity that wants to perform V2X communication can register with the SCMS, obtain a security certificate from the certification authority, and authenticate the message to prove that it is a trusted entity. Certificates used in SCMS are largely divided into certificates for the on-board equipment (OBE) and certificates for the RSU. Certificates for OBE include OBE Enrollment Certificate, Pseudonym Certificate, and Identification Certificate. The OBE can request for another certificate using Enrollment Certificate. Pseudonym Certificate is mainly used for short-term, basic safety message (BSM) authentication, and misbehavior reporting. Multiple Pseudonym Certificates obtained from the Pseudonym Certificate Authority are changed after expiration validity period. The OBE uses Identification Certificates to identify special and public vehicles. Certificates used by the RSU include the RSU Enrollment Certificate and Application Certification. An Enrollment Certificate is used by the RSU to receive application certification. Application certification is used by the RSU to provide secure transportation services, such as signing over air messages. During V2X communication, each entity can report misbehaving *o* malfunctioning. CRLs are created through misbehavior report and added to the blacklist inside SCMS. Each entity can block messages from revoked entities using CRLs.

Society of Automotive Engineers (SAE) has created a standard that defines the Dedicated Short Range Communications (DSRC) message set and On-Board System Requirements for V2V Safety Communications. The message used by each entity to exchange information with each other uses the message defined in document SAE J2735 [8]. SAE J2735 is Dedicated Short Range Communications (DSRC) message set. SAE J2735 includes a set of DSRC messages, a data frame, and data elements that make up each message. Some of these message sets include the basic safety message (BSM), common safety request (CSR), and emergency vehicle alert (EVA). BSM is a message that contains basic information about a vehicle, including its current location, speed, gear information, and braking information. BSM broadcasts 10 messages per second to surrounding vehicles. CSR can be unicast as a message asking for additional information between vehicles exchanging BSMs. Additional information requested by CSR includes light, wiper, brakeStatus, brakePressure, and weather data measured by sensors. The EVA message broadcasts a warning message that an emergency vehicle is operating nearby and that the vehicle's drivers need attention. In addition to the above-mentioned messages, other messages defined in J2735 are used to communicate with vehicles to exchange road and driving information for safe driving.

SAE J2945/1 [9] is a standard document that contains the system requirements of the on-board unit (OBU) for secure V2V communication proposed by SAE. The standards specify the standard profiles, functional requirements, and performance requirements. The standard profiles contain 802.11 related requirements for basic communication and IEEE 1609.2 [10] related to security. In particular, it is required to use Secure Hash Algorithm (SHA) 256 as the hashing algorithm and Elliptic Curve Digital Signature Algorithm (ECDSA) 256 with NIST p256 as the signature. Symmetric encryption requires support for AES-128. In addition, there are requirements for recording the position of the vehicle and the route it travels.

Despite these security systems, V2X communication has a big security threat. That is a denial-of-service (DoS) attack. DoS attacks on SCMSs can cause delays in traffic flow as well as car crashes. Therefore, the goal of this paper is to propose advanced verify-on-demand (AVoD), a technique to prevent DoS attacks on autonomous cooperative vehicles using SCMS, and to validate its effectiveness in preventing DoS attacks.

The remainder of this paper is organized as follows. Section 2 discusses related works on DoS attack in V2X environment and its countermeasure. Section 3 describes the security analysis in autonomous cooperative driving. Section 4 explores the AVoD proposed in this study. Section 5 examines the actual implementation of AVoD, and Section 6 presents the conclusions.

2. Related Works

In the V2X environment, a safe driving environment is provided by exchanging information between each entity. Due to these characteristics, an attack that reduces availability can have a fatal impact on the whole network. This section discusses research on attacks that compromise availability and studies on countermeasures for such attacks.

Trkulja et al. [11] introduced a set of denial-of-service attacks on C-V2X networks operating in Mode 4. The attack presented in this research is caused by adversarial resource block selection. This attack is a very sophisticated and efficient attack. In this study, each attack is analyzed by setting three types of enemies. With a fixed number of attackers, this study shows that smart and cooperative attacks can have a significant impact on network performance when the vehicle density is low, whereas when the vehicle density is high, the unconscious attack is more effective than sophisticated attack.

Another type of attack that reduces availability is a jamming attack. Safety applications in vehicle networks include real-time information contained in periodically exchanged messages called beacons. A jamming attack that interferes with beacon transmission is studied in the work of Benslimane et al. [12]. This study investigates the effect of jamming attack on beacon broadcast and proposes a real-time MAC (Media Access Control) based detection method for jamming attack. This method works well when the number of vehicles constituting the platoon is fixed, while it

does not work well when the number of vehicles belonging to the platoon changes frequently.

Studies have been conducted on the use of lightweight protocols in the authentication process to prevent attacks that compromise availability [13–15].

In 2020, Vasudev et al. [13] proposed lightweight mutual authentication scheme for V2V Communication in Internet of Vehicles. In their work, a scheme with a lower computation cost was proposed compared to the efficient mutual authentication schemes that were previously proposed [16–18]. In addition, those authors used SHA-3 with 256 bit, which is relatively robust in collision attacks compared to the study using SHA-1 [19]. This scheme performs 17 hash functions including the registration phase to perform mutual authentication. This method consumes lower computation cost compared to the existing methods, but still has vulnerabilities to collision attacks. In particular, the security strength is lower than that of ECDSA, which provides strong authentication. In addition, it cannot be applied to environments using SCMS. S.A.A. Hakeem et al. [14] proposed lightweight message authentication and privacy preservation protocol for V2X communications. This scheme uses hash chain of secret keys for a Message Authentication Code (MAC). It reduces computation overhead and communication overhead compared to using standard security protocols. The advantages of this technology are attractive, but useless in an environment where standards are enforced. In 2018, S. Taha et al. [15] proposed lightweight group authentication scheme for achieving low latency with high mobility in vehicular networks. For this reason, the authors clustered vehicles and assigned each vehicle a role within the cluster. Their scheme aims to create a shared group key within the cluster as well as mutual authentication between vehicles in the cluster. However, their scheme is a group authentication method and cannot be applied to the V2X network using SCMS targeted in this paper.

Another countermeasure is to improve the authentication speed using hardware. Using General-Purpose computing on Graphics Processing Units (GPGPU) to accelerate the hardware, ECDSA authentication speed improvement was achieved [20]. However, in OBU or RSU that performs ECDSA, the performance of the GPU is low, so it is difficult to exert a great effect. Another scheme [21] is to use parallel programming. This method is available because most embedded CPUs have multicores. Those authors performed ECDSA signature verification in parallel using 16 threads. As a result, the processing speed was four times faster than that of a single thread.

The last countermeasure is to change the authentication policy. SAE proposed verify-on-demand (VoD) [22] to increase the processing speed of the encryption module, address the security vulnerability presented above, and ultimately prevent DoS attacks. VoD changes the authentication policy instead of using the authentication protocol specified in IEEE1609.2 by using this; the number of messages to be authenticated is reduced. VoD will be covered in more detail in Section 3.

3. Security Analysis

There are two security vulnerabilities in autonomous cooperative driving using the SCMSs. The first is the processing speed of the encryption module. In the Notice of Proposed Rulemaking for Federal Motor Vehicle Safety Standards [23] issued by the National Highway Traffic Safety Administration (NHTSA), it is stated that DSRC equipment must perform validation of at least 5500 BSMs per second. Message processing proceeds in the order of interpreting the content of the message after its authentication. Therefore, in crowded situations, the OBU must be able to perform more than 5500 BSM signature verifications per second. However, ECDSA cannot process 5500 BSMs per second because it requires more time to authenticate than the existing RSA [24–27]. According to the research that measured the authentication time in the actual OBU [14], processing speed of OBU is only 35 verifications per second in the case of the software module. In the case of the hardware module, processing speed of OBU is only 163 verifications per second. Eventually, owing to limitations in the performance of hardware and cryptographic modules, many BSM authentications arising from congestion situations are not performed well. The second vulnerability is DoS attack. This vulnerability arises from the aforementioned vulnerability, which prevents the normal behavior of OBUs by sending more BSM messages than they can handle. These vulnerabilities can lead to traffic accidents or congestion during autonomous cooperative driving using SCMSs. Furthermore, if authentication is omitted to prevent DoS attacks, the risk of forgery or tampering attacks is encountered. In this section, situations in which conflicts occur in autonomous cooperative driving environments and security threats that can arise in each situation are discussed.

3.1. Attacker Model. In this study, the attacker launches a DoS attack on a vehicle that performs autonomous cooperative driving. The goal of a DoS attack is to disrupt the road and ultimately paralyze it. Vehicles targeted for attack are vehicles located within 1 km radius of the attacker, which is the propagation range of basic DSRC/WAVE messages. The attacker must be able to generate a large number of BSMs. The attacker can send BSMs more than 10 times per second using the modified program. The attacker has multiple certificates normally issued from the SCMS. It is assumed that the attacker remotely penetrates vehicle of the normal user through backdoor for obtain a certificate. Using these certificates, the attacker sends BSMs as impersonating normal user. The victim reports to Registration Authority (RA; RA manages CRLs) to add a certificate that signed the attack BSMs to CRLs. Since then, when BSMs signed with a certificate listed in CRLs are received, those are dropped. For this reason, attackers use a different certificate for each attack to effectively perform attacks. It is also assumed that more than one attacker OBU is used to transmit 5500 BSMs per second.

3.2. Attack Situation. In SAE J2945/1 [9], seven threatening crash-imminent scenarios were selected considering the frequency, cost, and functional years lost. It is designed to prevent collisions by operating safety applications for each scenario. Crash-imminent scenarios are shown in Table 1. All of them, except blind spot warning (BSW) and lane change warning (LCW), operate using sensors and BSM. Therefore, in order to avoid collisions, it is important to receive the BSM without interruption. The attacker uses this point to perform an attack. As shown in Figure 1, the attacker sends a large number of BSMs to the target vehicle. Vehicles within the attacker's DSRC/WAVE transmission range become the target vehicle. When the attack starts, the target cars cannot receive BSMs normally. Because BSMs cannot be received normally, among the scenarios shown in Table 1, other scenarios except 'Vehicle(s) Changing Lanes-Same Direction' have a high probability of collision. For example, when a lead vehicle is stopped, it is necessary to detect the sudden halt of the vehicle in front, using FCW. However, it does not receive the BSM due to a DoS attack, which causes a crash owing to the delay in understanding the situation that occurred earlier.

3.3. Countermeasures. As discussed in Section 2, two countermeasures are considered to prevent this security threat. The first is using hardware. A simple method is to use a high-performance processor in OBU for authentication. Other methods are methods of using a hardware module. Authentication speed can be improved using GPGPU [20] or Hardware Security Module. However, this method is not covered in this paper because this method requires additional hardware.

The second countermeasure is the software method. Software methods include changing authentication policies and using a lightweight authentication algorithm. A lightweight authentication algorithm can be used to perform authentication quickly and securely. However, since this paper targets the V2X network using SCMS and IEEE 1609.2, this method is not discussed. Finally, the method presented in this paper changes the authentication policy. This paper uses the ECDSA required by the standard and proposes a more efficient authentication policy.

The method of performing message authentication in SCMS is the verification and then process (VATP). As shown in Figure 2, the OBU first authenticates the messages received from the antenna, checks the threat level, and then notifies the driver if they are determined to be a threat. This method performs authentication on every message; thus, the slower the encryption module processes, the slower the driver is informed of the threat. In addition, a DoS attack that consumes all the computing power of OBUs for authentication, thereby preventing it from performing other functions, can occur.

The basic flow of the VoD is shown in Figure 3. Unlike VATP, which performs message authentication first, VoD first checks the threat level of the message after receiving the message. Subsequently, only messages judged to be threats are authenticated. This message is called "threat message (threat BSM)." In conclusion, if it is not a threat message, it is ignored. It means that authentication is not performed. This approach prevents DoS attacks by reducing messages to be processed, compared to conventional methods, because only threat messages are authenticated.

Table 1: Crash-imminent scenarios and its safety applications [9].

Crash scenario	Safety applications
Lead vehicle stopped	FCW
Lead vehicle decelerating	EEBL FCW
Control loss without prior vehicle action	CLW
Vehicle turning at nonsignalized junctions	IMA LTA
Straight crossing paths at nonsignalized junctions	IMA
Vehicle) changing lanes—same direction	BSW/LCW
Left turn across path—opposite direction	LTA

4. AVoD (Advanced Verify-on-Demand)

4.1. AVoD Overview. The VoD discussed above is a technique that prevents DoS attacks by reducing the number of messages to be processed. However, this technique is still vulnerable to DoS attacks. VoD only authenticates messages deemed to be threats, and if the number of such messages exceeds the processor's throughput, the DoS attack will still be valid. Therefore, in this section, advanced verify-on-demand (AVoD) is proposed. The basic flow of the AVoD is shown in Figure 4. AVoD is a method that performs authentication smoothly, even when many BSMs that are considered to be threats are received. AVoD prevents DoS attacks by classifying messages deemed as threats and authenticating only the first message represented in the group for a unit time.

4.2. AVoD Components. The AVoD module consists of a threat table and a threat classifier. The threat classifier categorizes messages based on the criteria for current driving conditions. Classified messages are stored in threat table and sent to signature verification module. If there is a message already classified as the same type, the next message is ignored without signature verification. Threat table is a table that stores messages classified by the threat classifier. Table 2 shows an example of threat table in which packets are recorded. It records safety applications, vehicle locations, vehicle directions, and number of packets.

4.3. AVoD Algorithm. A Threat classifier classifies each received BSM according to relative location with other vehicles and stores it in threat table. In this way, when more than 5500 threat BSMs occur per second, these BSMs can be classified into several types. After that, only the first message of each type performs signature verification and subsequent messages are ignored. The reason is that the threat to the ignored message is already generating an alert.

There are three criteria for classifying BSMs in a threat classifier. The first is the safety application to be used. The message is classified by determining the safety application

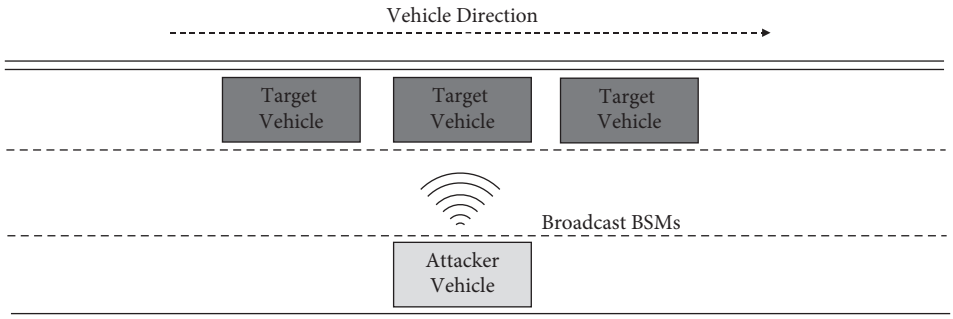


Figure 1: Attack situation.

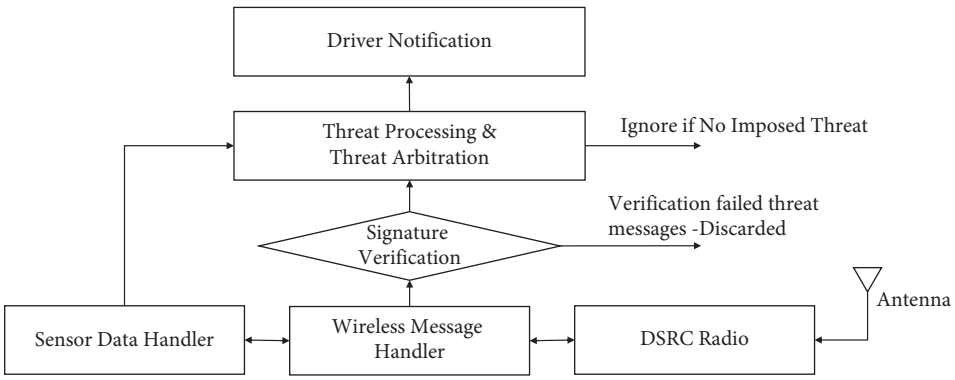


Figure 2: Verify-and-then-process flow [22].

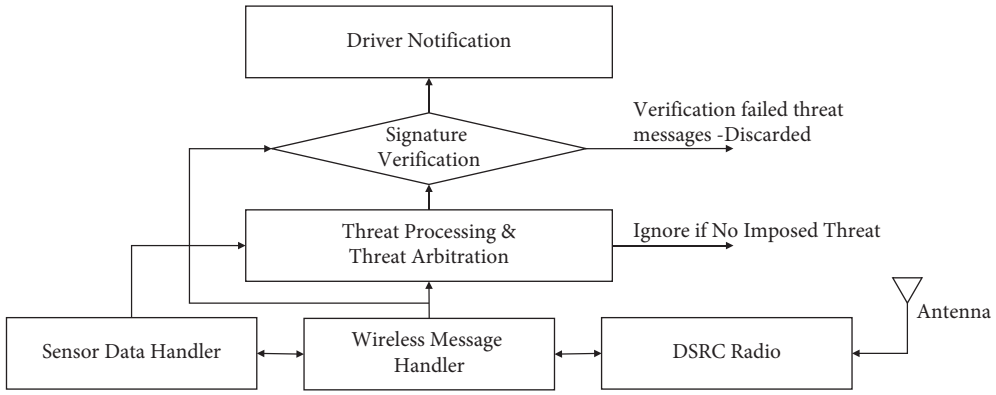


Figure 3: Verify-on-demand flow [22].

that is used to alert the driver. The next step is the location of the Remote Vehicle (RV). RV is the sender of BSMs. It is not a real sender (attacker). It is the sender written in BSMs generated by attacker. Eight spaces are defined based on the Host Vehicle (HV) to represent the location of the RV. HV is receiver of BSMs. As shown in Figure 5, it is possible to classify the location of the RV from which the message is sent, by separating it into eight zones, based on the direction in which the HV proceeds. The third factor is the direction of travel of the vehicle. The traveling direction of the vehicle can be divided into four types: the same direction as the HV, the opposite direction, the left direction at a right angle, and the right direction at a right angle. By combining these three criteria, the BSM transmitted for a unit of time is classified in real time. When AVoD module classifies BSMs, it checks to see if there are BSMs of the same type in threat table. If the same type of BSM exists in threat table, column of "number of packets" is increased by one and that BSM is ignored. On the contrary, if the same type of BSM does not exist in threat table, that BSM is delivered to the signature verification module.

4.4. Attacks on AVoD. This section describes how AVoD works using several situations. There is more than one vehicle that can be classified into the same category. In that case, the situation is divided into two. First situation is when the same type of BSM is received. Corresponding

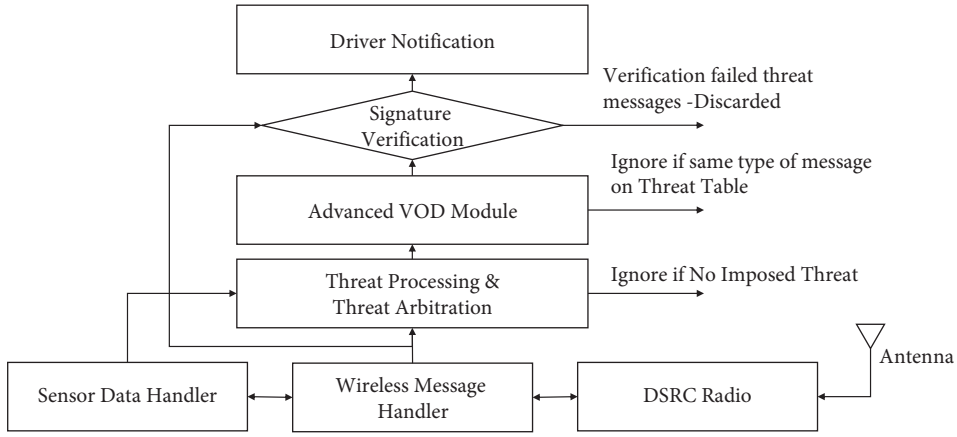


Figure 4: Advanced verify-on-demand flow.

Table 2: Threat table.

No	Safety application	Vehicle location	Vehicle direction	Number of packets
1	FCW	Center forward	Same	2
2	LTA	Left forward	Opposite	1
3	FCW	Left forward	Opposite	4
...	...	...	...	...

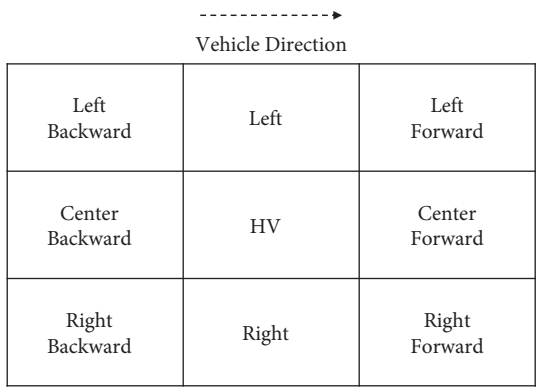


Figure 5: Vehicle location.

alert is already occurring, so signature verification is not performed. And this BSM is ignored. Second is when the different type of BSM is received. AVoD module adds it to threat table and passes it to the signature verification module.

It is assumed that the attacker transmits fake messages using fake certificates after getting verified by the RV. The AVoD module analyzes the BSMs regardless of the certificate. Even if it is a fake certificate, check whether the BSM is on the threat table. Theoretically, the thread table can store 160 rows (5 safety applications, 8 vehicle locations, and 4 vehicle directions). Eventually, OBU only needs to perform 160 signature verifications during the unit time, even if it receives BSMs more than 5500 per second. For this reason, AVoD only focuses on BSMs.

5. Experiment

The AVoD proposed in this paper is a proposed technique to prevent DoS attacks. In this section, an experiment is conducted to measure AVoD performance.

5.1. Experiment Overview. The experiment is conducted by measuring the time taken to process 10,000 BSMs per second on a personal computer (PC). 10 test message sets are used, in which the proportion of threat BSM among 10,000 messages increased by 10% from 10% to 100%. These message sets are named test case (TC) 1 to 10. Using the message sets from TC1 to TC10, the processing time in VATP, VoD, and AVoD is measured, and the average value is obtained by repeating this ten times in total. This is used to examine the results of a DoS attack with many threat BSMs. This is also a weakness of the existing VoD, and the effectiveness of the AVoD in the attack is examined.

PC specifications of experimental environment are shown in Table 3.

The BOGOMIPS measurement method used in this experiment is a certain program that consists of sleep function, time calculation function, and loop. It is similar implementation of BogoMips program in Linux kernel. It is used to compare the performance difference with the PC, by measuring BOGOMIPS through the execution of the same code in the OBU.

5.2. Result of the Experiment. Figure 6 shows the experimental results. Experimental results from TC1 through TC10 show that VATP takes approximately 660 ms, all of

Table 3: PC specification of experimental environment.

Specification	
OS	Linux Kernel version 4.15.0, Ubuntu 18.04.5 LTS
CPU	Intel i7-8550U @ 1.80 GHz
RAM	32 GB
BOGOMIPS	1465.83

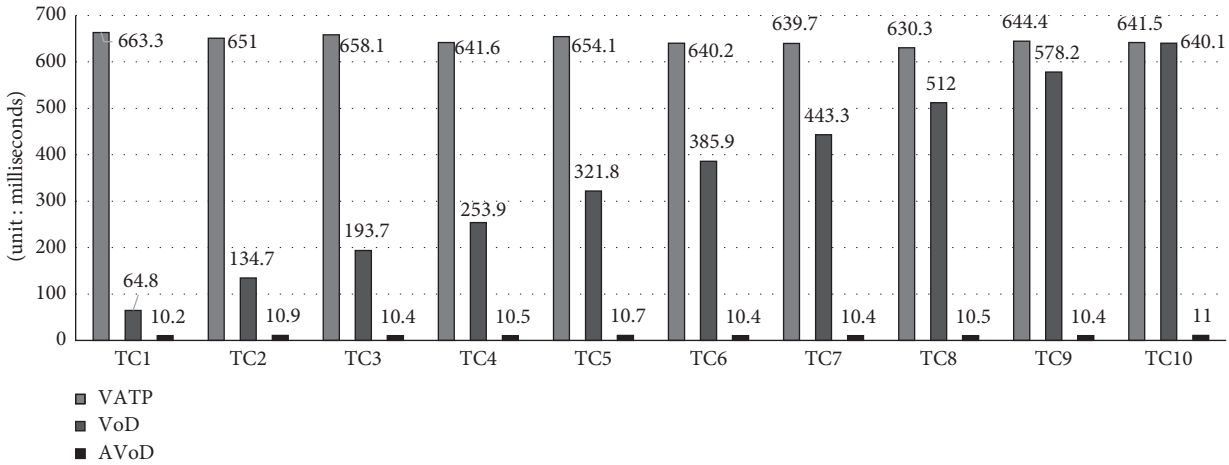


Figure 6: Result of experiment on PC.

Table 4: OBU specification of experimental environment.

Specification	
OS	Linux Kernel version 3.10.17, Ubuntu 14.04 LTS
CPU	NXP i.MX 6DualLite,800 MHz
RAM	1 GB SDRAM
BOGOMIPS	196.66 (7.45 times slow than PC)

which are similar. In the case of VoD, the execution time linearly increases as the ratio of messages containing risk increases. Finally, in the case of AVoD, it is observed that minimal execution time of 10 to 11 milliseconds is required because all attack messages are classified into several types. Compared to VoD, AVoD is processed approximately 6.35 times faster for TC1. In the case of TC10, the processing speed is up to 58 times faster. Based on this, it is established that the performance of AVoD is excellent, when the ratio of messages containing collisions, also mentioned as a weakness of the existing VoD, increased. In addition, the throughput rates in real OBUs are approximated for comparison, using BOGOMIPS figures measured in OBUs.

OBU specifications of experimental environment are shown in Table 4.

OBU's BOGOMIPS is 7.45 times slower than that of PCs, so the outcomes reflecting the corresponding values in the experimental results are shown in Figure 7. The TC2 results show that VoD takes more than one second to perform authentication. As there are 2000 threat BSMs in TC2, it is determined that the OBU can process approximately 2000 threat BSMs per second. Therefore, a DoS attack occurs even when using VoD in case the OBU receives more than 2000 threat BSMs. In the case of AVoD, TC10 takes approximately 82 milliseconds, which can be used to prevent DoS attacks by performing authentication in a short period of time, even if OBU receives more than 10,000 threat BSMs per second.

5.3. Comparing with Related Works. In this section, we compare with studies to speed up ECDSA verification discussed in Section 2. A study [20] that improved the ECDSA verification speed using GPGPU was conducted using ODROID-XU4. ODROID-XU4 have Cortex-A15 Quad Core 2.0 Ghz, Cortex-A7 Quad Core 1.4 GHz, and Mali-T628 MP6 (256core). In their study, the best performance using ODROID-XU4 was 15.4 signature verifications per second. Device with better hardware performance showed lower ECDSA verification performance than the proposed scheme. Lee et al. [21] use parallel

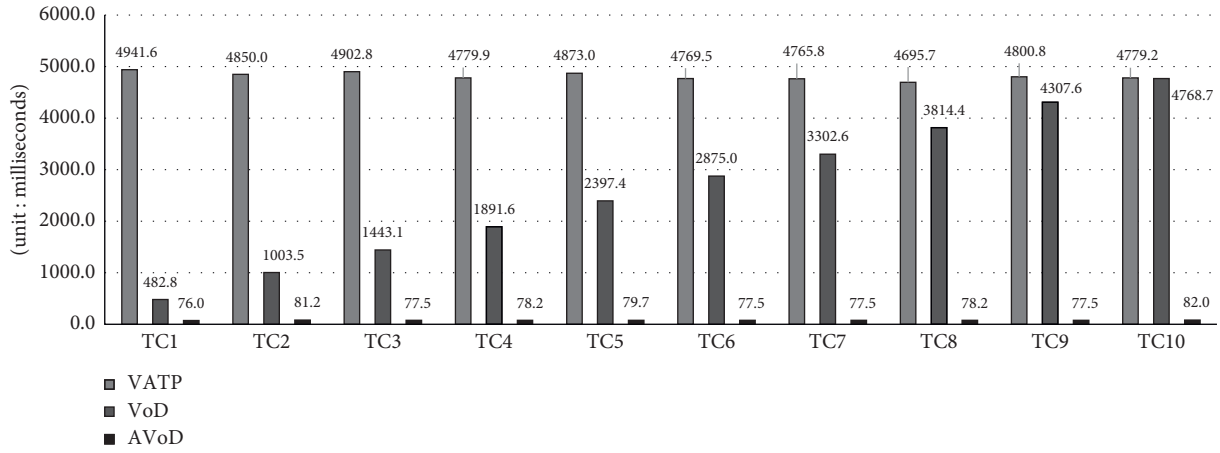


Figure 7: Result of experiment (2), approximated on OBU.

Table 5: Result of experiment, compared with S. Lee et al. [21].

Scheme	Number of threads	Verification time per one signature (unit: milliseconds)	Number of verifications per second	Time it takes to process 10,000 signatures (unit: milliseconds)
Lee et al. [21]	1	1.616	619.70	16160
	2	0.810	1233.53	8100
	4	0.417	2395.69	4170
	8	0.413	2417.32	4130
	16	0.409	2439.69	4090
Proposed scheme	1	—	—	82

programming to speed up signature verification. Those authors experimented using NXP i.MX 6 same as CPU used in this paper. Using the time to perform 10,000 ECDSA signature verifications, the verification time per unit and the number of verifications per second were calculated. Table 5 shows the experimental results. Compared indirectly to other schemes, the proposed scheme is more effective on verification BSMs.

6. Conclusion and Future Works

In this study, a technique to prevent DoS attacks in autonomous cooperative driving using SCMSs is proposed. The proposed method classifies threat messages to authenticate only the first messages received within the same group. Compared to VoD, which is a technique for preventing DoS attacks, this method has demonstrated the ability to process messages at a speed of 6.3 times to 58 times faster, depending on the situation. This proves to be a technique that effectively prevents DoS attacks that transmit a large number of messages. However, in the experiment, it is difficult to announce that an accurate result was obtained owing to the performance difference in the processing speed between the PC and the OBU when performing it on the PC rather than in the actual OBU. In future research, the challenge is to experiment with the actual OBU and determine its efficiency.

Data Availability

The cryptographic library used in the experiment for measuring the performance was OpenSSL version 3.0.0, and the curve used for ECDSA is secp256r1. As mentioned in the previous section, the experimental computer has a 1.80 GHz Quad Core CPU and a 32 GB RAM.

Conflicts of Interest

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

Acknowledgments

This research was supported by the Korea Ministry of Land, Infrastructure and Transport. It was also supported by the Korea Agency for Infrastructure Technology Advancement (Project no. 21PQOW-B152473-03).

Supplementary Materials

The supplementary material contains BOGOMIPS measurement source code. This code is used in the Result of the Experiment section. It is used to compare the performance difference with the PC, by measuring BOGOMIPS through

the execution of the same code in the OBU. (*Supplementary Materials*)

References

[1] European Automobile Manufacturers Association, "The automobile industry pocket guide 2020–2021," 2020, https://www.acea.be/uploads/publications/ACEA_Pocket_Guide_2020-2021.pdf.

[2] European Automobile Manufacturers Association, "The automobile industry pocket guide 2019–2020," 2019, https://www.acea.be/uploads/publications/ACEA_Pocket_Guide_2019-2020.pdf.

[3] C.-Y. Chan, "Advancements, prospects, and impacts of automated driving systems," *International Journal of Transportation Science and Technology*, vol. 6, no. 3, pp. 208–216, 2017.

[4] S. Singh, "Critical reasons for crashes investigated in the national motor Vehicle crash causation survey," *Traffic Safety Facts Crash Stats*, vol. 812 115, 2015.

[5] "Hackers remotely kill a jeep on the Highway - with me in it," 2015, https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

[6] C. Miller and C. Valasek, *Remote Exploitation of an Unaltered Passenger Vehicle*, Black Hat, USA, 2015.

[7] B. Brecht, D. Therriault, A. Weimerskirch et al., "A security credential management system for V2X communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3850–3871, 2018.

[8] Society of Automotive Engineers International, *Dedicated Short Range Communications (DSRC) Message Set Dictionary: J2735*, SAE International, Warrendale, PA, US, 2018.

[9] Society of Automotive Engineers International, *On-Board System Requirements for V2V Safety Communications: J2945/1*, SAE International, Warrendale, PA, US, 2016.

[10] IEEE, *Standard for Wireless Access in Vehicular Environments-Security Services for Applications and Management Messages*, IEEE, Piscataway, NJ, USA, 2016.

[11] N. Trkulja, D. Starobinski, and R. A. Berry, "Denial-of-Service attacks on C-V2X networks," 2020, http://arXiv:2010.13725.

[12] A. Benslimane and H. Nguyen-Minh, "Jamming attack model and detection method for beacons under multichannel operation in Vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6475–6488, 2016.

[13] H. Vasudev, V. Deshpande, D. Das, and S. K. Das, "A lightweight mutual authentication protocol for V2V communication in Internet of Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6709–6717, 2020.

[14] S. A. A. Hakeem, M. A. A. El-Gawad, and H. Kim, "Comparative experiments of V2X security protocol based on hash chain cryptography," *Sensors*, vol. 20, no. 19, 2020.

[15] S. Taha and X. S. Shen, "Lightweight group Authentication with dynamic Vehicle-clustering for 5G-based V2X communications," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Abu Dhabi, UAE, December 2018.

[16] H. Mun, K. Han, Y. S. Lee, C. Y. Yeun, and H. H. Choi, "Enhanced secure anonymous authentication scheme for roaming service in global mobility networks," *Mathematical and Computer Modelling*, vol. 55, no. 1–2, pp. 214–222, 2012.

[17] C. C. Wu, W. B. Lee, and W. J. Tsaur, "A secure authentication scheme with anonymity for wireless communications," *IEEE Communications Letters*, vol. 12, no. 10, pp. 722-723, 2008.

[18] D. Zhao, H. Peng, L. Li, and Y. Yang, "A secure and effective anonymous authentication scheme for roaming service in global mobility networks," *Wireless Personal Communications*, vol. 78, no. 1, pp. 247–269, 2014.

[19] M. Prerna, A. Ruhul, and G. P. Biswas, "Design of authentication protocol for wireless sensor network-based smart vehicular system," *Vehicular Communications*, vol. 9, pp. 64–71, 2017.

[20] S. Lee, H. Seo, B. Chunng, J. Choi, H. Kwon, and H. Yoon, "OpenCL based implementation of ECDSA signature Verification for V2X communication," in *Proceedings of the International Conference on Multimedia and Ubiquitous Engineering*, pp. 711–716, Salerno, Italy, November 2018.

[21] S. Lee, H. Seo, B. Chunng, and H. Kwon, "Study on parallel processing of ECDSA Verification for V2X communication," in *Proceedings of the Korea Information Processing Society Conference*, pp. 216-217, Busan, Republic of Korea, 2018.

[22] Society of Automotive Engineers International, *Verify on Demand*, 2017.

[23] National Highway Traffic Safety Administration, *Notice of Proposed Rulemaking: Federal Motor Vehicle Safety Standards*, V2V Communications, National Highway Traffic Safety Administration(NHTSA), Washington, DC, USA, 2017.

[24] T. Oder, T. Pöppelmann, and T. Güneysu, "Beyond ECDSA and RSA: lattice-based digital signatures on constrained devices," in *Proceedings of the 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, San Francisco, CA, USA, June 2014.

[25] S. A. Manuel, F. L. Paula, and M. F. C. Tiago, "A practical evaluation on RSA and ECC-based cipher suites for IoT high-security energy-efficient fog and mist computing devices," *Sensors*, vol. 18, no. 11, 2018.

[26] C. Zhang, X. Lin, R. Lu, and P.-H. Ho, "RAISE: an efficient RSU-aided message authentication scheme in Vehicular communication networks," in *Proceedings of the 2008 IEEE International Conference on Communications*, pp. 1451–1457, Beijing, China, May 2008.

[27] S. Jha, C. Yavvari, and D. Wijesekera, "Pseudonym certificate Validations under heavy Vehicular traffic loads," in *Proceedings of the IEEE Vehicular Networking Conference*, pp. 1–7, Taipei, Taiwan, December 2018.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 2147958, 13 pages
https://doi.org/10.1155/2021/2147958

Research Article

Adaptive Fault-Tolerant System and Optimal Power Allocation for Smart Vehicles in Smart Cities Using Controller Area Network

Anil Kumar Biswal,[1] **Debabrata Singh**,[1] **Binod Kumar Pattanayak**,[1] **Debabrata Samanta**,[2] **Shehzad Ashraf Chaudhry**,[3] **and Azeem Irshad**[4]

[1]*Department of CSE, ITER, SOA Deemed to be University, Bhubaneswar, Odisha, India*
[2]*Department of Computer Science, CHRIST University, Bangalore, Karnataka, India*
[3]*Department of Computer Engineering, Istanbul Gelisim University, Istanbul, Turkey*
[4]*Computer Science Software Engineering, International Islamic University, Islamabad, Pakistan*

Correspondence should be addressed to Azeem Irshad; azeem.phdcs66@iiu.edu.pk

Received 7 June 2021; Revised 8 September 2021; Accepted 28 September 2021; Published 13 October 2021

Academic Editor: Emanuele Maiorana

Nowadays, the power consumption and dependable repeated data collection are causing the main issue for fault or collision in controller area network (CAN), which has a great impact for designing autonomous vehicle in smart cities. Whenever a smart vehicle is designed with several sensor nodes, Internet of Things (IoT) modules are linked through CAN for reliable transmission of a message for avoiding collision, but it is failed in communication due to delay and collision in communication of message frame from a source node to the destination. Generally, the emerging role of IoT and vehicles has undoubtedly brought a new path for tomorrow's cities. The method proposed in this paper is used to gain fault-tolerant capability through Probabilistic Automatic Repeat Request (PARQ) and also Probabilistic Automatic Repeat Request (PARQ) with Fault Impact (PARQ-FI), in addition to providing optimal power allocation in CAN sensor nodes for enhancing the performance of the process and also significantly acting a role for making future smart cities. Several message frames are needed to be retransmitted on PARQ and fault impact (PARQ-FI) calculates the message with a response probability of each node.

1. Introduction

In recent days, the avoidance of the traffic congestion in smart cities has become a major research trend [1, 2]. The autonomy nature of IoT brings virtual representation and unique identification of devices, applications, and services which helps for building a block of future smart cities [3, 4]. The emerging characteristics of the smart sensing system and vehicles make it possible for its enormous use like huge data collection and processing, transferring of data among the nodes of the network, connectivity of the nodes, decision making, and working accordingly by the devices [5, 6]. Due to that reason, to design a smart vehicle that contains some sensor nodes, Arduino microcontroller and also Controller Area Network (CAN) can be used. The CAN protocol provides a serial bus communication for a reliable and robust platform to transmit and receives a message from one sensor node to another [7, 8]. Using the CAN bus also provides a backbone for managing smart or automotive vehicle system engine control units (ECUs), brakes, and other components [9]. This CAN protocol is also used to improve the performance through distributed control and monitoring the congestion by virtue of sending priority messages from the various units of the vehicle [9, 10]. In this system, a carrier sense multiple access/collision detection (CSMA/CD) protocol and priority message scheduling are implemented to control overall communication, thereby reducing the cause of delays [11].

The CAN incorporates mechanism to design a real-time system with fault tolerance for communication at the time of occurrence of an error [12]. This system generally works in an electrical environment, whereas the communication is affected by electromagnetic interference (EMI) or any internal/external noise [13]. This noise can lead to reduction of performance of an automotive vehicle [13, 14]. There is a reason for causing an error or fault due to the unintended use of automatic repeat request (ARQ) [15]. Here, the corrupted message frames need to be retransmitted, but in this case, noncorrupted messages are also equally reforwarded, which causes inefficient utilization of the system resources [16]. So, it also takes more time to resend a long message, and thus, the response time of this message can be increased to cause an error. The system efficiency and overall performance are reduced due to the stop-and-wait resending strategy and wrong assigning of message length of the CAN bus [17]. This paper aims at enhancing the performance by adding a single error bit that depends on the impact of a faulty or corrupted message [18]. At the same time, CAN communication process is also improved through the analysis of probability automatic repeat request (PARQ) with its fault impact (PARQ-FI) which can verify the information message priority before retransmission.

This smart vehicle is connected with the Arduino controller to manage the various transmissions of messages from one sensor node to another [19]. When this controller is integrated with the CAN bus, then its transmission method is enhanced. There are also other factors like vehicle cabling pattern, unnecessary retransmission of frame, and poor energy source communication with distinct sensor nodes [20, 21]. Due to collision, data packets are consumed with excess energy which needs to retransmit fault packet for making an error-free system. Energy collection in today's world is being achieved from environmental sources of energy, such as solar, thermal, and vibration, to increase the source of sensor nodes from renewable energy [22–24]. Thus, the CAN protocol uses some fault controlling mechanism such as bit stuffing, acknowledgment verify, cyclic redundancy checking (CRC), and fault signaling, thereby saving power for retransmission of frame [25].

1.1. Problem Statement. In recent days, the main cause of road accidents arises very highly due to drowsiness and careless driving. Whenever the vehicle driver suddenly becomes conscious of braking, then the collision happens at the time of message packet communication in various nodes of the existing vehicle system. Generally, the data packets are lost or corrupted and also delayed in response that causes a fault in message transmission. Due to that, it is required to plan a smart autonomous vehicle to avoid a collision which is integrated with IoT modules and CAN serial bus protocol to reduce many road mishaps. So, it is used to implement an efficient fault-tolerant CAN protocol with an energy harvesting method to avoid the cause of the collision at the time of driving. Adding an autonomous movement is highly needed to an existing vehicle system availing for road safety as well as to making future smart cities.

1.2. Contribution to This Work. An efficient fault-tolerant and also optimal power allocation system is implemented on the smart autonomous vehicle by IoT modules using the Arduino controller and CAN bus on various sensor nodes of the system. The controlling of fault handling mechanism is applied through PARQ and PARQ-FI for assigning the priority of retransmission of message frames to transmit data to the brake of a vehicle for avoiding the collision. It is also providing an optimal approach to enhancement of power allocation in CAN network by implementing energy cost model with retransmission of data packet. Through effective packet routing and a congestion monitoring approach, the suggested paper optimally decreased power usage and data loss.

The rest of the paper is arranged accordingly. The literature review is discussed in Section 2. Section 3 provides a more detailed overview of the proposed system and the methodology in Section 4. Section 5 describes the tests and interpretation of the results. Section 6 concludes the article together with the likely scope for the future.

2. Literature Review

An efficient fault-tolerant algorithm for controller area network (CAN) is discussed in [26] that provides a technique to detect a fault at run time. This diagnosis algorithm is developed for a low-cost integrated system that can detect all defective nodes on CAN. Then, these faulty nodes are repaired during the detection process. So, it provides a single-way communication process with the help of standard CAN protocol with reliable communication of messages. This method only detects all faulty nodes at a particular period. When the testing process is not synchronized in time, then its performance is reduced, so it can be enhanced by a new perceptive of an algorithm for good repairing logic through proper synchronization of timing. Authors in [27] have proposed fault-tolerant convolutional neural networks (CNNs) for correcting serious errors in the real-time system. This technique provides safety for critical soft errors that are caused by an excess of voltage, temperature, and abnormal formation of an energy participle. The main cause of these soft errors is controlled through the implementation of three-way methods like using the technique of checksum, standard matrix-matrix multiplication, and the evaluation way of ImageNet using CNN models (AlexNet, VGG-19, etc.). But these experiments can be tested for soft errors in the restricted situation to minimize the overhead of an execution. That is why it is further planned to extend their work through more implementation of CNN model frameworks.

A novel identification technique is proposed in [28], which runs on the physical layer of the vehicular CAN network. The objective of this paper is to detect the erroneous electrical signaling of electronic control units (ECUs) of CAN messages. This process provides an efficient method for improving the CAN standard network for optimally detecting correct ECUs and improvising the logic for correction and detection of corrupted frame of ECUs, which is a vital requirement for avoiding surface attacks, road safety,

and flexibility to driver. So, it does not provide the optimal frame format of standard CAN without modifying the architecture of ECUs and thus, it can give better performance by enhancing the frame format of CAN. A dynamic energy harvesting torque technique is capable of providing a vector process to control and improve the efficiency of the motor wheel movement as elaborated in [29]. This vector process algorithm is capable of efficiently optimizing power consumption and executive torque of each side of the vehicle, which very swiftly reduces the run-time computation load, thereby allocating reasonable torque space for each side wheel of the vehicle. The nonlinear optimization criticality is swiftly transformed to optimal torque distribution of each side, which provides better optimal reduction of power loss between 13.9% and 18.9%, and also it needs a more advanced method for limiting the range of a torque of the wheels; otherwise, it fails.

Several security tools are embedded on the Internet for the vehicle which is detailed in [30] that integrates IoT modules with the CAN bus. This paper dynamically detects any malicious attacks and also hacking the control of a vehicle that causes safety issues for drivers and passengers. So, these CAN security measuring tools are used for evaluating the risk zones of a vehicle and major critical situations of passengers. Here, this system consumes more energy for handling their several sensor nodes and ECUs in a vehicle. The adaptable network protocol (CAN) is used in the automobiles communication process for detecting irregular connection fault that is proposed in [31]. The random cable connection in the car generates a communication problem that impairs the network system efficiency. In this paper, the fault detection is effective as well as economically detected by tree-based detection process, which is capable of eminently finding the location of the faulty node in overall communication. When the counter of sending data packet error crosses the threshold value that causes the failure of the system, it is rectified through the tree-based economical cost method. Thus, this tree-based process is required for optimal localization of internal collision, thereby detecting faults in the various critical topological environments.

The autonomous vehicle optimally performs the activity in the complex intersection of roadside mapping problem that is proposed in [32]. This paper claims that the traffic control signals do not efficiently manage and schedule the action of many vehicles in the recent situation; that is why it deigns an effective distributed algorithm and scheduling method that control the intersection of road path by minimizing the possible ratio of intersection delay. So, the case of complicated traffic system and first-in first-out (FIFO) scheduling are observed to have significantly minimized the ratio of the delay. Moreover, performance can be improved by using fault detection techniques. Authors in [33] described a clock synchronization and fault avoidance mechanism in CAN bus. This paper claims to have provided very optimal precision, supporting less transmission and operational overheads at the time of exchange of data packets. In the real-time implementation, this system is archiving better performance without an exchange of components of the CAN system. Then, the influence of energy consumption in the message packet and several scheduling criteria are not used in the clock synchronization of this CAN system.

An electric vehicle is designed with the integration of the powertrain method that is unswervingly linked with the vehicle motor and gearbox, which is mentioned in [34, 35]. This mechanism is improved through a strong efficiency in the reduction of size and optimal energy harvesting in it. Using this technique, the cause of oscillation damping and also linear matrix inequalities in the vehicle system can be resolved. If the actuator and parameter variability do not work properly, then it needs to be enhanced in the proposed controller. The threat to the CAN network represents the cyber-attacks that are detailed in [36], which can evaluate the ratio of securities of passengers and vehicles. This paper is capable of detecting anomalies and critical message packets through a one-class classification that implements an advanced classifier (image processing) to assign a fault-tolerant handling mechanism to the CAN system, whereas this single-type classifier is only tested in the random parameter of the CAN packet. But it is not tested for a feasible value of the wheel, which is critical to find the fault of the system. Traditional vehicular network suffers various technical issues in implementation and management as technology evolves and the number of smart vehicles grows. These challenges include a lack of flexibility, scalability, poor connectivity, and insufficient intelligence. Due to the huge number of vehicles on the road, traffic accidents, road congestion, fuel consumption, and pollution have all become important global issues.

3. Proposed System

This proposed system provides a dynamic and flexible environment to avoid any kind of occurrence of a collision, engine fault, and communication error. The automation is implemented with the use of IoT modules like Arduino controller, motor controller, various sensor nodes, and CAN protocol, in addition to the supply of energy source for improving communication platform in different nodes of the system.

3.1. Controller Area Network (CAN) for Proposed Architecture. The IoT-based modules are used to maximize the reconfiguration method and dynamic changes through its hardware and software [37, 38]. So, this is implemented in the CAN bus controller, in addition to the bit-stream controller (BSC) and PARQ/PARQ-FI that are used for retransmission of priority messages in the sub-block of the CAN protocol. This proposed CAN architecture is depicted in Figure 1. This controller controls the transmission, retransmission of error priority messages and reception of frames through various medium access control (MAC) or logical link control (LLC) by adding CRC field and extra control field.

At the time of communication to handle read and write process in configuration buffer which acts as a register, a microcontroller is used. The transmitting end (TE) and receiving end (RE) units are accessed that maintain separate

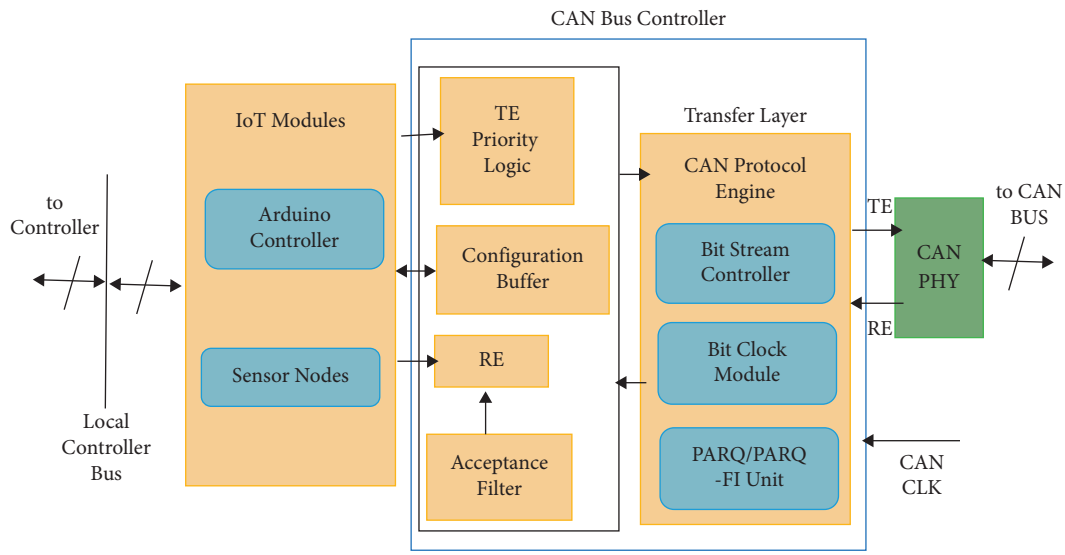


Figure 1: Proposed CAN bus block diagram.

buffers for communication of messages with the use of an acceptance filter (AF). The CAN protocol engine synchronizes the transmitted and received messages by using bit clock module and BSC module.

3.2. Retransmission of CAN Message Frame. The fault handling technique is effectively used for the communication of messages in CAN through the automatic repeat request (ARQ) method, which is a fault or error handling mechanism of resending the error message by proper observation of acknowledgments and timeouts for data transmission [39–41]. The redundant bits and CRC bit are used in the sending message stream for fault or an error detection which ensures reliable data communication link from source to a target node [14].

Whenever a message frame is sent in a noisy environment, then it is divided into tiny data packets with fault detection coding. If an error bit is zero, the packet will be accepted without error by the receiver. This indicates that the receiver is getting positive acknowledgment for the successful receipt of a message. But if there is no acknowledgment received, then the sender waits for a specified amount of time before resending the message packet [42, 43]. However, a negative acknowledgment (NACK) will be forwarded to the sender for resending the respective packet, if the receiver error bit is not the same as zero. Generally, the receiver node is waiting corrected message packet instead of a corrupted message packet. The acknowledgment feedback is used by the fault or error handling technique to inform whether the communication is correctly accomplished or not.

The message packet-1 is sent, and when the receiver receives it and finds that there is no-fault, then it forwards a positive acknowledgment (ACK) to signal a successful receipt of the packet [43, 44]. So, the communication continues with a message packet-2 which is received with an error and then, it sends a negative acknowledgment (NACK), which is shown in Figure 2.

Figure 3 explains the proposed PARQ/PARQ-FI communication technique, which is a combination of automatic error correction (AEC) and ARQ fault handling method to allow a corrupted packet to retransmit to the receiver node. But the corrupted packet is saved in a buffer for resending after correction. If an erred packet is received, then it sends the request for a new one, which is checked through AEC and the fault handling ARQ method that allow retransmitting corrupted packet from a buffer to avoid the collision [45, 46]. This technique conducts the communication in three ways as follows:

(i) Discarding an erred message packet, after receiving the retransmitted packet.

(ii) Redundant bits are used for resending actual bit message packets with the use of AEC.

(iii) The process of self-decoding method is used for recovering missing message bits.

Single-ended and differential CAN signals are processed by the CAN transceiver (CANH and CANL). In perfect condition, the CAN High and CAN Low lines are at 2.5 volts. The dominant bit in CAN is logic "zero," and the recessive bit is logic "one." When the dominant bit is communicated, CAN High rises to 3.5 volts and CAN Low falls to 1.5 volts, resulting in a 2-volt differential voltage. When the recessive bit is sent, the CAN High and CAN Low lines are both driven to 2.5 volts, indicating that the recessive bit's differential voltage is 0 volts. To eliminate signal reflections, a 120-ohm CAN bus terminal resistor should be placed to the physical end of the CANH and CANL lines.

4. Methodology

A dynamic and adaptive fault-tolerant approach is implemented to avoid the collision at the time of message packet transmission in CAN bus with the implementation of an optimal energy harvesting technique; that is the objective of this paper.

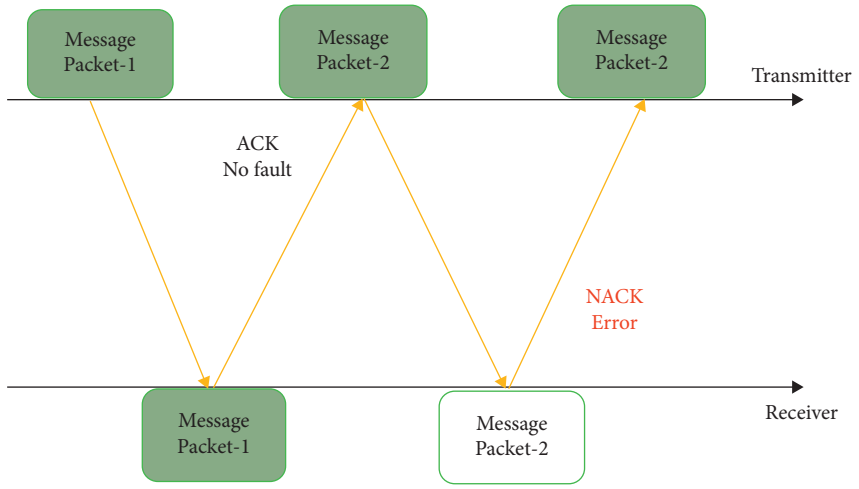


Figure 2: ARQ communication technique.

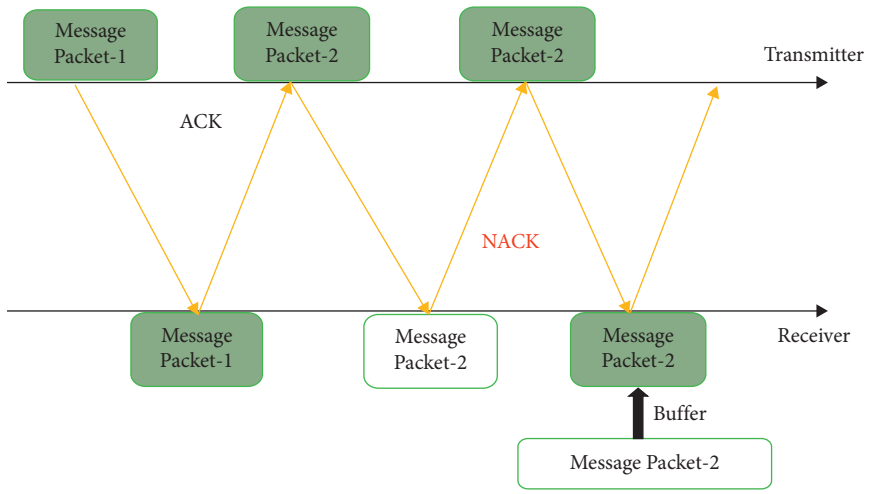


Figure 3: Proposed PARQ/PARQ-FI communication technique.

4.1. Fault/Error Handling Methods in CAN. Out of a number of messages transmitted from one node to another, some messages get corrupted in the CAN bus frame due to collision [47, 48]. So, it necessitates the use of fault or error handling technique to find a fault and request for resending of the message packet [39]. So, the CAN protocol is used for fault handling which is detailed below.

(i) When the bus line signal is varied from the transmitted signal, then the sender forwards an error message packet, which is analyzed by the bit monitoring field.

(ii) When the five sequential equal bits are found, then the bit stuffing assigns a complementary bit at the 6th bit location of the frame.

(iii) When a fixed frame format in the received message does not comply with the requirements of the protocol, a corrupted frame is sent by the recipient and the received frame is not accepted. So, it needs to verify the message frame format.

(iv) The acknowledgment check is checking the dominant bit frame slot for the data and distant frame.

(v) The cyclic redundancy check (CRC) bit is accessed for fault finding from the beginning of the frame (BOF) to the message packet field.

The standard CAN's message frame format is depicted in Figure 4.

4.1.1. Bit and Cyclic Redundancy Check (CRC) Errors. The analysis of CAN inaccessibility is considered at the time of communication of message bits from a source node to [49]. As soon as the first bit of a dataset is transmitted, bit error detection as per a transmitting node can take place. The corresponding network inaccessibility period is set to the best time required to signal the error:

$$C_{\text{inac}\leftarrow\text{bterr}} = C_{\text{bt}} + C_{\text{err}} + C_{\text{IFS}}, \quad (1)$$

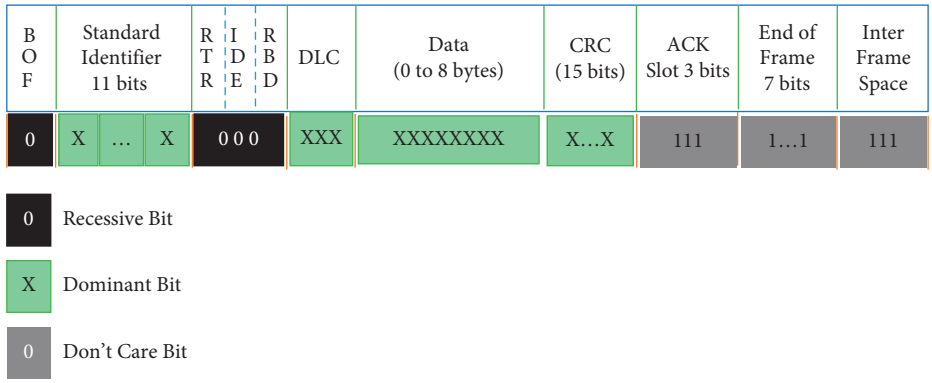


Figure 4: Standard message frame format of CAN.

where $C_{\text{inacbterr}}$ is denoted as error detection in the duration of communication network inaccessibility, C_{bt} confirms the communication of first bit of a bit stream, C_{err} is used for finding error at the of communication, and C_{IFS} is denoted as intermission field in the communication of message packet.

The CAN bus uses a 15-bit CRC frame with 1-bit delimiter, which is required to check the correctness of data transfer. Then, the inaccessibility time of communication network is expressed as

$$C_{\text{inac}\leftarrow\text{crc}} = C_{\text{data}} - t_{\text{EOF}} + C_{\text{err}} + C_{\text{IFS}}. \tag{2}$$

Here, the actual data frame is represented as C_{data} and t_{EOF} is denoted as time period of end-of-frame (EOF) field.

4.1.2. Bit Stuff Errors. The CAN access protocol, data transfer, and remote frames bit streams are carried out by means of a bit stuffing system until the end of their CRC 15-bit sequence. Frame receiving entities must therefore monitor this bit stream and provide bit reception decoding and error detection [50]. When a receiving node monitors l_{stff} consecutive bits of the same level, the next received (stuff) bit is automatically deleted. The deleted bit shows a polarity opposite to the above under error-free operation, and if this condition is violated, an error will be reported on the auto bus when an error frame is initiated. It is required to find communication duration of inaccessibility, due to stuff error that is expressed as

$$C_{\text{inac}\leftarrow\text{stuff}} = (l_{\text{stff}} + 1).C_{\text{bt}} + C_{\text{err}} + C_{\text{IFS}}. \tag{3}$$

4.2. Optimal Power Allocation Model for Sensor Node. The individual sensor node needs transmission and receiving energy. The initial energy (E) is also available for each sensor node [51]. Each node requires a low level of energy for sensing and computing so that we can ignore them. We need to take into consideration energy consumption across the nodes, as in the Load Balancing Network [52]. A sensor node of the smart vehicle with l_b-bit message packet received as $\text{RP}_x = E_{\text{pow}}.l_b$ that represents the consumed power through the first-order radio model. Here, the symbol E_{pow} stands for the necessary power to transmit all l_b-bit message packets for the transmitter or receiver sensor nodes. The communication energy consumed is thus indicated in the following equation:

$$\text{CP}_x = \text{RP}_x + E_{\text{pow}} * l_b * T_R^{\alpha}, \tag{4}$$

where E_{pow} is utilized for signaling a bit of power and the radius of transmission is indicated by T_R, and the propagation loss component is represented by α, where $2 \le \alpha \le 6$. This model incorporates uniform sensor nodes with the length of the data packet l_b and is then required to transmit the same power range for each message packet.

4.2.1. CAN Message Flow Conservation of Sensor Nodes. Here, the data transfer rates and how to send in the network via the flow conservation equation are discussed.

$$\sum_{j \epsilon M_i} \left(D_{ij}(t) - D_{ji}(t)\right) = R_{ij}(t), \quad \forall i \epsilon M,\ j \epsilon M_i, \tag{5}$$

where the above equation is expressed as follows:

(i) D_{ij} refers to the flow rate of messages from the node of the sensor.

(ii) R_{ij}represents the information rate of each sensor node acquired at the transmitter node (TR) i and to the recipient node (RN) $j \in M$.

(iii) M stands for all sets of sensor nodes, i = transmitting TR and j = receiving RN.

4.2.2. Estimation of Energy Cost System. The network life is dependent on the sensor node S_i energy consumption and active node scheduling time T_{Si}. The energy communicated per scheduled time $E_{\text{com}}(t)$ contains $E_{\text{RE}}(t)$ and $E_{\text{TE}}(t)$. The computed energy consists of $E_{\text{PE}}(t)$ and $E_{\text{SE}}(t)$.

Assuming the residual energy is represented by $E_B(t) \ge 0$, therefore, t's power consumption is defined as

$$\mathrm{PC}_i(t) = \sum_{i\epsilon N, j\epsilon N_i} D_{ij} E_{\mathrm{TE}}(t) + \sum_{i\epsilon N, j\epsilon N_i} D_{ij} E_{\mathrm{RE}}(t) + \sum_{i\epsilon N, j\epsilon N_j} R_{ij} E_{\mathrm{PE}}(t) + \sum_{i\epsilon N, j\epsilon N_i} R_{ij} E_{\mathrm{SE}}(t), \tag{6}$$

where $\mathrm{PC}_i(t)$ = power consumption in (t), $E_{\mathrm{RE}}(t)$ = receiver energy (t), $E_{\mathrm{SE}}(t)$ = sensing energy (t), $E_{\mathrm{TE}}(t)$ = transmitted energy (t), and $E_{\mathrm{PE}}(t)$ = processing energy (t).

For the transmission of one bit of data, the power transmitter from $i\epsilon S$ to $j\epsilon S_i$ over a distance (dt) is

$$\mathrm{E}_{\mathrm{TE}} = c_1 + c_2 * dt_{ij}^{k}, \tag{7}$$

where k = the transmission loss of path exponent, dt = interval of transmission, and c_1 and c_2 = constants according to T_i.

4.3. Corrupted Packet and Retransmission (PARQ/PARQ-FI) of CAN Message. Here, message analysis implements message error without fault (PARQ scheme), where the message is decoded by a technique as cyclic redundancy check (CRC) as

$$\mathrm{MP}_e^{\mathrm{PARQ}} = 1 - \left(1 - \mathrm{MP}_b\right)^{L_{mp}}, \tag{8}$$

where $\mathrm{MP}_e^{\mathrm{PARQ}}$ = probability of on event without fault impact, L_{mp} = message range, MP_e = messages loss rate, and MP_b = error bit rate (EBR). Similarly, the evaluation of message packet interference with the impact of fault (PARQ-FI scheme) is carried out, where the message communication is between the smart vehicle's sink node and sensor node. The loss rate of the sink node is

$$\mathrm{MP}_e^{\mathrm{PARQ-FI}} = 1 - \left(1 - \sum_{i=e+1}^{n} \left(\frac{n}{i}\right)\left(1 - \mathrm{MP}_b\right)^{n-i}\right)^{\left(L_{mp}/k\right)}. \tag{9}$$

Then, $\mathrm{MP}_e^{\mathrm{PARQ-FI}}$is the probability of on event with fault impact and here, the number of retransmission is given by

$$\mathrm{EX}(\mathrm{Tr}) = \frac{1}{\left(1 - \mathrm{MP}_e\right)}. \tag{10}$$

Here, EX (Tr) = denotes the expected number of repeated messages. For Hp-hop scenario, the data loss rate is examined when each node transmission is taken independently.

$$\mathrm{EX}(\mathrm{Tr}, \mathrm{Hp}) = \frac{\mathrm{Hp}}{\left(1 - \mathrm{MP}_e\right)}, \tag{11}$$

where Hp = set of hops. The fact that the values for signals u_{k-4}, u_{k-3}, u_{k-2}, u_{k-1}, v_{k-4}, v_{k-3}, v_{k-2}, v_{k-1} required for the controller are not defined during start-up is primarily the reason for the slow convergence to the constant status value. We then develop a heuristic process to identify appropriate values for a maximum of four periodic retransmitted message (RTM) periods as illustrated in Algorithm 1.

This algorithm can record a fault value from the beginning of the RTM received at $k = 1$. There will be no fault correction while waiting for four RTM (steps 3 and 5). The third RTM (step 6) is used to determine the input signal u_k and the clock error $\hat{v}$ at e_k as a result of a clock error difference between v_{k-1} and v_{k-2} (step 7).The timestamp and logical clock (LC) at e_4 will then be updated with n. The proposed correction of the fault is used in (step 9) after e_4 (step 11). One of the main advantages is that less clock variance is attained after four RTMs have been received when a CAN node is initiated.

Whenever all faults or corrupted message frame packets are collected at the period of transmission, each erroneous bit of message frame can be easily traced and also the parity bit correction can be achieved through the proposed algorithmic technique before transmission, which is shown in Figure 5. If the message frame is removed from an error or fault, then it is readily attached with the remaining bits of the frame before transmission. Otherwise, it is retransmitted for avoiding the cause of an erroneous situation.

4.4. Probability of CAN Message Packet. A transmitter node utilizes the following node to transmit message and the chance that a packet will receive the data for the other nodes [25]. It also specifies the time for the message to be sent after transferring information opportunities between nodes. The probabilities of their nearby nodes being received are modified. For the whole of the CAN bus node, the energy is adequate to transfer the message for the sake of energy harvesting with the probability of receiving the packet [53].

The performance of the system must be considered in view of the delivery ratio (DR) and delay. The expected count of the data packet is measured by dividing the sink node with a packet node that is needed to define the delivery ratio. The delay is the time taken to produce all the instances with the waiting time period to complete the entire pervious task. We must simulate and compare the PARQ and PARQ-FI methods of CAN to find the performance of the entire process.

5. Simulation Setup and Results Analysis

A smart fault-tolerant and optimal energy harvesting method are used to design a vehicle through CAN bus that is dynamically avoiding the cause of an internal message communication error of several sensor nodes, which is depicted in Figures 6(a) and 6(b). Due to this, dynamism is allowed to robustly detect and reduce the collision that occurs due to drowsiness or careless driving.

This proposed system is tested through the simulation setup which consists of 500 nodes and 1000 nodes. So, this setup is prepared with deployment area of 500 m × 600 m size where these two distinct ranges of nodes are implemented. Through the center of the implementation zone, the sink node is allocated which optimally provides energy and facilitates an exchange of message packets. We presume that

Step 1: **InsertParameter**: v_{k-1}, v_{k-2}, v_{k-3}, v_{k-4}, T
Step 2: **Result**: u_k
Step 3: **Init**: $u_{k-1} = 0$; $u_{k-2} = 0$; $u_{k-3} = 0$; $u_{k-4} = 0$; $k = 0$
Step 4: **if** RTM is accepted **then**
Step 5: $k = k + 1$; $n_{k-1} = e_{k-1,u} - e_{k-1,s}$
Step 6: **if** $k = 4$ (4^{th} RTM is received at e_4) **then**
Step 7: $u_k = (v_{k-1} - v_{k-2})/T$; $\hat{v}_k = v_{k-2} + (2.T.u_k)$
Step 8: $c_i(e_k) = c_i(e_k) + \hat{v}_k$ (clock update)
Step 9: $e_{k-1,s} = e_{k,s} + \hat{v}_k$ (timestamp overwrite)
Step 10: **else if** $k > 4$ **then**
Step 11: Find u_k according to (step 9)

Algorithm 1: Fault correction with initialization.

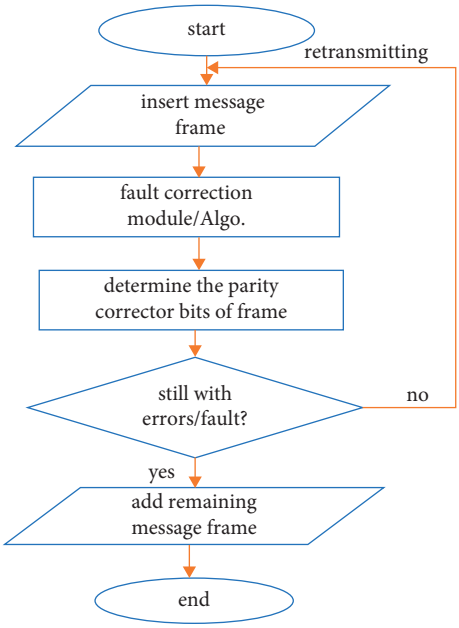
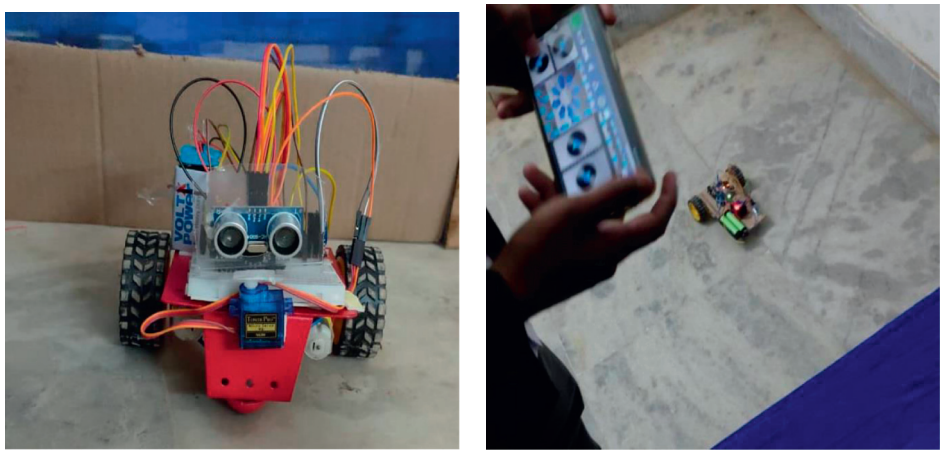


Figure 5: Flowchart of proposed fault handling (PARQ/PARQ-FI).

no packets can be received when the collision happens and data packet losses are not permitted in the entire network. This setup is set with the reliability threshold $T_h = 0.9$ for 500 nodes and 1.9 T_h value for 1000 nodes, as well as that, have to set the time depth to be 0 to Θ in a channel where the sink node in the center is located. Suppose that during the simulation phase, no message loss is feasible within the transmission range, but the lack of a precise message is a network collision, which means no correct packet could be accepted. As per the consideration, the communication power consumption is assigned as transmitting end $P_{te} = 77.1$ MW and receiving end $P_r e = 84.2$ MW, message packet sending energy $E_s = -3.2$ dbm, and message packet receiving energy $E_r = -84.7$ dbm. To set other parameters are the simulation period = 1000 s, data packet size set to 900 bits, rate of communication $R_c = 255$ kbps, and in addition, the frequency $(f) = 2.5$ GHz. These parameters are used for simulation result analysis, but this process also depends on several factors which are discussed below.

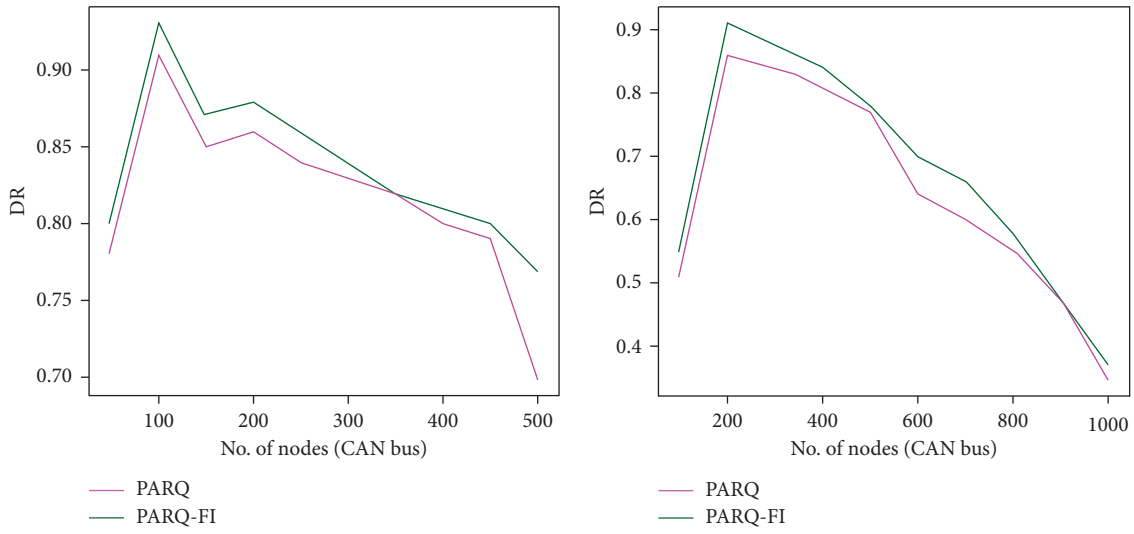
5.1. Analysis of Number of Sensor Nodes in CAN Bus versus Delivery Ratio (DR). When the number of sensors inside the CAN bus system varies, the delivery ratio is modified. Each node alters its density when it achieves the optimum amount of data transfer via PARQ and PARQ-FI. Figures 7(a) and 7(b) show that the PARQ-FI performs better than PARQ in 500 nos and 1000 nos of a sensor; if the fault does not occur frequently, then the collision is not considered to be successful. But the delivery ratio (DR) also decreases by raising the CAN message packet size.

5.2. Analysis of Delivery Ratio (DR) versus Charging Rate (Ch) of Nodes in CAN Bus. The system efficiency of the CAN bus is shown by the charge rate as shown in Figures 8(a) and 8(b). So, the PARQ performs better than PARQ-FI in 500 nos and 1000 nos of a sensor; if any protocol reaches a greater charge rate, the delivery ratio also rises, but the delivery ratio decreases for the charge time after some time.



(a) (b)

Figure 6: (a) Front view. (b) Test view of fault-tolerant smart vehicle.



(a) (b)

Figure 7: (a) 500 sensor nodes (n) versus delivery comparison ratio (DR) at PARQ and PARQ-FI. (b) 1000 sensor nodes (n) versus delivery comparison ratio (DR) at PARQ and PARQ-FI.

With a higher operating cycle, the sensor nodes charge quickly while the charge rate is increased and more packets are transmitted to the sink.

5.3. Analysis of Delivery Ratio (DR) versus Delay Factor (m) of Nodes in CAN Bus. In the communication range through PARQ and PARQ-FI, sensors of each node provide their next neighbors with their information as shown in Figures 9(a) and 9(b). Here the PARQ-FI performs better than PARQ in 500 nos and 1000 nos of a sensor which is already shown in the figures. The delivery ratio also decreases, if the protocol provides a higher delay ratio. As delay factor (m) is increased, the packets in the protocol are not guaranteed to be sent.

5.4. Analysis of Delivery Ratio versus Reliability Threshold T_h of Nodes in CAN Bus. Increased data delivery in PARQ PARQ-FI in Figures 10(a) and 10(b) depicts that the PARQ-FI performs better than PARQ in 500 nos and 1000 nos of a sensor if the reliability threshold of CAN bus node increases. The sensor sends the identical message back to obtain greater reliability if the reliability threshold (Th) value increases.

5.5. Analysis of the Number of Sensor Nodes versus Delay in CAN Bus. If more sensors are placed in the field of a smart vehicle installation, some sensor nodes require quick data recovery from the CAN bus. The time factor likewise grows with the increasing number of sensor nodes. The delay is evaluated with different nodes in Figures 11(a) and 11(b); as shown, the PARQ-FI performs better than PARQ in 500 nos

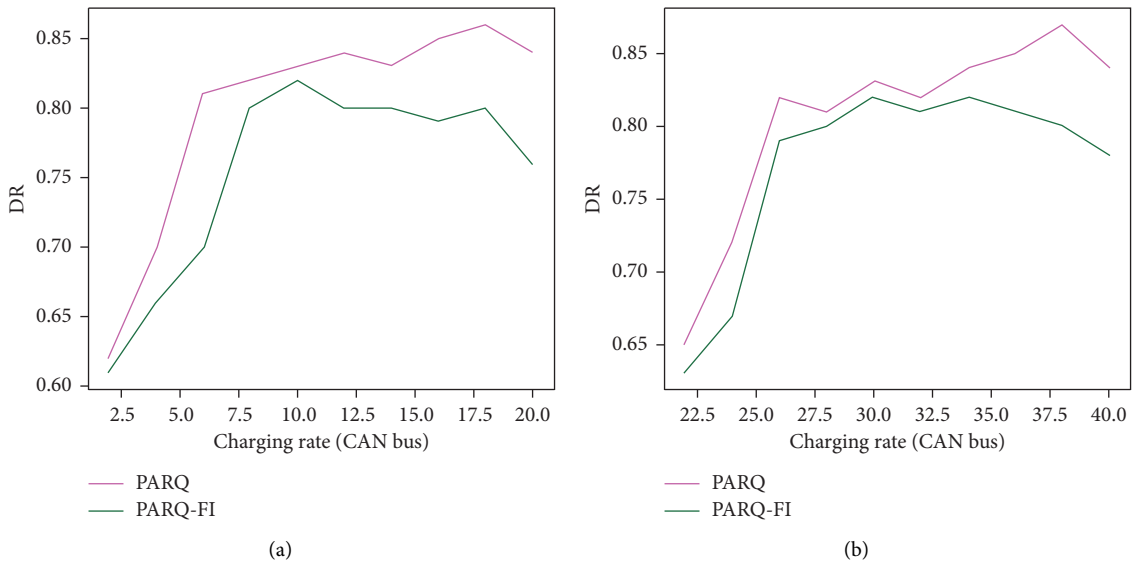


Figure 8: (a) Charging rate (Ch) versus delivery comparison ratio (DR) at PARQ and PARQ-FI for $n = 500$. (b) Charging rate (Ch) versus delivery comparison ratio (DR) at PARQ and PARQ-FI for $n = 1000$.

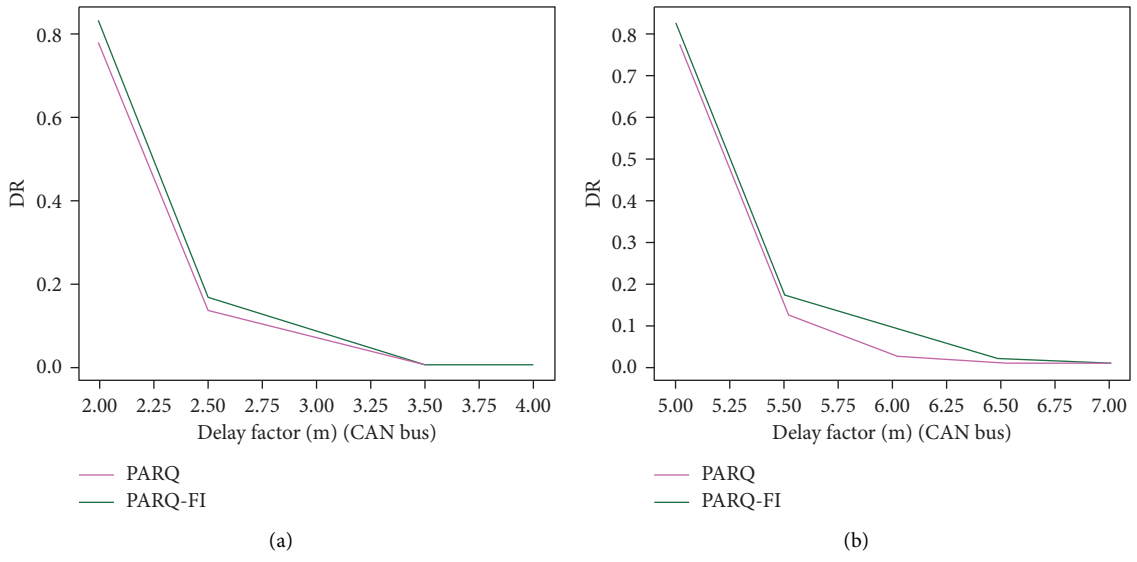


Figure 9: (a) Delivery ratio (DR) versus delay factor (m) of nodes comparison in PARQ and PARQ-FI for $n = 500$. (b) Delivery ratio (DR) versus delay factor (m) of nodes comparison in PARQ and PARQ-FI for $n = 1000$.

and 1000 nos of a sensor. Due to the limited charge time interval, the delay decreases as the charging rate rises in CAN protocol.

6. Discussion

In a test scenario, quite some parameters such as sensor nodes, delivery ratio, charging rate, delay factor, and reliability threshold have to be considered from the above analysis. Using these parameters in PARQ and PARQ-FI improves the fault-tolerant and precise power allocation for building smart vehicles in smart cities. The analysis of PARQ-FI provides better performance than PARQ as a comparison with sensor nodes versus delivery ratio, delivery ratio versus delay factor, delivery ratio versus reliability threshold, and sensor nodes versus delay in CAN bus. But a single variation consequence is simulated that the PARQ performs better than PARQ-FI in comparison with the sensor nodes versus the charging rate of nodes in the CAN bus. Therefore, we take the two different sets of 500 and 1000 sensor nodes in this paper and check or evaluate the fault-tolerant as well as optimal energy allocation in the vehicles. In order to maintain identification and responsiveness capacity, as well as improve data reliability, transmission, and

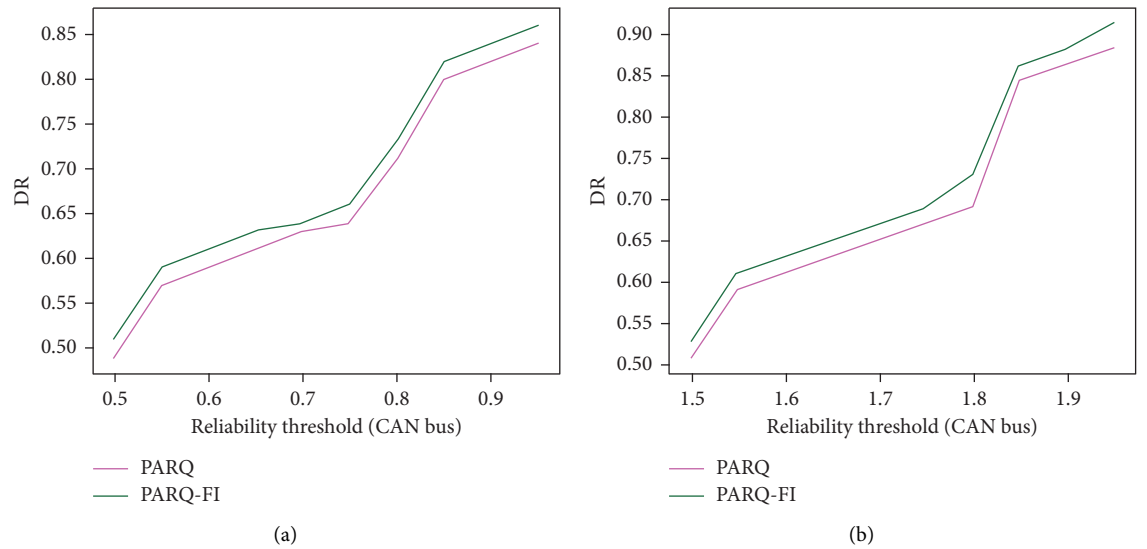


Figure 10: (a) Delivery ratio (DR) versus reliability threshold (Th = 0.9) of nodes comparison in PARQ and PARQ-FI for $n = 500$. (b) Delivery ratio (DR) versus reliability threshold (Th = 1.9) of nodes comparison in PARQ and PARQ-FI for $n = 1000$.

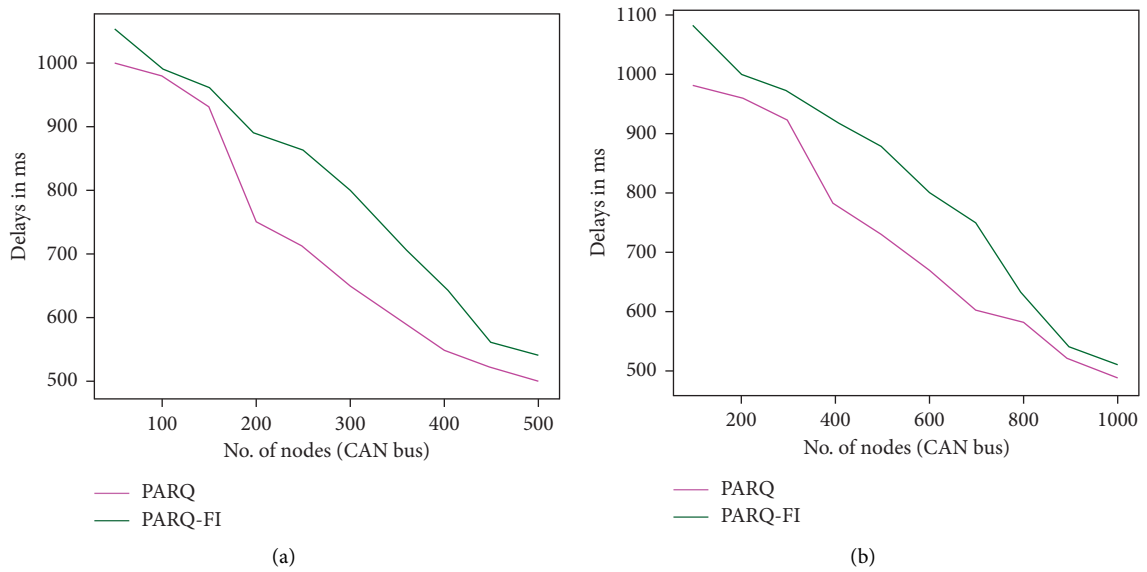


Figure 11: (a) 500 sensor node numbers versus delay (ms) comparison in PARQ and PARQ-FI (b) 1000 sensor node numbers versus delay (ms) comparison in PARQ and PARQ-FI.

redundancy, CAN network often requires high data dependability.

7. Conclusion

We analyzed the loss of data packet in the smart vehicle of CAN bus sensor nodes as a result of increased energy consumption and transmission collision in the proposed paper. To overcome this problem in future smart cities, the energy efficiency and automotive repeat priority message packet collection among neighboring nodes of a bus should be enhanced. That is why two methods are proposed for collection packets, such as Probabilistic Automatic Repeat Request (PARQ) and Probabilistic Automatic Repeat Request with Fault Impact (PARQ-FI). Similarly, data packets have not collided due to the adoption of an effective power distribution at the next node. In a sender node, the probability of accepting a message packet from their own enabled or active times and their larger archive delivery ratio are evaluated by simulating the results of each protocol. Thus, it is better usages of the protocol to observing the performance of smart vehicles in smart cities. The future study is focused on how best to identify the prospective threshold value T_h and the appropriate tracks/updates region for the message request τ may be created and the numerical analysis of CAN models and protocols enhanced in smart vehicles for various sensor network regions.

Data Availability

No data are available for this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

References

[1] S. S. Shah, M. Ali, A. W. Malik, M. A. Khan, and S. D. Ravana, "vFog: a vehicle-assisted computing framework for delay-sensitive applications in smart cities," *IEEE Access*, vol. 7, Article ID 34900, 2019.

[2] M. R. Boukhari, A. Chaibet, M. Boukhnifer, and S. Glaser, "A review on fault-tolerant control for vehicle dynamics," *International Journal of Digital Signals and Smart Systems*, vol. 1, no. 3, pp. 181–203, 2017.

[3] S. Tanwar, A. Popat, P. Bhattacharya, R. Gupta, and N. Kumar, "A taxonomy of energy optimization techniques for smart cities: architecture and future directions," *Expert Systems*, Article ID e12703, 2021.

[4] F. A. Turjman, "Smart-city medium access for smart mobility applications in internet of things," *Transactions on Emerging Telecommunications Technologies*, Article ID e3723, 2019.

[5] K. Soomro, M. N. M. Bhutta, Z. Khan, and M. A. Tahir, "Smart city big data analytics: an advanced review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 5, Article ID e1319, 2019.

[6] F.A. Turjman, H. Zahmatkesh, and R. Shahroze, "An overview of security and privacy in smart cities' iot communications," *Transactions on Emerging Telecommunications Technologies*, Article ID e3677, 2019.

[7] Y. Zhang, M. Chen, N. Guizani, D. Wu, and V. C. M. Leung, "SOVCAN: safety-oriented vehicular controller area network," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 94–99, 2017.

[8] X. Ge, I. Ahmad, Q. L. Han, J. Wang, and X. M. Zhang, "Dynamic event-triggered scheduling and control for vehicle active suspension over controller area network," *Mechanical Systems and Signal Processing*, vol. 152, Article ID 107481, 2021.

[9] G. I. Mary, Z. C. Alex, and L. Jenkins, "Time analysis of wireless controller area network-based fire and gas safety system," *International Journal of Communication Networks and Distributed Systems*, vol. 18, no. 1, pp. 1–17, 2017.

[10] M. K. Ishak and F. K. Khan, "Unique message authentication security approach based controller area network (CAN) for anti-lock braking system (ABS) in vehicle network," *Procedia Computer Science*, vol. 160, pp. 93–100, 2019.

[11] K. W. Schmidt, B. Alkan, E. G. Schmidt, D. C. Karani, and U. Karakaya, "Controller area network with priority queues and FIFO queues: improved schedulability analysis and message set extension," *International Journal of Vehicle Design*, vol. 71, no. 1–4, pp. 335–357, 2016.

[12] H. Olufowobi, U. Ezeobi, E. Muhati et al., "Anomaly detection approach using adaptive cumulative sum algorithm for controller area network," in *Proceedings of the ACM Workshop On Automotive Cybersecurity*, pp. 25–30, Richardson, TX, USA, March 2019.

[13] D. Kwon, S. Park, and J.-T. Ryu, "A study on big data thinking of the internet of things-based smart-connected car in conjunction with controller area network bus and 4G-long term evolution," *Symmetry*, vol. 9, no. 8, p. 152, 2017.

[14] N. M. B. Lakhal, O. Nasri, L. Adouane, and J. B. H. Slama, "Controller area network reliability: overview of design challenges and safety related perspectives of future transportation systems," *IET Intelligent Transport Systems*, vol. 14, no. 13, pp. 1727–1739, 2020.

[15] M. B. Hassan, E. S. Ali, R. A. Mokhtar, R. A. Saeed, and B. S. Chaudhari, "Nb-iot: concepts, applications, and deployment challenges," in *LPWAN Technologies For IoT and M2M Applications*Elsevier, Amsterdam, Netherlands, 2020.

[16] R. Kurachi, H. Takada, H. Ueda, and S. Takimoto, "Towards minimizing mac utilization for controller area network," in *Proceedings of the Second ACM Workshop on Automotive and Aerial Vehicle Security*, pp. 45–50, New Orleans, LA USA, March 2020.

[17] M. S. Chen and H. W. Yen, "Online controller area network message scheduling: analysis, implementation and applications," *International Journal of Systems, Control and Communications*, vol. 2, no. 4, pp. 418–430, 2010.

[18] V. Tanksale, "Anomaly detection for controller area networks using long short-term memory," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 1, pp. 253–265, 2020.

[19] S. Sicari, A. Rizzardi, and A. C. Porisini, "Smart transport and logistics: a node-red implementation," *Internet Technology Letters*, vol. 2, no. 2, p. e88, 2019.

[20] G. I. Aswath, S. K. Vasudevan, and R. M. D. Sundaram, "Emerging security concerns for smart vehicles and proposed iot solutions," *International Journal of Vehicle Autonomous Systems*, vol. 14, no. 2, pp. 107–133, 2018.

[21] A. K. Biswal, D. Singh, and B. K. Pattanayak, "IoT-based voice-controlled energy-efficient intelligent traffic and street light monitoring system," in *Proceedings of the Green Technology For Smart City And Society*, pp. 43–54, Springer, Bhubaneswar, India, August 2021.

[22] S. Almishari, N. Ababtein, P. Dash, and K. Naik, "An energy efficient real-time vehicle tracking system," in *Proceedings of the 2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pp. 1–6, IEEE, Victoria, BC, Canada, August 2017.

[23] J. Zhang, "Distributed network security framework of energy internet based on internet of things," *Sustainable Energy Technologies and Assessments*, vol. 44, Article ID 101051, 2021.

[24] N. A. Li, Y. Wei, M. Song, and X. Wang, "Energy-efficiency-aware flow-based access control in HetNets with renewable energy supply," *International Journal of Computational Science and Engineering*, vol. 21, no. 3, pp. 437–445, 2020.

[25] R. Alaei, P. Moallem, and A. Bohlooli, "Statistical based algorithm for reducing residual error in embedded systems implemented using the controller area network," *IEEE Access*, vol. 8, Article ID 133817, 2020.

[26] S. Kelkar and R. Kamal, "Adaptive fault diagnosis algorithm for controller area network," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5527–5537, 2014.

[27] K. Zhao, S. Di, S. Li et al., "FT-CNN: algorithm-based fault tolerance for convolutional neural networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1677–1689, 2021.

[28] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee, "Identifying ecus using inimitable characteristics of signals in controller area networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4757–4770, 2018.

[29] H. Chen, P. Du, Y. Wang, D. Jin, and X. Lian, "Dynamic energy-efficient torque allocation algorithm for in-wheel

motor-driven vehicle," *Proceedings of the Institution of Mechanical Engineers-Part D: Journal of Automobile Engineering*, vol. 234, no. 7, pp. 1815–1825, 2020.

[30] H. Zhang, X. Meng, X. Zhang, and Z. Liu, "CANsec: a practical in-vehicle controller area network security evaluation tool," *Sensors*, vol. 20, no. 17, p. 4900, 2020.

[31] L. Zhang, F. Yang, and Y. Lei, "Tree-based intermittent connection fault diagnosis for controller area network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 9151–9161, 2019.

[32] A. A. Hassan and H. A. Rakha, "A fully-distributed heuristic algorithm for control of autonomous vehicle movements at isolated intersections," *International Journal of Transportation Science and Technology*, vol. 3, no. 4, pp. 297–309, 2014.

[33] G. Rodriguez-Navas, S. Roca, and J. Proenza, "Orthogonal, fault-tolerant, and high-precision clock synchronization for the controller area network," *IEEE Transactions on Industrial Informatics*, vol. 4, no. 2, pp. 92–101, 2008.

[34] X. Zhu, H. Zhang, D. Cao, and Z. Fang, "Robust control of integrated motor-transmission powertrain system over controller area network for automotive applications," *Mechanical Systems and Signal Processing*, vol. 58-59, pp. 15–28, 2015.

[35] K. Jiang, H. Zhang, H. R. Karimi, J. Lin, and L. Song, "Simultaneous input and state estimation for integrated motor-transmission systems in a controller area network environment via an adaptive unscented Kalman filter," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 4, pp. 1570–1579, 2018.

[36] A. Tomlinson, J. Bryans, and S. A. Shaikh, "Using a one-class compound classifier to detect in-vehicle network attacks," in *Proceedings of the Genetic And Evolutionary Computation Conference Companion*, pp. 1926–1929, Kyoto Japan, July 2018.

[37] M. Priyan and G. U. Devi, "A survey on internet of vehicles: applications, technologies, challenges and opportunities," *International Journal of Advanced Intelligence Paradigms*, vol. 12, no. 1-2, pp. 98–119, 2019.

[38] A. K. Biswal, D. Singh, B. K. Pattanayak, D. Samanta, and M.-H. Yang, "IoT-based smart alert system for drowsy driver detection," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6627217, 13 pages, 2021.

[39] B. Palaniswamy, S. Camtepe, E. Foo, and J. Pieprzyk, "An efficient authentication scheme for intra-vehicular controller area network," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3107–3122, 2020.

[40] K. B. Kiadehi, A. M. Rahmani, and A. S. Molahosseini, "A fault-tolerant architecture for internet-of-things based on software-defined networks," *Telecommunication Systems*, vol. 77, pp. 1–15, 2021.

[41] A. de Baynast, P. Mähönen, and M. Petrova, "ARQ-based cross-layer optimization for wireless multicarrier transmission on cognitive radio networks," *Computer Networks*, vol. 52, no. 4, pp. 778–794, 2008.

[42] S. Garg, D. Mehrotra, H. M. Pandey, and S. Pandey, "Accessible review of internet of vehicle models for intelligent transportation and research gaps for potential future directions," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 1–28, 2021.

[43] M. Nakamura, M. Ohara, A. Saysanasongkham, M. Arai, K. Sakai, and S. Fukumoto, "Hybrid ARQ for DC-DC converter noise in controller area networks," in *Proceedings of the 2014 43rd International Conference On Parallel Processing Workshops*, pp. 375–379, IEEE, Minneapolis, MN, USA, September 2014.

[44] R. Shirai and T. Shimizu, "Failure protection for controller area network against EMI emitted by buck converter," in *Proceedings of the 2019 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pp. 644–649, IEEE, Anaheim, CA, USA, March 2019.

[45] G. A. A. Suhail, "Adaptive hybrid ARQ for mode switching receiver in wireless cellular networks," *International Journal of Reasoning-Based Intelligent Systems*, vol. 4, no. 4, pp. 192–196, 2012.

[46] M. Nakamura, M. Ohara, A. Saysanasongkham et al., "Testbeds of a hybrid-ARQ-based reliable communication for cans in highly electromagnetic environments," in *Proceedings of the 2015 IEEE 2nd International Future Energy Electronics Conference (IFEEC)*, pp. 1–6, IEEE, Taipei, Taiwan, November 2015.

[47] O. Avatefipour and H. Malik, "State-of-the-art survey on in-vehicle network communication (CAN-bus) security and vulnerabilities," 2018, https://arxiv.org/abs/1802.01725.

[48] P. S. Murvay, L. Popa, and B. Groza, "Securing the controller area network with covert voltage channels," *International Journal of Information Security*, pp. 1–15, 2021.

[49] R. Islam and R. U. D. Refat, "Improving CAN bus security by assigning dynamic arbitration ids," *Journal of Transportation Security*, vol. 13, no. 1, pp. 19–31, 2020.

[50] L. M. Zhang, Y. C. Sun, and Y. Lei, "Message delay time distribution analysis for controller area network under errors," *Frontiers of Information Technology and Electronic Engineering*, vol. 20, no. 6, pp. 760–772, 2019.

[51] C. Schmutzler, A. Krüger, F. Schuster, and M. Simons, "Energy efficient automotive networks: state of the art and challenges ahead," *International Journal of Communication Networks and Distributed Systems*, vol. 9, no. 3-4, pp. 266–285, 2012.

[52] S. S. Thale and V. Agarwal, "Controller area network assisted grid synchronization of a microgrid with renewable energy sources and storage," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1442–1452, 2015.

[53] R. Alaei, P. Moallem, and A. Bohlooli, "Statistical based algorithm for reducing bit stuffing in the controller area networks," *Microelectronics Journal*, vol. 101, Article ID 104794, 2020.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6537678, 13 pages
https://doi.org/10.1155/2021/6537678

Research Article

Improved Secure and Lightweight Authentication Scheme for Next-Generation IoT Infrastructure

Chien-Ming Chen and Shuangshuang Liu

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China

Correspondence should be addressed to Shuangshuang Liu; shuangliu0309@163.com

Received 25 June 2021; Revised 1 September 2021; Accepted 14 September 2021; Published 30 September 2021

Academic Editor: Azeem Irshad

The Internet of Things (IoT) is a huge network formed by connecting various information sensing devices through the Internet. Although IoT has been popularized in many fields, connected devices can be used only when network security is guaranteed. Recently, Rana et al. proposed a secure and lightweight authentication protocol for the next-generation IoT infrastructure. They claim that their protocol can resist major security attacks. However, in this study, we prove that their protocol is still vulnerable to offline password guessing attacks and privilege internal attacks. In order to solve these shortcomings, we propose an improved protocol, which is proved to be secure by formal and informal analysis. In addition, after comparing the time and memory consumption with other protocols, we find that our protocol has more advantages.

1. Introduction

In recent years, the Internet of Things (IoT) [1–4] has become popular in our everyday life. IoT refers to the real-time collection of any information that needs to be monitored, connected, and interacted with through the use of various devices and technologies such as sensors, radio frequency identification technology, global positioning system, and laser scanners. In the IoT environment, every object (virtual or physical) can be perceived, identified, accessed, and interconnected in a dynamic, ubiquitous network through the Internet. IoT brings great convenience to our lives. Vehicular ad hoc networks [5, 6] are considered to be one of the most promising applications of IoT. They allow people, vehicles, and roadside units to cooperate closely. IoT is also applied to medical healthcare, which is also closely related to our lives. Through the use of IoT, medical healthcare environments have taken on a new look. In an IoT-enabled healthcare system [7–9], wearable sensors can be used to collect information about patients and the surrounding environment. Another example of an IoT application is the smart home [10, 11]. Smart homes improve people's lifestyles and make them more comfortable, safer, and more efficient. In addition, the cloud system based on IoT can help the national government manage some resources to a great extent. The management data through the cloud system greatly reduces human resources and greatly improves the utilization rate of resources. These advantages are mainly based on the principle of the cloud-based Internet of Things. The application of such technology supports legitimate users to access normal data from hospitals, homes, borders, and other areas, which can better manage data to a certain extent.

Because IoT has grown so seamlessly, many end users are ignorant of the existence of these devices. Due to their invisibility, IoT device security is crucial, yet challenging to manage. Several IoT networks have recently been taken over to carry out malicious attacks. For these reasons, addressing these IoT security challenges is critical to their successful development. However, there has been a significant expansion in the number of IoT devices. Designing security mechanisms for all of these devices is complicated due to the heterogeneity and complexity of IoT networks.

For an IoT network to be secure, all the entities (servers, end users, and devices) must mutually authenticate their identities. In addition, all communication should be encrypted to maintain data confidentiality. This means that

a common session key for both sides of the communication is required. Therefore, designing a secure and efficient authenticated key agreement (AKA) scheme is crucial [12–15].

Various AKA schemes for IoT have been proposed. In 2004, Kumari et al. [16] found that Chang et al.'s scheme [17] is vulnerable to offline password-guessing attacks, internal attacks, and server masquerading attacks. They also pointed out that the protocol [17] has security vulnerabilities during the password update phase. To overcome these security weaknesses, Kumari et al. designed an improved scheme. Kumari et al. claimed that their scheme is more secure, more efficient, and more suitable for real-life IoT network use. However, Kaul and Awasthi [18] discovered that Kumari et al.'s protocol [16] is still vulnerable to some attacks. In their scheme, attackers can easily capture some security parameters transmitted on a public channel and then calculate the session key. In response to this, Kaul and Awasthi [18] proposed a robust and secure user authentication protocol based on resource-friendly symmetric cryptographic primitives. Unfortunately, Rana et al. [19] proved that the protocol [18] cannot resist various types of attacks. Therefore, they proposed a secure, lightweight AKA scheme for next-generation IoT infrastructure.

In this study, however, we found that Rana et al.'s scheme [19] is still vulnerable to offline password-guessing attacks and privileged-insider attacks. In their scheme [19], an illegal insider or malicious attacker can calculate the session key or guess passwords if they can capture a user's smart card. Therefore, we propose a new AKA scheme. In the proposed scheme, we utilize the biological information of the users because it is difficult for attackers to obtain this information. To demonstrate that the proposed scheme is indeed secure, we analyze it using Burrows–Abadi–Needham (BAN) logic [20] and also show that it is secure against various types of attacks. Compared with the previous scheme, the proposed scheme has better performance in terms of memory overhead.

The remainder of this paper is organized as follows. In Section 2, we briefly review the scheme proposed by Rana et al. [19]. Section 3 demonstrates that Rana et al.'s scheme [19] is vulnerable to offline password-guessing attacks and privileged-insider attacks. Our proposed scheme is described in Section 4. Sections 5 and 6 provide security and performance analyses and comparisons. Finally, Section 7 concludes the paper.

2. Review of Rana et al.'s Scheme

In this section, we briefly review Rana et al.'s AKA scheme. Their scheme contains three phases: user registration, login, and authentication, and the steps of their scheme are described below. Notations used in this paper are listed in Table 1.

2.1. User Registration Phase.

(1) First, the user U_c selects their own identification ID_c, password PW_c, and an arbitrary number b. Then, the following is calculated:

$$RPW_c = h(m \parallel PW_c), \tag{1}$$

and $\{ID_c, RPW_c\}$ is transmitted to the server through a secure channel.

(2) After the server receives the information from the user, it selects an arbitrary number y_c and calculates

$$\begin{aligned} DI\ D_c &= Enc_{ds}(ID_c \parallel y_c),\\ \alpha_c &= h(ID_c \oplus a) \parallel b,\\ \beta_c &= \alpha_c \oplus h(ID_c \oplus RPW_c),\\ \gamma_c &= y_c \oplus h(\alpha_c \oplus RPW_c),\\ \chi_c &= h(ID_c \parallel RPW_c \parallel y_c \parallel \alpha_c). \end{aligned} \tag{2}$$

(3) Then, the server stores the parameters $\{\beta_c, \gamma_c, \chi_c, DI\ D_c, h(\cdot)\}$ in the smart card memory and sends them to the user U_c through a secure channel.

(4) Finally, the user calculates

$$\eta_c = h(ID_c \oplus PW_c) \parallel m, \tag{3}$$

and stores η_c in the smart card. Now, the smart card contains parameters $\{\beta_c, \gamma_c, \chi_c, \eta_c, DI\ D_c, h(\cdot)\}$.

2.2. Login Phase. When a registered user wants to log in to the system, they perform the following operations:

(1) User U_c enters their ID_c' and PW_c' and inserts the smart card

(2) The smart card reader extracts parameters $m = \eta_c \oplus h(ID_c' \oplus PW_c')$ and $RPW_c' = h(m \parallel PW_c')$

(3) Further, the smart card reader can extract parameters $\alpha_c' = \beta_c \oplus h(ID_c' \oplus RPW_c')$ and $y_c' = \gamma_c \oplus h(\alpha_c' \oplus RPW_c')$ and calculate

$$\chi_c' = h(ID_c' \parallel RPW_c' \parallel y_c' \parallel \alpha_c'). \tag{4}$$

If $\chi_c' = \chi_c$, it means that the legitimate user is allowed to log in; otherwise, the login is refused

(4) After verifying the legitimacy of the user, the reader calculates

$$\begin{aligned} \omega_c &= y_c \oplus (ID_c' \oplus \alpha_c') \oplus h(ID_c' \oplus \alpha_c' \oplus T_1),\\ \nu_c &= h(ID_c' \parallel \alpha_c' \parallel y_c \parallel (\alpha_c' \oplus y_c) \parallel T_1). \end{aligned} \tag{5}$$

The reader then sends the login request $\{DI\ D_c, \omega_c, \nu_c, T_1\}$ to the server through a secure channel.

2.3. Authentication Phase. In this phase, the smart card reader and server authenticate each other by performing the following steps:

(1) S first verifies the validity of the timestamp by calculating T_2-T_1. If the calculated value is less than the given threshold δT, the login request proceeds; otherwise, it is rejected.

Table 1: Notations used in the proposed scheme.

Notations	Descriptions
U_c	c_{th} legal user
ID_c	c_{th} user identity
S	Legal server
PW_c	c_{th} user password
a, b	Private key and number of server
y_c	Arbitrary number for U_c
SC_c	User's smart card
T_1	Time stamp obtained at user's side
T_2	Server's current time stamp
T'	Threshold value
δT_c	Time of transmission delay
$\oplus$	XOR operator
$\|$	Concatenation operator
$h(\cdot)$	Noncollision hash function
SK	Session key
$\mathscr{A}$	The attacker
R_i	Biometric of U_c
ds	Long-term key
$\Rightarrow$	Private communication channel
$\longrightarrow$	Public communication channel

(2) After that, S extracts and calculates ID_c' using $(ID_c' \| y_c) = \mathrm{Dec}_{ds}(DI\ D_c)$ and then calculates the values:

$$\begin{aligned} \alpha_c' &= h(ID_c' \oplus a) \| b, \\ y_c' &= \omega_c' \oplus (ID_c' \oplus \alpha_c') \oplus h(ID_c' \oplus \alpha_c' \oplus T_1), \\ \nu_c' &= h(ID_c' \| \alpha_c' \| y_c' \| (\alpha_c' \oplus y_c') \| T_1). \end{aligned} \tag{6}$$

Then, S verifies the validity of the login by comparing the calculated ν_c' with the stored ν_c. If the two are equal, the verification passes; otherwise, the verification fails and the server refuses to accept the login request.

(3) After verifying the correctness of ν_c, the server continues to calculatez

$$\mu_c = h(ID_c' \| y_c' \| (\alpha_c' \oplus y_c') \| T_2). \tag{7}$$

Then, S sends the calculated μ_c and timestamp T_2 to U.

(4) When U receives the information from the server, it first verifies the validity of T_2 and then calculates

$$\mu_c' = h(ID_c \| y_c \| (\alpha_c \oplus y_c) \| T_2). \tag{8}$$

U checks whether μ_c' is equal to μ_c. If so, S is successfully verified.

(5) Finally, after mutual verification, the session key SK can be calculated:

$$SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2). \tag{9}$$

3. Cryptanalysis of Rana et al.'s Scheme

In this section, we first describe the threat model. Then, we show that Rana et al.'s scheme is insecure against offline password-guessing attacks and privileged-insider attacks.

3.1. Threat Model. This threat pattern shows the capabilities of an adversary, which are also considered and discussed in [21, 22]. $\mathscr{A}$'s capabilities are as follows:

(1) $\mathscr{A}$ can perform complete access control on the transmission channel. It can block, change, remove, replay, and hinder the messages passed between participants through a public channel.

(2) $\mathscr{A}$ can get the information stored in the smart card using power analysis [23, 24].

(3) $\mathscr{A}$ can obtain the information in the smart card and the information transmitted by the user on the secure channel during the registration process [25].

(4) $\mathscr{A}$ can simultaneously obtain the information in the smart card and perform offline password guessing as stated in [26, 27].

(5) $\mathscr{A}$ can know any two of the user's password, smart card, and biological information.

(6) $\mathscr{A}$ can obtain the session key that the user communicated with the server before.

(7) $\mathscr{A}$ can register as a legitimate user in a legitimate way.

3.2. Offline Password-Guessing Attack.

(1) First, the attacker $\mathcal{A}$ steals the smart card and gets the information $\{\beta_c^N, \gamma_c^N, \chi_c^N, \eta_c^N\}$

(2) $\mathcal{A}$ guesses the user's ID_c and PW_c at the same time

(3) According to the user's ID_c, password PW_c, and η_c and γ_c values obtained from the smart card, $\mathcal{A}$ calculates

$$\begin{aligned} m &= \eta_c \oplus h(ID_c \oplus PW_c), \\ RPW_c &= h(m \parallel PW_c), \\ \alpha_c &= \beta_c \oplus h(ID_c \oplus RPW_c), \\ y_c &= \gamma_c \oplus h(\alpha_c \oplus RPW_c). \end{aligned} \tag{10}$$

(4) Finally, $\mathcal{A}$ obtains the session key SK according to the value of α_c and y_c calculated above:

$$SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2). \tag{11}$$

3.3. Privileged-Insider Attack.

(1) First, the attacker $\mathcal{A}$ steals the smart card and gets the information $\{\beta_c^N, \gamma_c^N, \chi_c^N, \eta_c^N\}$

(2) Then, privileged insiders can obtain the information ID_C and RPW_c of legitimate users during registration

(3) $\mathcal{A}$ can calculate the following parameters by using the information β_c obtained in the smart card and the information ID_C and RPW_c obtained during user registration:

$$\begin{aligned} \alpha_c &= \beta_c \oplus h(ID_c \oplus RPW_c), \\ y_c &= \gamma_c \oplus h(\alpha_c \oplus RPW_c). \end{aligned} \tag{12}$$

(4) Finally, the attacker can calculate the session key SK according to the above parameters:

$$SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2). \tag{13}$$

4. Proposed Scheme

In this section, we describe the specific process of the protocol and the overall architecture diagram. The main body of the protocol includes users and servers. The agreement consists of four phases: user registration, login, authentication, and password change. Figure 1 illustrates the architecture of the proposed protocol. User represents the main participant in the communication, and server represents the entity that communicates with the user.

4.1. User Registration Phase. Figure 2 illustrates the user registration phase. The detailed steps are as follows:

(1) First, U_c selects their ID_c, password PW_c, and bio information R_i, as well as an arbitrary number m, to calculate

$$BRPW_c = (h(Ri) \oplus PW_c) \parallel m. \tag{14}$$

Then, ds is used to encrypt ID_c, with the result:

$$DI\ D_c = \mathrm{Enc}_{ds}(ID_c). \tag{15}$$

U_c then transmits $\{DI\ D_c, BRPW_c\}$ to S through a secure channel.

(2) After receiving the information from U, S selects an arbitrary number y_c to decrypt $DI\ D_c$, obtains the value of ID_c, and then calculates

$$\begin{aligned} ID_c &= De\ c_{ds}(DI\ D_c), \\ \alpha_c &= h(ID_c \oplus a) \parallel b, \\ \beta_c &= \alpha_c \oplus h(ID_c \oplus BRPW_c), \\ \gamma_c &= y_c \oplus h(\alpha_c \oplus BRPW_c), \\ \chi_c &= h(ID_c \parallel BRPW_c \parallel y_c \parallel \alpha_c). \end{aligned} \tag{16}$$

(3) Finally, the calculated parameters $\{\beta_c, \gamma_c, \chi_c, DI\ D_c, h(\cdot)\}$ are stored in the smart card, and S sends the smart card to U through a secure channel. U calculates η_c after receiving the message:

$$\eta_c = R_i \oplus m \oplus h(ID_c \oplus PW_c). \tag{17}$$

Then, η_c is saved in the smart card, and the registration process of the user is complete.

4.2. Login Phase.

(1) U enters their own ID_c', PW_c', and bio information R_i.

(2) After inputting the information, calculate

$$\begin{aligned} m &= \eta_c \oplus R_i \oplus h(ID_c' \oplus PW_c'), \\ BRPW_c' &= (h(R_i) \oplus PW_c') \parallel m, \\ \alpha_c' &= \beta_c \oplus h(ID_c' \oplus BRPW_c'), \\ y_c' &= \gamma_c \oplus h(\alpha_c' \oplus BRPW_c'), \\ \chi_c' &= h(ID_c' \parallel BRPW_c' \parallel y_c' \parallel \alpha_c'). \end{aligned} \tag{18}$$

Then, verify whether χ_c' and χ_c are equal. If they are equal, the verification passes; otherwise, the login request sent by U to S is rejected.

(3) If the verification passes, the reader will calculate

$$\begin{aligned} \omega_c &= y_c' \oplus h(ID_c' \oplus \alpha_c') \oplus h(ID_c' \oplus \alpha_c' \oplus T_1), \\ v_c &= h(ID_c' \parallel \alpha_c' \parallel y_c' \parallel (\alpha_c' \oplus y_c') \parallel T_1). \end{aligned} \tag{19}$$

Then, the login request $\{DI\ D_c, \omega_c, v_c, T_1\}$ is sent to the server.

4.3. Authentication Phase. This section describes the process of mutual authentication between S and U. After the user sends the login request to the server, the server starts to

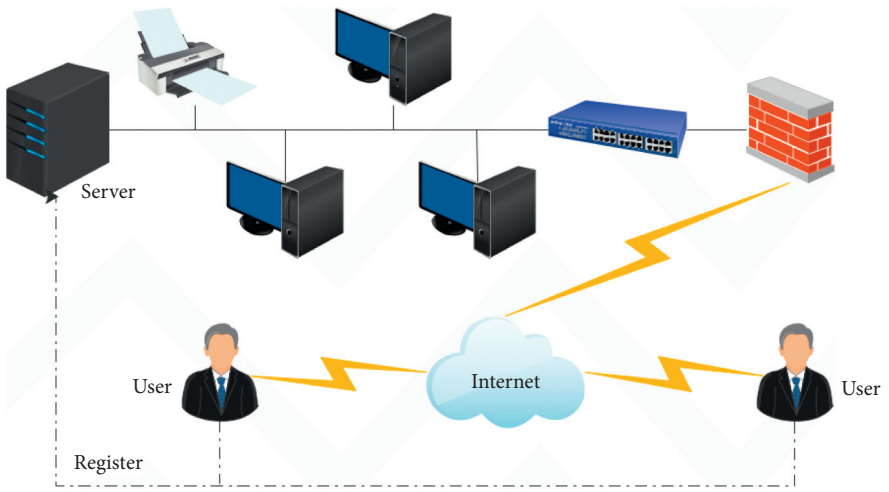


Figure 1: Network architecture.

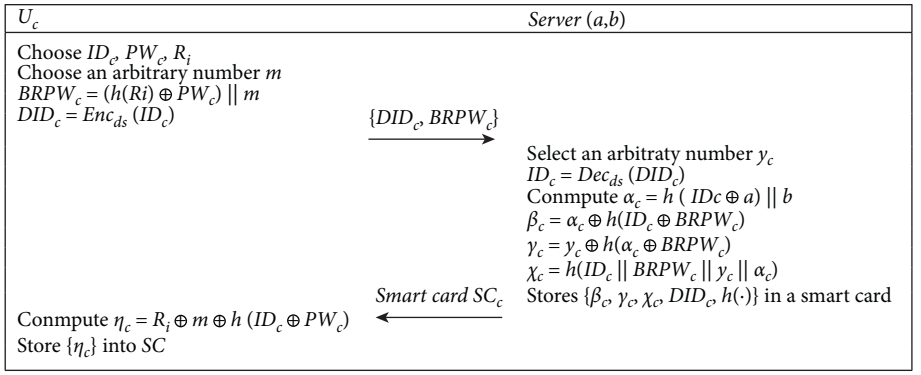


Figure 2: User registration phase.

verify whether U is legitimate by calculating a series of parameters, and U verifies the validity of S by calculating the values of some parameters. The authentication process is described in detail below. The login phase and authentication phase are shown in Figure 3.

(1) After S receives the request from U, it first verifies whether the present timestamp is reasonable. It then decrypts $DI\ D_c$ to obtain ID_c and calculates

$$\begin{aligned} \alpha_c' &= h((ID_c' \oplus a) \| b), \\ y_c' &= \omega_c' \oplus h(ID_c' \oplus \alpha_c') \oplus (ID_c' \oplus \alpha_c' \oplus T_1), \\ v_c' &= h(ID_c' \| \alpha_c' \| y_c' \| (\alpha_c' \oplus y_c') \| T_1). \end{aligned} \tag{20}$$

S verifies whether v_c' and v_c are equal. If not, S rejects the login request from U. If equal, S receives the login request from U and then calculates the session key of both sides:

$$SK = h(ID_c' \oplus \alpha_c' \oplus y_c' \oplus T_1 \oplus T_2). \tag{21}$$

(2) After calculating the session key, S continues to calculate

$$\mu_c = h(ID_c' \| y_c' \| (\alpha_c' \oplus y_c') \| T_2). \tag{22}$$

Then, S passes $\{\mu_c, T_2\}$ to U

(3) After receiving the message from S, the user first verifies the validity of the timestamp T_2 and then calculates

$$\mu_c' = h(ID_c \| y_c' \| (\alpha_c' \oplus y_c') \| T_2). \tag{23}$$

U verifies whether μ_c' is equal to μ_c. If it is equal, U calculates the session key:

$$SK = h(ID_c \oplus \alpha_c' \oplus y_c' \oplus T_1 \oplus T_2). \tag{24}$$

Here, the authentication process for U and S is completed.

4.4. Password Change Phase. If U wants to change their password PW_c to PW_c^N, the following steps are performed:

(1) U first inserts their own smart card and enters their ID_c, current password PW_c, bio information R_i, and new password PW_c^N.

$User U_c$	Server S
Input ID_c', PW_c', R_i	
Compute $m = \eta_c \oplus R_i \oplus h(ID_c' \oplus PW_c')$	
$BRPW_c = (h(R_i) \oplus PW_c') \mid\mid m$	
$\alpha_c' = \beta_c \oplus h(ID_c' \oplus BRPW_c')$	
$y_c' = \gamma_c \oplus h(\alpha_c' \oplus BRPW_c')$	
$\chi_c' = h(ID_c' \mid\mid BRPW_c' \mid\mid y_c' \mid\mid \alpha_c')$	
Check $\chi_c' \stackrel{?}{=} \chi_c$	
Compute $\omega_c = y_c \oplus (ID_c' + \alpha_c') \oplus h(ID_c' \oplus \alpha_c' \oplus T_1)$	
$v_c = h(IDc' \mid\mid \alpha_c' \mid\mid y_c \mid\mid (\alpha_c' \oplus y_c) \mid\mid T_1)$	
$DID_c, \omega_c, v_c, T_1 \longrightarrow$	
	$T_1' - T_1 < \Delta T$
	$(ID_c) = Dec_{ds}(DID_c)$
	Compute $\alpha_c' = h(ID_c' \oplus a) \mid\mid b$
	$y_c' = \omega_c' \oplus h(ID_c' \oplus \alpha_c') \oplus h(ID_c' \oplus \alpha_c' \oplus T_1)$
	$v_c' = h(ID_c' \mid\mid \alpha_c' \mid\mid y_c' \mid\mid \alpha_c' \oplus y_c') \mid\mid T_1$
	Check $v_c' \stackrel{?}{=} v_c$
	$SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$
	$\mu_c = h(ID_c' \mid\mid y_c' \mid\mid \alpha_c' \oplus y_c') \mid\mid T_2$
	$\mu_c, T_2 \longleftarrow$
Checks $T_2' - T_2 < \Delta T$	
Compute $\mu_c' = h(ID_c \mid\mid y_c \mid\mid (\alpha_c \oplus y_c) \mid\mid T_2)$	
Check $\mu_c' = \mu_c$	
$SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$	
Input $ID_c', PW_c', R_i, PW_c^{new}$	
Compute $m = \eta_c \oplus R_i \oplus h(ID_c' \oplus PW_c')$	
$BRPW_c = (h(R_i) \oplus PW_c') \mid\mid m$	
$\alpha_c' = \beta_c \oplus h(ID_c' \oplus BRPW_c')$	
$y_c' = \gamma_c \oplus h(\alpha_c' \oplus BRPW_c')$	
$\chi_c' = h(ID_c' \mid\mid BRPW_c' \mid\mid y_c' \mid\mid \alpha_c')$	
Check $\chi_c' \stackrel{?}{=} \chi_c$	
Compute $BRPW_c^N = h(m \mid\mid PW_c^N)$	
$\beta_c^N = \alpha_c \oplus h(ID_c \oplus BRPW_c^N)$	
$\gamma_c^N = y_c \oplus h(\alpha_c \oplus BRPW_c^N)$	
$\chi_c^N = h(ID_c \mid\mid BRPW_c^N \mid\mid y_c \mid\mid \alpha_c)$	
$\eta_c^N = h(ID_c \oplus PW_c^N) \oplus m \oplus R_i$	
Update $\beta_c^N, \gamma_c^N, \chi_c^N, \eta_c^N$ on the smart cars	

Figure 3: Login and authentication phase.

(2) According to some parameter values in the smart card and their own identity information, the following are calculated:

$$\begin{aligned} m &= \eta_c \oplus R_i \oplus h(ID_c' \oplus PW_c'), \\ BRPW_c' &= (h(R_i) \oplus PW_c') \parallel m, \\ \alpha_c' &= \beta_c \oplus h(ID_c' \oplus BRPW_c'), \\ y_c' &= \gamma_c \oplus h(\alpha_c' \oplus BRPW_c'), \\ \chi_c' &= h(ID_c' \parallel BRPW_c' \parallel y_c' \parallel \alpha_c'). \end{aligned} \tag{25}$$

If the calculated value of χ_c' is equal to the value of χ_c stored in the smart card, the user is considered legitimate and allowed to change the password.

(3) Some parameter values need to be updated in the process of password modification. The specific calculation process is as follows:

$$\begin{aligned} BRPW_c^N &= \left(h(Ri) \oplus PW_c^N\right) \parallel m, \\ \beta_c^N &= \alpha_c \oplus h\left(ID_c \oplus RPW_c^N\right), \\ \gamma_c^N &= y_c \oplus h\left(\alpha_c \oplus RPW_c^N\right), \\ \chi_c^N &= h\left(ID_c \parallel RPW_c^N \parallel y_c \parallel \alpha_c\right), \\ \eta_c^N &= R_i \oplus m \oplus h\left(ID_c \oplus PW_c^N\right). \end{aligned} \tag{26}$$

(4) Finally, the values $\{\beta_c, \gamma_c, \chi_c, \eta_c\}$ stored in the smart card are updated to the modified values $\{\beta_c^N, \gamma_c^N, \chi_c^N, \eta_c^N\}$, and the process of password modification is completed.

5. Security Analysis

5.1. Formal Security Analysis. Burrows–Abadi–Needham (BAN) logic [20] has been used in several studies to prove whether a protocol can be executed securely. This section uses BAN logic to prove the security and reliability of our proposed protocol. This proof verifies that our protocol can successfully establish and share a session key between the user and server. In the following proof, U represents the user and S represents the server. The specific proof rules and process are as follows:

5.1.1. BAN Logic Rules.

(i) Message-meaning rule (R1): $(U \mid \equiv U \overset{K}{\leftrightarrow} S, P \triangleleft \{M\}_K) / (U \mid \equiv S \mid \sim M)$ and $(U \mid \equiv U \overset{N}{\rightleftharpoons} NS, U \triangleleft \langle M \rangle_N) / (U \mid \equiv S \mid \sim M)$

(ii) Nonce-verification rule (R2): $(U \mid \equiv \sharp(M), U \mid \equiv S \mid \sim M) / (U \mid \equiv S \mid \equiv M)$

(iii) Jurisdiction rule (R3): $(U \mid \equiv S \mid \Rightarrow M, U \mid \equiv S \mid \equiv M) / (U \mid \equiv M)$

(iv) Freshness rule (R4): $(U \mid \equiv \sharp(M)) / (U \mid \equiv \sharp(M, N))$

(v) Belief rule (R5): $(U \mid \equiv M, U \mid \equiv N) / (U \mid \equiv (M, N))$

(vi) Session key rule (R6): $(U \mid \equiv \sharp(M), U \mid \equiv S \mid \equiv M) / (U \mid \equiv U \overset{K}{\leftrightarrow} S)$

5.1.2. Goals.

(i) G1: $U \mid \equiv U \overset{SK}{\leftrightarrow} S$

(ii) G2: $S \mid \equiv U \overset{SK}{\leftrightarrow} S$

(iii) G3: $U \mid \equiv S \mid \equiv U \overset{SK}{\leftrightarrow} S$

(iv) G4: $S \mid \equiv U \mid \equiv U \overset{SK}{\leftrightarrow} S$

5.1.3. Idealizing Communication.

(i) M1: $U \longrightarrow S$: $\{DI \quad D_c, \omega_c, v_c, T_1\}$

(ii) M2: $S \longrightarrow U$: $\{\mu_2, T_2\}$

5.1.4. Initial State Assumptions.

(i) A1: $U \mid \equiv U \overset{ds}{\rightleftharpoons} S$

(ii) A2: $S \mid \equiv U \overset{ds}{\rightleftharpoons} S$

(iii) A3: $S \mid \equiv \sharp(ID_c, \alpha_c, y_c,)$

(iv) A4: $S \mid \equiv U \mid \Rightarrow ID_c$

(v) A5: $S \mid \equiv U \overset{ID_c}{\rightleftharpoons} S$

(vi) A6: $U \mid \equiv U \overset{ID_c}{\rightleftharpoons} S$

(vii) A7: $S \mid \equiv U \mid \Rightarrow (\alpha_c, y_c)$

(viii) A8: $S \mid \equiv \sharp(ID_c, \alpha_c, y_c,)$

(ix) A9: $U \mid \equiv S \mid \Rightarrow (\alpha_c, y_c)$

5.1.5. Detailed Steps.

By considering the message M1 and using the seeing rule, we get

S1: $S \triangleleft \{\langle ID_c \rangle_{ds}, \langle \alpha_c, y_c \rangle_{ds}, T_1\}$.

Using S1, we get

S2: $S \triangleleft \{\langle ID_c \rangle_{ds}\}$.

Under the assumption of A2, using S2, R1 can be used to obtain

S3: $S \mid \equiv U \mid \sim (ID_c)$.

With conclusion S3, using A3 and R2, the following can be obtained:

S4: $S \mid \equiv U \mid \equiv (ID_c)$.

Using A4, R3, and conclusion S4, the following can be obtained:

S5: $S \mid \equiv (ID_c)$.

According to conclusion S1, the following can be obtained:

S6: $S \triangleleft \{\langle \alpha_c, y_c \rangle_{ID_c}\}$.

Using A6, R1, and conclusion S6, the following can be obtained:

S7: $S \mid \equiv U \mid \sim (\alpha_c, y_c)$.

Using A3, R2, and conclusion S7, the following can be obtained:

S8: $S \mid \equiv U \mid \equiv (\alpha_c, y_c)$.

Using A7, R3, and conclusion S8, the following can be obtained:

S9: $S \mid \equiv (\alpha_c, y_c)$.

Because $SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$, using S5 and S9, we obtain

S10: $S \mid \equiv U \overset{SK}{\leftrightarrow} S$ (G2).

Using A3 and R4, we can obtain

S11: $S \mid \equiv U \mid \equiv U \overset{SK}{\leftrightarrow} S$ (G4).

In addition, considering the message M2, we obtain

S12: $U \lhd \{\langle \alpha_c, y_c \rangle_{ID_c}, T_2\}$.

By using A6, S1, and R1, we obtain

S13: $U \mid \equiv S \mid \sim (\alpha_c, y_c)$.

With conclusion S13, using A8 and applying R2, we obtain

S14: $U \mid \equiv S \mid \equiv (\alpha_c, y_c)$.

Applying A9, S14, and R3, we obtain

S15: $U \mid \equiv (\alpha_c, y_c)$.

Because $SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$, using S5 and S9, we obtain

S16: $U \mid \equiv S \overset{SK}{\leftrightarrow} S$ (G1).

With conclusion S16, using A8 and R4, we can obtain

S17: $U \mid \equiv S \mid \equiv U \overset{SK}{\leftrightarrow} S$(G3).

5.2. ROR Formal Security Proof

5.2.1. ROR Model. This paper follows the ROR (Random Oracles) model under the proof of security, and two participants U and S are mentioned in the paper. First, let H_U^x and H_S^y as the xth user and yth server, respectively. Then, let $\mathcal{U} = \{H_U^x, H_S^y\}$ and $\mathcal{A}$ can perform the following operations.

Execute$(\mathcal{U})$: by executing this query, $\mathcal{A}$ can get the messages transmitted by U and S through the common channel.

Send$(\mathcal{U}, \mathcal{M})$: with the help of send query, $\mathcal{A}$ can send messages to U and S. In addition, $\mathcal{A}$ can also receive response messages from two participants.

Corrupt$(\mathcal{U})$: with the help of this query, $\mathcal{A}$ can obtain the parameters information stored in the smart card as well as some temporary parameters information and long-term key.

Hash(String): by performing this operation, $\mathcal{A}$ can obtain the value in the hash.

Test$(\mathcal{U})$: this operation is mainly used to verify whether the session key between the user and the server is secure. By tossing a homogeneous coin $\mathcal{C}$, the result of the coin is known only to $\mathcal{A}$. If $\mathcal{C} = 1$, $\mathcal{A}$ can know the correct session key. If $\mathcal{C} = 0$, a null value is an output.

Definition 1 (one-way anticollision hash function): this is a common mathematical function that inputs a variable length field and then produces a fixed length output. If $\mathrm{Adv}(m) = \Pr[(m, n)\varepsilon_R A; h(m) = h(n)] \le t$ for at most run time m, the hash function is considered hash collision proof.

Definition 2 Symmetric encryption method is used in the proposed protocol. Suppose $E_{K_1}, E_{k_2}, \ldots, E_{K_n}$ are encryption methods based on different keys K. In the model, the probability that $\mathcal{A}$ can crack the correct session key is $\mathrm{Adv}_A^K(\eta) = |2\Pr[A \longleftarrow E_{K_1}; (b_0, b_1) \longleftarrow A; \alpha \longleftarrow 0, 1; \beta \longleftarrow E_{K_1}(b_\alpha) : \mathcal{A}(\beta) = \alpha] - 1|$.

Theorem 1. *If $\mathcal{A}$ is a polynomial time η opponent executing our scheme under the ROR model and we choose to look at Zipf's law [28] for the user's password, the possibility of $\mathcal{A}$ damaging the session key is $Adv_A^P(\eta) \le (t_{send} + t_{exe})^2/2^{u-1} + 2\,Adv_A^K(\eta) + t_{hash}^2 \cdot 2^{l-1} + 2\max\{D' \cdot t_{send}^{X'}, t_{send}/2^l\}$ + where l represents the length of the password.*

5.2.2. Security Proof

Proof. In the proof process, we define six games GM_0 to GM_5 and prove the theorem mentioned above according to the defined six game rules. $\mathrm{Succ}_A^{GMi}(\eta)$ represents the probability of $\mathcal{A}'$s success in the game. The specific proof is as follows.

GM_0: in the initial game, $\mathcal{A}$ does not perform any query operations. According to the definition of security primitives, we can get $\mathrm{Adv}_A^P(\eta) = |2\Pr[\mathrm{Succ}_A^{GM0}(\eta)]|$.

GM_1: GM_1 adds the *execute* operation on the basis of GM_0, that is, $\mathcal{A}$ can intercept and tamper with the information transmitted on the public channel $M_1 = \{DI\ D_c, \omega_c, v_c, T_1\}$ and $M_2 = \{\mu_c, T_2\}$. However, $\mathcal{A}$ cannot obtain the session keys of both parties according to the information obtained on the public channel, so the probability of GM_1 is equal to that of GM_0, $\Pr[\mathrm{Succ}_A^{GM1}(\eta)] = \Pr[\mathrm{Succ}_A^{GM0}(\eta)]$.

GM_2: GM_2 adds Hash and Send query operations on the basis of GM_1. According to the birthday paradox, it can be concluded that the maximum probability of hash collision is $t_{\mathrm{hash}}^2/2^{l+1}$. Therefore, it can be concluded that the maximum probability of hash collision of text transmitted by both sides of the session is $(t_{\mathrm{send}} + t_{\mathrm{exe}})^2/2^u$. Finally, we can draw a conclusion $|\Pr[\mathrm{Succ}_A^{GM2}(\eta)] - \Pr[\mathrm{Succ}_A^{GM1}(\eta)]| \le t_{\mathrm{hash}}^2/2^{l+1} + (t_{\mathrm{send}} + t_{\mathrm{exe}})^2/2^u$. The symbol l appearing in the formula represents the length of the hash value and u represents the length of the transmitted text.

GM_3: on the basis of the above game rules, we added the provision that $\mathcal{A}$ can obtain the parameters information stored in the smart card in the new round of game, that is, $\mathcal{A}$ can obtain the parameters $\{\beta_c, y_c, \gamma_c, DI\ D_c\}$ by executing the *Corrupt* operation. On this basis, we perform an offline password guessing attack. First, $\mathcal{A}$ calculates $\alpha_c = \beta_c \oplus h(ID_c' \oplus BRPW')$, $BRPW' = (h(R_i) \oplus PW_c') \parallel m$, but U's identity ID_c and

U's biological information R_i are confidential to us, so they cannot be obtained. According to Zipf's law [28], we can draw a conclusion: $|Pr[Succ_A^{GM3}(\eta)] - Pr[Succ_A^{GM2}(\eta)]| \le max\{D' \cdot t_{send}^{X'}, t_{send}/2^l\}$.

GM_4: in this game rule, we analyze the security of the communication session key between both sides. We mainly analyze it from the following three aspects. The first is to prove that the protocol has perfect forward security. The second is to prove that $\mathscr{A}$ can block the user impersonation attacks. The third is that $\mathscr{A}$ can block the known session-specific temporary information attacks.

Perfect forward security: $\mathscr{A}$ obtains the value of the long-term key ds through $Corrupt$.

Known session-specific temporary information attacks: $\mathscr{A}$ obtains the value of temporary information m or y_c through $Corrupt$ query.

User impersonation attacks: $\mathscr{A}$ obtains the information $\{DI\ D_c, \omega_c.\nu_c, T_1\}$ transmitted by both communication parties through the public channel through Exe query, but U's identity ID_c is obtained by symmetric encryption with the long-term key ds. However, the value of the long-term key ds cannot be obtained.

The session key $SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$ of both communication parties: in the first case, $\mathscr{A}$ must obtain the values of α_c and y_c in order to obtain the session key, but the value of α_c needs U's biological information. In the second case, $\mathscr{A}$ obtains the value of temporary information, but U's identity ID_c is obtained through symmetric encryption. In the third case, because U's identity ID_c is obtained through symmetric encryption, $\mathscr{A}$ cannot obtain U's real identity, so it is impossible to carry out simulated attacks. Therefore, we can conclude that $|\Pr[\mathrm{Succ}_A^{GM4}(\eta)] - \Pr[\mathrm{Succ}_A^{GM3}(\eta)]| \le A\ DV_A^K(\eta)$.

GM_5: in the final rule of the game, $\mathscr{A}$ uses hash query $h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$; then, $\mathscr{A}$ can guess the possibility of the session key: $|\Pr[\mathrm{Succ}_A^{GM5}(\eta)] - \Pr[\mathrm{Succ}_A^{GM4}(\eta)]| \le t_{\mathrm{hash}}^2/2^{l+1}$.

As we all know, the probability of guessing the session key correctly is $|\Pr[\mathrm{Succ}_A^{GM5}(\eta)] = 1/2$.

To sum up, we can get it according to the above formula:

$$\begin{aligned}\frac{1}{2}\mathrm{Adv}_A^P(\eta) &= \Pr\left[\mathrm{Succ}_A^{GM_0}(\eta)\right] - \frac{1}{2} = \Pr\left[\mathrm{Succ}_A^{GM_0}(\eta)\right] - \Pr\left[\mathrm{Succ}_A^{GM_5}(\eta)\right] = \Pr\left[\mathrm{Succ}_A^{GM_1}(\eta)\right] \\ &\quad - \Pr\left[\mathrm{Succ}_A^{GM_5}(\eta)\right] \le \Pr\left[\mathrm{Succ}_A^{GM_5}(\eta)\right] - \Pr\left[\mathrm{Succ}_A^{GM_4}(\eta)\right] + \Pr\left[\mathrm{Succ}_A^{GM_4}(\eta)\right] - \Pr\left[\mathrm{Succ}_A^{GM_3}(\eta)\right] \\ &\quad + \Pr\left[\mathrm{Succ}_A^{GM_3}(\eta)\right] - \Pr\left[\mathrm{Succ}_A^{GM_2}(\eta)\right] + \Pr\left[\mathrm{Succ}_A^{GM_2}(\eta)\right] - \Pr\left[\mathrm{Succ}_A^{GM_1}(\eta)\right] \\ &= \frac{(t_{\mathrm{send}} + t_{\mathrm{exe}})^2}{2^u} + \mathrm{Adv}_A^K(\eta) + t_{\mathrm{hash}}^2 2^l + \max\left\{D' \cdot \frac{t_{\mathrm{send}}^{X'}, t_{\mathrm{send}}}{2^l}\right\}.\end{aligned} \tag{27}$$

So, we come to the final conclusion $\mathrm{Adv}_A^P(\eta) \le (t_{sen\ d} + t_{exe})^2\ /2^{u-1} + 2\,\mathrm{Adv}_A^K(\eta) + t_{\mathrm{hash}}^2\ \cdot 2^{l-1}\ + 2\max\{D' \cdot t_{\mathrm{send}}^{X}\ \prime, t_{\mathrm{send}}\ /2^l\}$. □

5.3. Informal Security Analysis. In this section, we further show that the proposed scheme is secure against the following attacks.

5.3.1. Privileged-Insider Attack. In this protocol, even if the attacker obtains the information $\{DI\ D_c, BRPW_c\}$ of the user in the registration process and the information $\{\beta_c, \gamma_c, \chi_c, \eta_c\}$ in the smart card, they cannot successfully obtain the session key. Because $SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$ and the user's ID_c is encrypted by ds before being transmitted to the server, even if the attacker obtains the value of $DI\ D_c$ and $BRPW_c$, the attack is futile. Therefore, this protocol can resist privileged-internal attacks.

5.3.2. Offline Password-Guessing Attacks. Suppose the attacker gets the message in the smart card; then, based on this message, they can guess the password offline. Even if the η_c value in the smart card is obtained and the values of ID_c and PW_c are guessed, the offline password-guessing operation cannot be successful. This is because the calculation of m also involves the value of the user's biological information R_i, and the value of R_i is difficult to obtain. Therefore, this protocol can effectively resist offline password-guessing attacks.

5.3.3. Replay Attack. Suppose that the malicious attacker intercepts the login information $\{DI\ D_c, \omega_c, \nu_c, T_1\}$ and authentication information $\{\mu_c, T_2\}$ and attempts to replay the login request. The request is invalid because we use the timestamp T_1 in the protocol to verify whether the time difference is within the set time threshold. Similarly, if the attacker intercepts the authentication message and attempts to make the authentication request, the user will also test the validity of the timestamp. Therefore, the protocol can effectively resist replay attacks.

5.3.4. Forward Secrecy. Assuming that the attacker obtains the value of the long-term password ds, they can only use this value to decrypt $DI\ D_c$ to obtain the value of the user's ID_c. However, because $SK = h(ID_c \oplus \alpha_c \oplus y_c \oplus T_1 \oplus T_2)$, it is not

Table 2: Comparisons of security.

Protocols	A1	A2	A3	A4	A5	A6
Rana et al. [19]	No	No	Yes	Yes	Yes	Yes
Kaul and Awasthi [18]	Yes	Yes	Yes	Yes	Yes	No
Xue et al. [29]	Yes	No	Yes	Yes	Yes	Yes
Lin et al. [30]	Yes	Yes	Yes	Yes	Yes	No
Chang et al. [17]	No	No	No	Yes	Yes	No
Kumari et al. [16]	No	Yes	No	Yes	Yes	Yes
Ours	Yes	Yes	Yes	Yes	Yes	Yes

sufficient to only know the value of the user's ID_c. The values of the parameters α_c and y_c cannot be obtained. Therefore, this protocol can provide perfect forward security.

5.3.5. Known Session-Specific Temporary Information Attacks. Assuming that the attacker obtains the value of temporary session information m or y_c, the session key cannot be obtained successfully. Because the session key calculation is composed of ID_c, but ID_c is encrypted by long-term key ds, the ID_c cannot be obtained by the attacker. Therefore, this protocol can successfully resist known session-specific temporary information attacks.

5.3.6. User Impersonation Attacks. Suppose that the attacker wants to carry out a user impersonation attack. They must first obtain the value of ID_c, but ID_c is encrypted by the long-term key ds, and so, it is difficult for the attacker to obtain its value. In addition, assuming that the attacker intercepts the message $\{DI\ D_c, \omega_c, \nu_c, T_1\}$ from the public channel and sends it to the server for verification, the user needs a certain amount of time to decrypt $DI\ D_c$. Therefore, when the server receives the message from the attacker for verification of the timestamp, it will find that the timestamp exceeds the set time domain and reject the login request. In this way, our protocol successfully resists user impersonation attacks.

5.3.7. Mutual Authentication. In this protocol, users and servers can successfully authenticate each other. First of all, the server authenticates the user through the value of v_c sent by the user. Similarly, the user can verify whether the server is legitimate through the value of μ_c sent by the server. Only legitimate users and servers can pass the authentication. Therefore, this protocol can effectively provide mutual authentication between users and servers.

6. Security and Performance Comparisons

This section discusses the security and performance analysis of the proposed protocol. Security analysis is mainly conducted through a comparison with other proposed protocols in the resistance of some common attacks, and performance analysis is mainly performed through a comparison with the time and communication costs of other protocols.

6.1. Security Comparisons. In this section, the protocol proposed in this study is compared with recent related protocols. Owing to the development of different types of attack technology and methods, previous protocols are now incapable of resisting some common attacks. At present, the common network attacks include A1: privileged-internal attack, A2: offline password-guessing attack, A3: replay attack, A4: perfect forward secrecy, A5: known session-specific temporary information attacks, and A6: user impersonation attacks. The comparison results are presented in Table 2. A "Yes" means that the protocol can resist the attack, whereas a "No" means that it cannot.

While the other related protocols each fail in some of the security attacks mentioned above, our proposed protocol can resist all the attacks, making our proposed protocol more secure and reliable.

6.2. Performance Comparisons. To better analyze the performance of this protocol, we compared it with a previous protocol. To obtain more convincing results, we analyzed the protocol using the same tools and under the same conditions and used the data provided by Rana et al. [19] The results show that different protocols have different execution times in the same execution environment. The time required for the connection operation and the noncollision hash function was 0.00014 ms and 0.00089 ms, respectively. The time required for the exception and encryption and decryption operations was extremely small, and so, it was not calculated. In addition, the number of bits required for the user name, password, arbitrary number, and integer was 160; the number of bits required for the private key and public key of the server was 256; the number of bits required for encryption and decryption was 512; and, the number of bits required for the exclusive or operation and noncollision hash function was 160 and 256, respectively. The symbols for each encryption operation are as follows:

$T_{\|}$: time required for connection operation

$T_{\oplus}$: time required for XOR operation

$T_{\text{Enc/Dec}}$: time required for encryption/decryption

T_h: time required for hash operation

First, we compared the communication cost of our proposed protocol with that of previous protocols. In particular, our protocol was compared with those proposed by Rana et al. [19], Kaul and Awasthi [18], Khan et al. [31], Chang et al. [17], and Kumari et al. [16]. The communication overhead of our protocol is 3136 bits, whereas that of the protocols proposed by Rana et al. [19], Kaul and Awasthi [18], Khan et al. [31], Chang et al. [17], and Kumari et al. [16] are 3296, 2668, 3744, 2336, and 3296 bits,

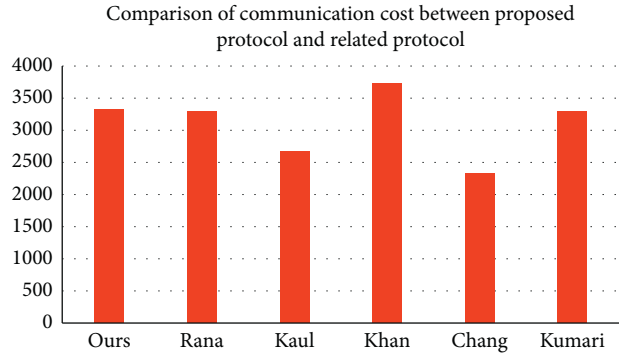


Figure 4: Communication cost.

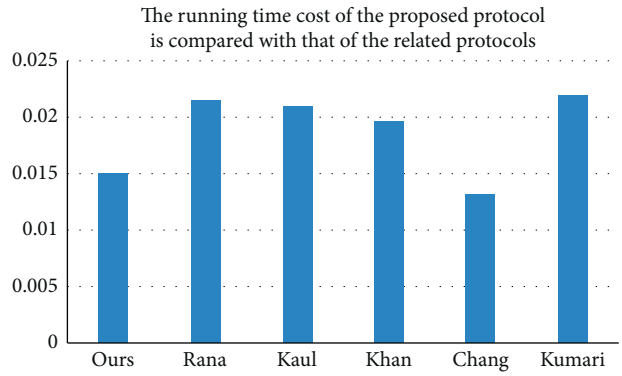


Figure 5: Running time.

Table 3: Proposed protocol comparison with related protocols.

	Running cost (ms)	Communication cost (bits)
Ours	$14T_h + 31T_\oplus + 19T_\parallel + 1T_{(\text{Enc/Dec})}$	3136
Rana et al. [19]	$20T_h + 29T_\oplus + 27T_\parallel + 3T_{(\text{Enc/Dec})}$	3296
Kaul and Awasthi [18]	$20T_h + 28T_\oplus + 23T_\parallel$	2668
Khan et al. [31]	$15T_h + 11T_\oplus + 45T_\parallel + 4T_{(\text{Enc/Dec})}$	3744
Chang et al. [17]	$12T_h + 7T_\oplus + 18T_\parallel$	2336
Kumari et al. [16]	$19T_h + 18T_\oplus + 36T_\parallel$	3296

respectively. As shown in Figure 4, the communication cost of our protocol is lower than that of Rana et al. and Khan et al., but slightly higher than that of Kaul and Awasthi [18]. Although the communication cost of Chang et al. is small, the protocol proposed by them cannot effectively resist privilege internal attacks, offline password guessing attacks, and replay attacks.

Next, we compare the running time cost of our proposed protocol with those of the three protocols mentioned above. The operating cost of our protocol is 0.01512 ms, whereas that of the protocols proposed by Rana et al. [19], Kaul and Awasthi [18], Khan et al. [31], Chang et al. [17], and Kumari et al. [16] are 0.0215 ms, 0.021 ms, 0.01965 ms, 0.01318 ms, and 0.02191 ms, respectively. As shown in Figure 5, the running time of our proposed protocol is shorter than that of the four protocols mentioned above. Although the time consumption of the protocol proposed by us is a little higher than that proposed by Chang et al., the protocol proposed by Chang et al. has the problem of security. It can be said that our protocol has better performance than the ones mentioned above.

Through the analysis of Tables 2 and 3, our protocol is slightly higher than Kaul and Awasthi's [18] protocol in terms of communication cost, but Kaul and Awasthi's [18] protocol cannot resist user simulation attacks. Because our proposed protocol can more effectively resist various security attacks, our protocol is more applicable in future works.

7. Conclusions

In this study, we analyzed the next generation Internet of Things remote protocol proposed by Rana et al., and found that their protocol cannot resist all kinds of security attacks as they claim. Specifically, we found that their protocols are vulnerable to offline password-guessing attacks and

privileged-insider attacks. To solve these problems, we introduced a three-factor security protocol utilizing biological information. In addition, we proved the security and reliability of the protocol through BAN logic and ROR analysis. Finally, we compared the proposed protocol with the previous related protocols and found that our protocol is better in terms of both communication cost and time cost. Therefore, our proposed protocol is more applicable and referential for the development of the future work.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] E. K. Wang, C.-M. Chen, M. M. Hassan, and A. Almogren, "A deep learning based medical image segmentation technique in internet-of-medical-things domain," *Future Generation Computer Systems*, vol. 108, pp. 135–144, 2020.

[2] C.-T. Li, C.-C. Lee, C.-Y. Weng, and C.-M. Chen, "Towards secure authenticating of cache in the reader for RFID-based IoT systems," *Peer-to-Peer Networking and Applications*, vol. 11, no. 1, pp. 198–208, 2018.

[3] H. Xiong, Y. Zhao, Y. Hou et al., "Heterogeneous signcryption with equality test for IIoT environment," *IEEE Internet of Things Journal*, 2020.

[4] T. Y. Wu, T. Wang, Y. Q. Lee, W. Zheng, S. Kumari, and S. Kumar, "Improved authenticated key agreement scheme for fog-driven IoT healthcare system," *Security and Communication Networks*, vol. 2021, Article ID 6658041, 16 pages, 2021.

[5] T.-Y. Wu, Z. Lee, L. Yang, C.-M. Chen, and R. Tso, "A provably secure authentication and key exchange protocol in vehicular ad hoc networks," *Security and Communication Networks*, vol. 2021, Article ID 9944460, 17 pages, 2021.

[6] P. Wang, C. M. Chen, S. Kumari, M. Shojafar, R. Tafazolli, and Y. N. Liu, "HDMA: hybrid D2D message authentication scheme for 5G-enabled VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, 2020.

[7] Z. Ali, A. Ghani, I. Khan, S. A. Chaudhry, S. H. Islam, and D. Giri, "A robust authentication and access control protocol for securing wireless healthcare sensor networks," *Journal of Information Security and Applications*, vol. 52, Article ID 102502, 2020.

[8] Q. Jiang, J. Ma, C. Yang, X. Ma, J. Shen, and S. A. Chaudhry, "Efficient end-to-end authentication protocol for wearable health monitoring systems," *Computers & Electrical Engineering*, vol. 63, pp. 182–195, 2017.

[9] M. F. Ayub, K. Mahmood, S. Kumari, and A. K. Sangaiah, "Lightweight authentication protocol for e-health clouds in IoT-based applications through 5G technology," *Digital Communications and Networks*, vol. 7, no. 2, pp. 235–244, 2021.

[10] A. T. Khan, S. Li, and X. Cao, "Control framework for cooperative robots in smart home using bio-inspired neural network," *Measurement*, vol. 167, Article ID 108253, 2021.

[11] X. Chen, A. Li, X. e. Zeng, W. Guo, and G. Huang, "Runtime model based approach to IoT application development," *Frontiers of Computer Science*, vol. 9, no. 4, pp. 540–553, 2015.

[12] C. M. Chen, L. Chen, Y. Huang, S. Kumar, and J. M. T. Wu, "Lightweight authentication protocol in edge-based smart grid environment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–18, 2021.

[13] R. Vinoth, L. J. Deborah, P. Vijayakumar, and N. Kumar, "Secure multifactor Authenticated key agreement scheme for industrial IoT," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3801–3811, 2020.

[14] Y. Yu, L. Hu, and J. Chu, "A secure authentication and key agreement scheme for IoT-based cloud computing environment," *Symmetry*, vol. 12, no. 1, p. 150, 2020.

[15] M. Soni and D. K. Singh, "LAKA: lightweight Authentication and key agreement protocol for Internet of things based wireless body area network," *Wireless Personal Communications*, vol. 1–18, 2021.

[16] S. Kumari, M. K. Khan, and X. Li, "An improved remote user authentication scheme with key agreement," *Computers & Electrical Engineering*, vol. 40, no. 6, pp. 1997–2012, 2014.

[17] Y. F. Chang, W. L. Tai, and H. C. Chang, "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 3430–3440, 2014.

[18] S. D. Kaul and A. K. Awasthi, "Security enhancement of an improved remote user authentication scheme with key agreement," *Wireless Personal Communications*, vol. 89, no. 2, pp. 621–637, 2016.

[19] M. Rana, A. Shafiq, I. Altaf et al., "A secure and lightweight authentication scheme for next generation IoT infrastructure," *Computer Communications*, vol. 165, pp. 85–96, 2021.

[20] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.

[21] D. Wang, D. He, P. Wang, and C. H. Chu, "Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2014.

[22] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2016.

[23] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 5–27, 2011.

[24] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[25] D. Kumar, H. S. Grover, D. Kaur, A. Verma, K. K. Saini, and B. Kumar, "An efficient anonymous user authentication and key agreement protocol for wireless sensor networks," *International Journal of Communication Systems*, vol. 34, no. 5, Article ID e4724, 2021.

[26] S. Kumari and K. Renuka, "Design of a password authentication and key agreement scheme to access e-healthcare services," *Wireless Personal Communications*, vol. 117, pp. 1–19, 2019.

[27] D. Kang, H. Lee, Y. Lee, and D. Won, "Lightweight user authentication scheme for roaming service in GLOMONET

with privacy preserving," *PLoS One*, vol. 16, no. 2, Article ID e0247441, 2021.

[28] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[29] K. Xue, P. Hong, and C. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 195–206, 2014.

[30] I.-C. Lin, M.-S. Hwang, and L.-H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.

[31] A. A. Khan, V. Kumar, M. Ahmad, and S. Rana, "LAKAF: lightweight authentication and key agreement framework for smart grid network," *Journal of Systems Architecture*, vol. 116, Article ID 102053, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5533843, 11 pages
https://doi.org/10.1155/2021/5533843

Review Article

Internet of Things Security: Challenges and Key Issues

Mourade Azrour,[1] **Jamal Mabrouki**,[2] **Azidine Guezzaz**,[3] **and Ambrina Kanwal**[4]

[1]*Computer Sciences Department, Faculty of Sciences and Techniques, Moulay Ismail University, Errachidia, Morocco*
[2]*Laboratory of Spectroscopy, Molecular Modeling, Materials, Nanomaterial, Water and Environment, CERNE2D, Mohammed V University, Faculty of Science, Rabat, Morocco*
[3]*Department of Computer Science and Mathematics, High School of Technology, Cadi Ayyad University, Essaouira 44000, Morocco*
[4]*Computer Science Department, Bahria University (Islamabad Campus), Islamabad, Pakistan*

Correspondence should be addressed to Mourade Azrour; mo.azrour@umi.ac.ma

Received 30 January 2021; Accepted 30 August 2021; Published 15 September 2021

Academic Editor: Habib Ullah Khan

Internet of Things (IoT) refers to a vast network that provides an interconnection between various objects and intelligent devices. The three important components of IoT are sensing, processing, and transmission of data. Nowadays, the new IoT technology is used in many different sectors, including the domestic, healthcare, telecommunications, environment, industry, construction, water management, and energy. IoT technology, involving the usage of embedded devices, differs from computers, laptops, and mobile devices. Due to exchanging personal data generated by sensors and the possibility of combining both real and virtual worlds, security is becoming crucial for IoT systems. Furthermore, IoT requires lightweight encryption techniques. Therefore, the goal of this paper is to identify the security challenges and key issues that are likely to arise in the IoT environment in order to guide authentication techniques to achieve a secure IoT service.

1. Introduction

In recent years, technology sector has known a real evolution. Furthermore, it has become an indispensable tool in our everyday life. Among these recent technologies, the Internet of Things (IoT) has been improved continuously and has attracted more and more people. This growth has positively impacted many sectors, including social security, agriculture, education, water management, house security, smart grid, and so on. Therefore, the number of connected devices is increasing day after day. According to Strategy Analytics, the connected objects will reach more than 38 billion by the end of 2025 and 50 billion by 2030 [1].

IoT is a new technology that allows the implementation of systems interconnecting several objects, either in the physical or virtual world [2, 3]. In fact, the evolution of the Internet began with the creation of a simple computer network linking personal computers and then moved on to client-server architecture networks, World Wide Web, e-mail, file sharing, etc. Subsequently, it now reaches a wide area network interconnecting billions of intelligent objects, which were embedded in sophisticated systems. Their operation is based on sensors and actuators designed for monitoring, controlling, and interacting with the physical environment where they exist.

Despite many advantages, IoT has three main problems that are data collection, data transmission, and data security. To collect data, many sensing tools have been introduced and adapted to the IoT devices. For transferring collected data, various protocols have been developed and adapted in order to enable to the IoT devices to connect to existed networks and exchange data. However, for the last one, it does not give the attention that it merits. Consequently, many classic and recent security issues are closely related to the IoT as well as authentication, data security, authorization, etc. Indeed, a weakness in authentication can lead to numerous attacks, including replay attack, Denning–Sacco attack, denial of service attack, password guessing attack, etc.

On the other hand, the authentication of IoT devices throughout heterogonous and interconnected protocols is a great challenge. Moreover, these protocols should take into account issues related to limitation of IoT devices as well as energy consumption, small memory size, and low processing capability [4–33].

In the literature review, previous studies [34–45] have surveyed the security of IoT technology. However, our study reveals some security challenges and issues of IoT. Consequently, the focus of this review paper is to categorize the security tasks and topics that are encountered in the IoT environment. Hence, we provide here a short guidance to researchers to accomplish secure IoT services like authentication, access control, and so on.

The remainder of this paper is organized as follows. In Section 2, IoT architecture is detailed. Section 3 is reserved for discussing IoT security issues. IoT security requirements are presented in Section 4. In Section 5, we compare some authentication approaches applied in IoT authentication environment. Finally, conclusions are given in Section 6.

2. IoT Architecture

The concept "Internet of Things" may be defined as a standard that refers to a large network connecting various sensors, actuators, and microcontrollers introduced in distinct objects. A large number of interconnected equipment such as smartphone, industrial machines, computers, vehicles, medical tools, irrigation system, TVs, or refrigerators can be part of the IoT [46]. Furthermore, IoT is a rather recent design that stands out from its antecedents, including all traditional, mobile, and sensor-based Internet networks. IoT includes a very large number of hybrid terminals. Since the majority of these devices can be connected to the Internet, they generally support common web techniques, including HTTP, JSON, XML, etc. One of the strengths of this technology is that it is well supported and can therefore be adapted to different existing infrastructures. Furthermore, some new protocols are especially considered for IoT, for example, CoAP and MQTT are alternatives to HTTP and 6LoWPAN is also an alternative of IPv4/IPv6.

Due to non-standardization of IoT, there are various architectures that are different [47]. However, we focus here on two known ones that are three- and five-layer architectures. As illustrated in Figure 1, the three-layer architecture consists of three layers including perception, networking, and application layers. The role of each layer is described in the following.

(i) The perception layer is the first layer of IoT architecture. It is connected to the physical world for sensing and collecting data from their environment. This layer consists of sensors and actuators to measure some values such as temperature, pH, light, gas, and so on, and to detect some functionality such as location and motion.

(ii) The network layer is the second layer; its role is to connect to various smart devices, gateways, and servers. It is responsible for transferring the captured values to other IoT network components. For these reasons, IoT uses several kinds of communication protocols and norms such as 4G/5G, Wi-Fi, ZigBee, Bluetooth, 6LoWPAN, WiMAX, and so on [48].

(iii) The application layer can offer the specific service requested by user. For instance, this application can provide doctors some health parameters of patients. This layer determines which applications can be installed, such as smart environment [49–52], smart homes [53–55], and water monitoring [56, 57].

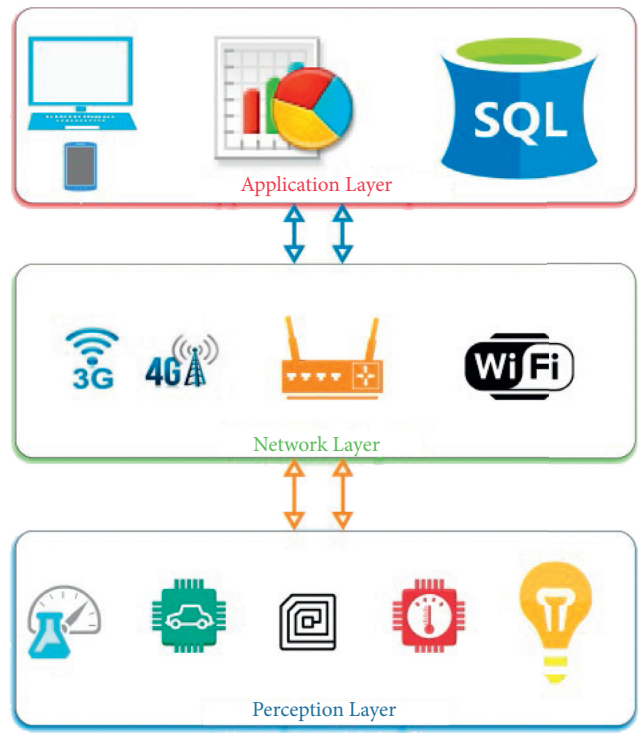


Figure 1: The three-layer architecture.

On the other hand, the five-layer architecture includes processing and business layers in addition to the three previous ones. As depicted in Figure 2, the five layers are perception, transport, processing, application, and business layers. The responsibilities of perception, transport, and application layers are identical to the similar layers in three-layer architecture. The roles of the addition layers are detailed as follows:

(i) The processing layer is also recognized as the middleware layer. It is responsible for controlling, analyzing, processing, and storing received data. It can make decisions according to the processing data without human intervention. This layer benefits from existing solutions including cloud computing, big data, and databases.

(ii) The business layer has a responsibility to manage the whole IoT systems [47]. So, its role is to control applications, business, and profit models. Furthermore, the users' privacy can be managed by this layer.

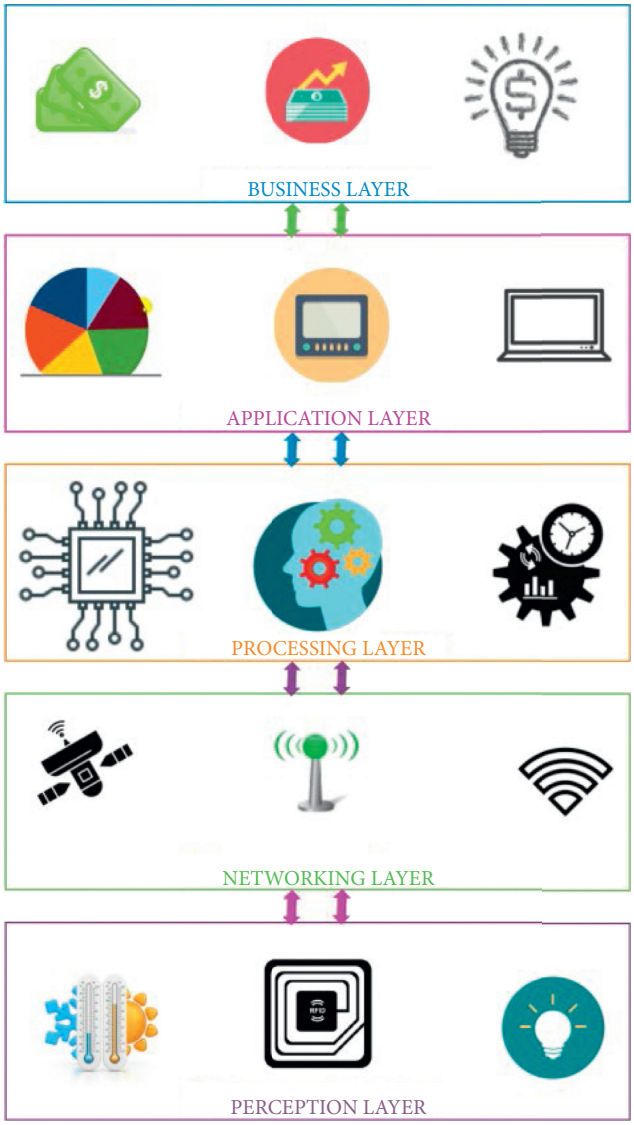


Figure 2: The five-layer architecture.

3. Security Issues in IoT

3.1. DOS. Denial of service (DOS) is a security attack that aims to prevent legitimate user and entity to have an authorized access to network resources. It is considered as the most popular and dominant attack. Generally, attackers can use flooding attack to exhaust system's resources including memory, CPU, and bandwidth [58–63]. Thus, he either prevents the system to provide service or he makes it ineffective. In this attack, pirates can use numerous skills such as sending unwanted packets or flooding network with multiple messages. Therefore, legitimate users are prevented from taking advantage of services.

3.2. Replay Attack. Replay attack is among old attacks on communication network, especially on authentication and key exchanging protocols. It allows the pirate to capture and store a fragment or the whole of captured session in a legitimate traffic [64, 65]. After gaining the trust in a public network, the attacker either sends the captured message to the entity that has participated in origin session or to another different destination [66]. Therefore, in IoT networks, replay attack is measured as a security weakness in which particular data are stored without any authorization before been sent back to the receiver. The goal of this attack is to trap the person in an unauthorized operation [67]. For example, in a smart home system, a temperature sensor is used to detect the temperature and then the measured values are sent to system controller. Based on these values, the system can run or stop the air conditioner to adapt the air temperature as desired by the personnel. However, if an attacker has pirated the sensor's temperature, he can save the day's values and send them at night. As result, the air conditioner will not be functioning normally.

To deal with replay attack, current solutions use three main mechanisms including timestamp, nonce, and response-challenge. The first one is the mechanism that helps to detect replay attack by checking the freshness of received message. Nonetheless, it is hard to assure time synchronization between IoT objects [68]. The second mechanism is the nonce, which is a series of random digits. However, the problem of this mechanism is that the node has no sufficient memory for keeping the list of received nonces. The last mechanism is the challenge-response. It has as objective to verify that the other party can resolve some challenges. But this technique necessitates that the two entities have a preshared secret.

3.3. Password Guessing Attack. Due to the importance of password in authentication process and its large adoption by numerous authentication protocols, pirates have invented various attacks to get the correct one. Hence, the most used attack is password guessing. Particularly, this attack can be executed either online or offline. In this attack, an attacker eavesdrops on the communication between two entities during authentication phase to get some useful values. Then, attacker must guess all probable passwords to succeed in the authentication [60, 69–75].

3.4. Spoofing Attack. In the network security context, spoofing attack is a situation when an unauthorized entity produces falsified parameter [76]. The goal of this attack is to make servers believe that the attacker is an authorized entity [62]. So, the pirate gains the trust of the authority. For example, in smart health, the pirate can send fake information to authentication server. So, if he performed the authentication phase successfully, he can request victim's sensor and then get the secret health information about this victim [38, 77–79].

3.5. Insider Attack. In cyber security field, insider attack occurs when a legitimate entity that has an authorized access tries to harm the system. The action of authorized entity can be either intentional or accidental [80–84]. In both cases, the system is considered vulnerable and we should find out the solution in the short term. According to [85], more than 57%

of confidential business data are targeted by insider attack. On the other hand, the study [86] confirms that more than 60% of existing attacks have been completed by insider.

4. Required Security Services for IoT

After debating various security attacks applied by attackers, this section mentions some security services. Thus, the objective of this section is to discuss the security requirements for IoT devices. As illustrated in Table 1, IoT solutions must come with some basic security services including authorization, authentication, confidentiality, availability, integrity, and non-repudiation.

4.1. Confidentiality. Generally, confidentiality can be defined as the capability and aptitude to prevent an unauthorized user to access private data. Therefore, it promises and guarantees that the personal information is only consulted, edited, or removed by authorized entity [38]. Particularly, in the Internet of Things network, confidentiality is one of the significant security services. However, the confidentiality is the most attacked service [87]. For example, viruses, spywares, and Trojans are considered as malware applications that attack the confidentiality of the user's private data. They can interact with system as executable codes or scripts with the aim to have an unauthorized access [88].

In an IoT context, for warranting and assuring the confidentiality of personal information captured by sensors and for preventing them from being discovered by the third party, the encryption algorithms and cryptographic methods can be used [89]. Therefore, all transmitted data between two devices must be encrypted. As a result, nobody can understand the message except legitimate entities [90].

4.2. Availability. An alternative required security service of IoT is the availability of resources to the legitimate entities independent of where and when they exist. Availability denotes that the resources and information must be easily reached by the legitimate user when he wants [91]. Moreover, in the IoT architecture, the sensor is available if it can communicate the sensed values in real time.

Likewise, the availability of an actuator means that it can execute user received commands immediately without any remarkable delay.

The availability of some particular resources could be interrupted as consequences of usage of dissimilar data transmission channel, networks, and protocols [46]. On the other hand, for damaging the availability, attackers may use three main malicious attacks including denial of service (DOS) attack, flooding attack, or black hole attack. For the first one, it is probably practiced in the availability situation. Pirates can use the simple denial of service (DOS) attack or distributed denial of service (DDOS) attack that necessitates the collaboration between various resources. For the flooding attack, the attacker can flood the networks by unwanted messages and commands for exhausting device resources. This attack not only targets bandwidth but also decreases CPU and memory capabilities. So, the device will not be reached or the communication will be slow [92].

Table 1: Security requirements for IoT basic layers.

Security services	IoT layers		
	Perception	Networking	Application
Authentication	✓	✓	✓
Authorization	✓	✓	✓
Confidentiality	✓		
Availability	✓	✓	
Integrity	✓	✓	✓
Non-repudiation		✓	

In order to guarantee the availability of appropriate resources, we can select distributed approach for operating the system and use numerous platforms which simplify the incorporation of various systems remotely [76].

4.3. Authentication. Authentication service is considered the biggest challenge in the IoT network. It includes verification of identity. On the one hand, in the authentication procedure, the devices must be able to check the validity and legitimacy of remote use in a public network. On the other hand, authentication prevents unauthorized person to take part in a private secured communication [38]. Previous authentication schemes are based on single factor that is a simple password. However, these schemes have to face various issues related to the password. First of all, users can easily forget the password. Secondly, users may have weak password. Finally, attackers are able to guess the correct password, either using exhaustive research attack or dictionary attack. Accordingly, password-based authentication is not enough to promise security. In our days, authentication schemes based on smart card offer multifactor authentication [4–9]. Typically, the system requires two factors including a valid smart card and correct preshared secret. Even so, it comprises the use of biometric print.

Due to the important position of authentication mechanism in the Internet of Things security, we have reserved the two following sections for discussing various techniques used for authentication in IoT and for studying some proposed IoT authentication schemes.

4.4. Authorization. With the growth of number of connected objects to the Internet network, authorization is becoming a critical issue in the IoT system. In fact, it refers to the security service responsible for determining user right and privileges (read, write, or delete). It identifies also the access control rules to allow or deny permissions to the IoT devices. Thus, the challenge is to prevent users with limited privileges to get additional ones to have an unauthorized access to devices and their data [93–97].

4.5. Integrity. Integrity means that the message was not reformed by an unauthorized entity in the transmission session. So, it guarantees that the receiver has received

exactly what the source has sent. The main objective is to stop an unauthorized object doing illegal modification.

For sustaining the safety of smart devices in IoT network, the system should guarantee data integrity. Therefore, neither unauthorized objects nor user access should be granted. Besides, the cryptography and encryption mechanisms can be applied when the transmitted data are very important [37]. For instance, the authors of [98] suggested the usage of HMAC-SHA 256 algorithm for reassuring data integrity.

4.6. Non-Repudiation. Non-repudiation is one of the security aspects, which insures that communication members have ability to send or receive information in its integrality [99]. In addition, it makes confident that the transfer of data or identifications between two IoT objects is undeniable [100]. Non-repudiation guarantees to a source node to send its data, as well as to a receiving node to confirm that the received data are matching with data's source [34].

5. IoT Authentication Techniques

Due to the ability of IoT to access to all users' information, the user's private life must be protected against the malicious attacks. Furthermore, the devices should not be accessed by unauthorized users. So, it is necessary to check the user's identity before getting the authorization. Hence, the verification of user's identity can be done in many ways. Nevertheless, the most frequently used is authentication system, which is based on the prior sharing secrets, keys, or passwords. Consequently, in this section, we review the techniques that are applied for reinforcing the authentication in IoT environment.

5.1. One Time Password Authentication. One time password (OTP) which is also called dynamic password is a password that is valid for authentication in one transaction. In the literature survey, various OTP authentication protocols are proposed for securing the communication in IoT environment. These protocols are founded based on various mechanisms such as time synchronization, hash factions (MD5, SHA1, and SHA256), and cryptography RSA. Besides, they are all based on the OTP algorithm created by Lamport [101–104]. Unfortunately, these protocols are vulnerable against some attacks as described in [105–108].

On the other hand, for reinforcing the OTP authentication, Lee and Kim [109] proposed in 2013 an insider attack-resistant OTP scheme based on bilinear maps. However, it needs complex computation. Based on this problem, Shivraj et al. [110] proposed a robust OTP scheme for IoT. The proposed protocol uses the principles of lightweight identity-based elliptic curve cryptography and Lamport's OTP algorithm.

5.2. ECC-Based Mutual Authentication. Generally, IoT devices have a limited resources. Besides, the communication between sensors, actuators, objects, and nodes must be in real time. For these reasons, it is indispensable to propose a lightweight authentication protocol for IoT. Accordingly, Azrour et al. [71] proposed an efficient authentication scheme for IoT. This protocol is based on elliptic curve cryptography (ECC) which is measured better than the traditional RSA encryption algorithm. Furthermore, in addition, various authentication protocol based on ECC are proposed in [111–115]. Elliptic curve cryptography is considered more efficient and more secure especially for systems with limited memory and processing capabilities.

5.3. ID- and Password-Based Authentication. ID-based authentication is an approach for distinguishing authorized entities from illegal ones. According to ID, the user is either allowed or denied to access the resource. User ID refers to all attributes that can characterize one user form another, for instance, username, e-mail, phone number, IP address, etc. In IoT environment, numerous protocols are proposed [74, 116–118] based on this technique. However, this method is generally adopted in the server/client authentication architecture. In view of that, a server is required in IoT environment for storing user's ID and secret in server's database.

On the other hand, the usage of ID-based authentication approach has some issues that are detailed in following lines. Firstly, how user's data are stored in server? Is the server capable to protect them against stolen verifier attack and insider attack? Secondly, users may forget their authentication parameters. Therefore, they cannot perform the next authentication. In this case, it is not suitable to save personal ID in an electronic device (laptop, tablet, and smartphone), even if it is not connected to public network. Thirdly, the transmission of user ID in public network is another challenge. In this situation, the hash functions or cryptography algorithm are recommended.

5.4. Certificate-Based Authentication. For addressing problems of ID- and password-based authentication, an alternative approach was proposed [119]. This technique is called certificate-based authentication. Certificate-based authentication has been commonly adopted by multiple applications. For example, in order to verify user's identity in banking application, Hiltgen et al. [120] proposed a new certificate-based authentication scheme. This approach has been also used in IoT environment [120–124]. Although certificate-based authentication provides more security, device certificate processing and used algorithms necessitate a high processing resource, which is not always available in IoT devices. As a result, this approach is not suitable for IoT objects [125].

5.5. Blockchain. Blockchain is a particular sort of database. It is different from a traditional database because of the specific way in which it stores data. Blockchains save data in a series of blocks that are then linked to each other. In recent years, different authors have taken advantage of this recent technology to propose authentication protocol for IoT [22, 31–33, 126, 127]. The sustainability and verification of the data stored in the blockchain provide

Table 2: Classification of some IoT authentication schemes.

Protocol	Proposed for securing		Method used			
	IoT	WSN	Encryption algorithm	Random number	Hash function	Others
[128]	—	✓	—	—	✓	—
[129]	✓	—	—	—	—	Time synchronization
[110]	✓	—	ECC	—	—	Lamport's OTP algorithm
[109]	—	—	—	✓	—	Zero-knowledge proof
[104]	—	—	AES-based MAC	—	—	—
[130]	✓	✓	ECC	—	✓	—
[131]	✓	✓	ECC	—	✓	Smart card
[132]	✓	—	ECC	✓	✓	—
[133]	✓	—	—	—	✓	—
[134]	✓	—	AES	—	—	—
[83]	✓	—	Symmetric encryption	—	—	—
[15]	✓	—	—	✓	✓	Fuzzy extractor mechanism
[20]	✓	—	ECC	✓	✓	Challenge-response
[135]	✓	—	Symmetric encryption	✓	✓	Blockchain machine learning
[136]	✓	—	ECC	✓	✓	-

ECC: elliptic curve cryptography; AES: Advanced Encryption Standard; OTP: one time password; WSN: wireless sensor network.

Table 3: Advantages and limitations in some IoT-based authentication schemes.

Protocol	Advantages	Limitations
[113]	Is lightweight	Uses only hash function
[114]	Can detect man-in-the-middle attacks	Uses certificates that need an important space in memory
[90]	Can be implemented in real-time IoT networks Based on two-factor authentication	Vulnerable
[89]	Can deal against insider attack based on bilinear maps	Needs complex computation
[84]	Surpasses HOTP	Is heavyweight Not efficient for IoT devices
[115]	Offers mutual authentication	Vulnerable against some attacks
[116]	Guarantees authentication and session key exchange	Does not cover all IoT service requirements
[74]	Can be used with cloud servers	Cannot resist all attacks
[117]	Very lightweight	Based only on one hash function
[118]	Lightweight mutual authentication	Operates only in CoAP-based IoT environment
[119]	Can be used for authentication protocol for IoT-based RFID systems	The running time of protocol is not very fast

the confidence to use accurately recorded data in the future and at the same time provide transparency, anonymity, and traceability.

Multiple and different authentication methods are used in the IoT environment. As demonstrated in Table 2, the majority of proposed IoT authentication protocols are based on encryption cryptography. In this situation, two types of cryptography are used. The first type is asymmetric encryption algorithm such as ECC, while the second one is symmetric encryption algorithm like AES. Furthermore, the hash functions are utilized in some authentication for hashing essential parameters. Finally, the random numbers are also adopted in certain protocol as they can be used to ensure the freshness of messages.

On the other hand, the advantages and limitations of some selected IoT authentication protocols are depicted in Table 3. As we can notice, the protocol is considered effective only if it is lightweight as well as fulfils all security requirements. To sum up, we can conclude that the running time and processing time are important due to the limitation capability of IoT devices.

6. Conclusions

Internet of Things has a significant role in the rapid development that recent technology has known recently. These technologies have made the exchange of data easier. However, the security of user's data should not be ignored. Accordingly, the study performed in this paper is mainly focused on the security of IoT technology. Hence, as we have mentioned before, IoT suffers from several attacks, namely, DOS, password guessing, replay, and insider attacks. Authentication is the first security services that IoT has to satisfy, so we have detailed the authentication approaches adopted for IoT. The most techniques used for rienforcing the authentication are one time password, ECC-based mutual authetication, ID-based authentication, certificate-based authentication, and blockchain. After comparing recent authentication protocols, we have concluded that the majority of them is based on encryption cryptography.

Finally, in our future work, we will try to enhance the security of IoT environment by proposing secure and efficient IoT authentication schemes.

Conflicts of Interest

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

References

[1] "Strategy analytics: internet of things now numbers 22 billion devices but where is the revenue? strategy analytics online newsroom." https://news.strategyanalytics.com/press-release/iot-ecosystem/strategy-analytics-internet-things-now-numbers-22-billion-devices-where (accessed Feb. 23, 2020).

[2] K. K. Patel, S. M. Patel, and P. Scholar, "Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges," *International journal of engineering science and computing*, vol. 6, 2016.

[3] O. Vermesan and P. Friess, *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*, River publishers, Denmark, 2013.

[4] M. Rana, A. Shafiq, I. Altaf et al., "A secure and lightweight authentication scheme for next generation IoT infrastructure," *Computer Communications*, vol. 165, pp. 85–96, 2021.

[5] S. A. Chaudhry, K. Yahya, F. Al-Turjman, and M.-H. Yang, "A secure and reliable device access control scheme for IoT based sensor cloud systems," *IEEE Access*, vol. 8, pp. 139244–139254, 2020.

[6] A. Irshad, S. A. Chaudhry, O. A. Alomari, K. Yahya, and N. Kumar, "A novel pairing-free lightweight authentication protocol for mobile cloud computing framework," *IEEE Syst. J.*vol. 99, pp. 1–9, 2020.

[7] Z. Ali, S. A. Chaudhry, K. Mahmood, S. Garg, Z. Lv, and Y. B. Zikria, "A clogging resistant secure authentication scheme for fog computing services," *Computer Networks*, vol. 185, Article ID 107731, 2021.

[8] S. A. Chaudhry, M. S. Farash, N. Kumar, and M. H. Alsharif, "PFLUA-DIoT: a pairing free lightweight and unlinkable user access control scheme for distributed IoT environments," *IEEE Syst. J.*vol. 99, pp. 1–8, 2020.

[9] A. Irshad, M. Usman, S. A. Chaudhry, H. Naqvi, and M. Shafiq, "A provably secure and efficient authenticated key agreement scheme for energy internet-based vehicle-to-grid technology framework," *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 4425–4435, 2020.

[10] A. Khan, V. Kumar, M. Ahmad, S. Rana, and D. Mishra, "PALK: password-based anonymous lightweight key agreement framework for smart grid," *International Journal of Electrical Power & Energy Systems*, vol. 121, Article ID 106121, 2020.

[11] S. A. Chaudhry, I. L. Kim, S. Rho, M. S. Farash, and T. Shon, "An improved anonymous authentication scheme for distributed mobile cloud computing services," *Cluster Computing*, vol. 22, no. 1, pp. 1595–1609, 2019.

[12] A. Irshad, M. Usman, S. A. Chaudhry, A. K. Bashir, A. Jolfaei, and G. Srivastava, "Fuzzy-in-the-Loop-Driven low-cost and secure biometric user access to server," *IEEE Transactions on Reliability*, vol. 70, no. 3, 2020.

[13] B. Alzahrani, A. Irshad, K. Alsubhi, and A. Albeshri, "A secure and efficient remote patient-monitoring authentication protocol for cloud-IoT," *International Journal of Communication Systems*, vol. 33, no. 11, Article ID e4423, 2019.

[14] S. Atiewi, A. Al-Rahayfeh, M. Almiani et al., "Scalable and secure big data IoT system based on multifactor authentication and lightweight cryptography," *IEEE Access*, vol. 8, pp. 113498–113511, 2020.

[15] M. Wazid, A. K. Das, V. Bhat K, and A. V. Vasilakos, "LAM-CIoT: lightweight authentication mechanism in cloud-based IoT environment," *Journal of Network and Computer Applications*, vol. 150, Article ID 102496, 2020.

[16] G. Sharma and S. Kalra, "Advanced lightweight multi-factor remote user authentication scheme for cloud-IoT applications," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1771–1794, 2020.

[17] M. Anuradha, T. Jayasankar, N. B. Prakash et al., "IoT enabled cancer prediction system to enhance the authentication and security using cloud computing," *Microprocessors and Microsystems*, vol. 80, Article ID 103301, 2021.

[18] M. Shahidinejad, G. Arani, A. Souri, M. Shojafar, and S. Kumari, "Light-edge: a lightweight authentication protocol for IoT devices in an edge-cloud environment," *IEEE Consumer Electronics Magazine*, vol. 2021, Article ID 3053543, 2021.

[19] M. Shahidinejad, G. Arani, A. Souri, M. Shojafar, and S. Kumari, "A technical report for light-edge: a lightweight authentication protocol for IoT devices in an edge-cloud environment," 2021, http://arxiv.org/abs/210106676.

[20] L. Loffi, C. M. Westphall, L. D. Grüdtner, and C. B. Westphall, "Mutual authentication with multi-factor in IoT-Fog-Cloud environment," *Journal of Network and Computer Applications*, vol. 176, Article ID 102932, 2021.

[21] B. D. Deebak and F. Al-Turjman, "Lightweight authentication for IoT/Cloud-based forensics in intelligent data computing," *Future Generation Computer Systems*, vol. 116, pp. 406–425, 2021.

[22] J. A. Alzubi, "Blockchain-based lamport merkle digital signature: authentication tool in IoT healthcare," *Computer Communications*, vol. 170, pp. 200–208, 2021.

[23] P. Kumar and L. Chouhan, "A privacy and session key based authentication scheme for medical IoT networks," *Computer Communications*, vol. 166, pp. 154–164, 2021.

[24] D. Deebak and F. Al-Turjman, "Secure-user sign-in authentication for IoT-based eHealth systems," *Complex & Intelligent Systems*, pp. 1–21, 2021.

[25] H. Luo, C. Wang, H. Luo, F. Zhang, F. Lin, and G. Xu, "G2F: a secure user authentication for rapid smart home IoT management," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10884–10895, 2021.

[26] P. Kumar and L. Chouhan, "A secure authentication scheme for IoT application in smart home," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 420–438, 2021.

[27] B. Alemu, R. Kumar, D. Sinwar, and G. Raghuwanshi, "Fingerprint based authentication architecture for accessing multiple cloud computing services using single user credential in IOT environments," *Journal of Physics: Conference Series*, vol. 1714, no. 1, Article ID 012016, 2021.

[28] M. I. Ahmed and G. Kannan, "Cloud-based remote RFID authentication for security of smart internet of things applications," *Journal of Information and Knowledge Management*, vol. 20, Article ID 2140004, 2021.

[29] M. Torabi and A. Shahidinejad, "A mutual authentication protocol for IoT users in cloud environment," *Electron Cyber Defense*, vol. 9, 2021.

[30] M. B. Mu'azu, "SIMP-REAUTH: a simple multilevel real user remote authentication scheme for mobile cloud computing," in *Proceedings of the Information and Communication Technology and Applications: Third International Conference, ICTA 2020*, November 2020.

[31] C. M. S. Ferreira, C. T. B. Garrocho, R. A. R. Oliveira, J. S. Silva, and C. F. M. d. C. Cavalcanti, "IoT registration and authentication in smart city applications with blockchain," *Sensors*, vol. 21, no. 4, 1323 pages, 2021.

[32] U. Narayanan, V. Paul, and S. Joseph, "Decentralized blockchain based authentication for secure data sharing in Cloud-IoT," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–19, 2021.

[33] H. Honar Pajooh, M. Rashid, F. Alam, and S. Demidenko, "Multi-layer blockchain-based security architecture for internet of things," *Sensors*, vol. 21, no. 3, 772 pages, 2021.

[34] D. Airehrour, J. Gutierrez, and S. K. Ray, "Secure routing for internet of things: a survey," *Journal of Network and Computer Applications*, vol. 66, pp. 198–213, 2016.

[35] M. Ammar, G. Russello, and B. Crispo, "Internet of things: a survey on the security of IoT frameworks," *Journal of Information Security and Applications*, vol. 38, pp. 8–27, 2018.

[36] Y. Atwady and M. Hammoudeh, "A survey on authentication techniques for the internet of things," in *Proceedings of the International Conference on Future Networks and Distributed Systems*, Cambridge, United Kingdom, July 2017.

[37] S. Hong, "Authentication techniques in the internet of things environment: a survey," *International Journal of Security and Networks*, vol. 21, no. 3, pp. 462–470, 2019.

[38] S. Panchiwala and M. Shah, "A comprehensive study on critical security issues and challenges of the IoT world," *Journal of Digital Information Management*, vol. 2, no. 4, pp. 257–278, 2020.

[39] P. P. Ray, "A survey on internet of things architectures," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, 2018.

[40] M. Heydari, A. Mylonas, V. H. F. Tafreshi, E. Benkhelifa, and S. Singh, "Known unknowns: indeterminacy in authentication in IoT," *Future Generation Computer Systems*, vol. 111, pp. 278–287, 2020.

[41] F. H. Al-Naji and R. Zagrouba, "A survey on continuous authentication methods in Internet of Things environment," *Computer Communications*, vol. 163, pp. 109–133, 2020.

[42] R. Yugha and S. Chithra, "A survey on technologies and security protocols: reference for future generation IoT," *Journal of Network and Computer Applications*, vol. 169, Article ID 102763, 2020.

[43] M. Mehta and K. Patel, "A review for IOT authentication - current research trends and open challenges," *Materials Today: Proceedings*, Article ID S2214785320384960, 2020.

[44] N. Yousefnezhad, A. Malhi, and K. Främling, "Security in product lifecycle of IoT devices: a survey," *Journal of Network and Computer Applications*, vol. 171, Article ID 102779, 2020.

[45] K. Sha, T. A. Yang, W. Wei, and S. Davari, "A survey of edge computing-based designs for IoT security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 195–202, 2020.

[46] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of things is a revolutionary approach for future technology enhancement: a review," *Journal of Big Data*, vol. 6, no. 1, 111 pages, 2019.

[47] M. Burhan, R. Rehman, B. Khan, and B.-S. Kim, "IoT elements, layered architectures and security issues: a comprehensive survey," *Sensors*, vol. 18, no. 9, 2796 pages, 2018.

[48] H. F. Atlam, R. J. Walters, and G. B. Wills, "Internet of things: state-of-the-art, challenges, applications, and open issues," *International Journal of Intelligent Computing Research*, vol. 9, no. 3, pp. 928–938, 2018.

[49] J. Mabrouki, M. Azrour, G. Fattah, D. Dhiba, and S. E. Hajjaji, "Intelligent monitoring system for biogas detection based on the internet of things: mohammedia, Morocco city landfill case," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 10–17, 2021.

[50] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, "IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.

[51] P. Visconti, N. I. Giannoccaro, R. d. Fazio, S. Strazzella, and D. Cafagna, "IoT-oriented software platform applied to sensors-based farming facility with smartphone farmer app," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1095–1105, Article ID 3, 2020.

[52] H. Andrianto, S. Suhardi, and A. Faizal, "Performance evaluation of low-cost IoT based chlorophyll meter," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 956–963, Article ID 3, 2020.

[53] M. Alilou, B. Tousi, and H. Shayeghi, "Home energy management in a residential smart micro grid under stochastic penetration of solar panels and electric vehicles," *Solar Energy*, vol. 212, pp. 6–18, 2020.

[54] M. S. Aliero, K. N. Qureshi, M. F. Pasha, and G. Jeon, "Smart home energy management systems in internet of things networks for green cities demands and services," *Environmental Technology & Innovation*, vol. 22, Article ID 101443, 2021.

[55] H. Kim, H. Choi, H. Kang, J. An, S. Yeom, and T. Hong, "A systematic review of the smart energy conservation system: from smart homes to sustainable smart cities," *Renewable and Sustainable Energy Reviews*, vol. 140, Article ID 110755, 2021.

[56] J. Mabrouki, M. Azrour, Y. Farhaoui, and S. El Hajjaji, "Intelligent system for monitoring and detecting water quality," in *Big Data And Networks Technologies*, Y. Farhaoui, Ed., vol. 81, pp. 172–182, Springer International Publishing, Cham, 2020.

[57] J. Mabrouki, M. Azrour, and S. El, "Use of internet of things for monitoring and evaluation water's quality: comparative study," *International Journal of Cloud Computing*, 2021, In press.

[58] S. Prabhakar, "Network security in digitalization: attacks and defence," *International Journal of Research in Computer Applications and Robotics*, vol. 5, no. 5, pp. 46–52, 2017.

[59] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT Security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[60] H. C. Hasan, F. N. Yusof, and M. Daud, "Comparison of authentication methods in internet of things technology," *International Journal of Computer and Systems Engineering*, vol. 12, no. 3, pp. 231–234, 2018.

[61] A. Ahmed, R. Latif, S. Latif, H. Abbas, and F. A. Khan, "Malicious insiders attack in IoT based multi-cloud e-healthcare environment: a systematic literature review," *Multimedia Tools and Applications*, vol. 77, no. 17, pp. 21947–21965, 2018.

[62] K. C. Archana and N. Harini, "Mitigation of spoofing attacks on IOT home networks," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 1S, pp. 240–245, 2019.

[63] Y. Javed, A. S. Khan, A. Qahar, and J. Abdullah, "Preventing DoS attacks in IoT using AES," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 3–11, pp. 3–11, 2017.

[64] H. C. A. van Tilborg and S. Jajodia, *Encyclopedia of Cryptography and Security*, Springer US, Boston, MA, 2011.

[65] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2027–2051, 2016.

[66] S. Behrooz and S. Marsh, "A trust-based framework for information sharing between mobile health care applications," *Trust Management X*, in *Proceedings of the IFIP International Conference on Trust Management*, pp. 79–95, Darmstadt, Germany, July 2016.

[67] P. Rughoobur and L. Nagowah, "A lightweight replay attack detection framework for battery depended IoT devices designed for healthcare," in *Proceedings of the 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS)*, pp. 811–817, Dubai, United Arab Emirates, December 2017.

[68] Y. Feng, W. Wang, Y. Weng, and H. Zhang, "A replay-attack resistant authentication scheme for the internet of things," in *Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 541–547, Guangzhou, China, July 2017.

[69] M. Azrour, Y. Farhaoui, and M. Ouanan, "Cryptanalysis of farash et al.'s SIP authentication protocol," *International Journal of Dynamical Systems and Differential Equations*, vol. 8, no. 1/2, 2018.

[70] M. Azrour, Y. Farhaoui, and A. Guezzaz, "Experimental validation of new SIP authentication protocol," in *Big Data And Networks Technologies*, Y. Farhaoui, Ed., vol. 81, pp. 1–11, Springer International Publishing, Cham, 2020.

[71] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for internet of things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.

[72] P. K. Roy, K. Parai, and A. Hasnat, "User authentication with session key interchange for wireless sensor network," in *Methodologies and Application Issues of Contemporary Computing Framework*, J. K. Mandal, S. Mukhopadhyay, P. Dutta, and K. Dasgupta, Eds., Springer Singapore, Singapore, pp. 153–165, 2018.

[73] J. Moon, T. Song, D. Lee, Y. Lee, and D. Won, "Cryptanalysis of chaos-based 2-party key agreement protocol with provable security," in *Advances in Human Factors in Cybersecurity*, D. Nicholson, Ed., vol. 593, pp. 72–77, Springer International Publishing, Cham, 2018.

[74] K. Park, S. Lee, Y. Park, and Y. Park, "An ID-based remote user authentication scheme in IoT," *Journal of Korea Multimedia Society*, vol. 18, no. 12, pp. 1483–1491, 2015.

[75] J. Ryu, H. Lee, H. Kim, and D. Won, "Improvement of Wu et al.'s three-factor user authentication scheme for wireless sensor networks," , 2018.

[76] K. Somasundaram and K. Selvam, "Iot - attacks and challenges," *International Journal of Engineering and Technical Research (IJETR)*, vol. 8, no. 9, pp. 9–12, 2018.

[77] R. Z. Naeem, S. Bashir, M. F. Amjad, H. Abbas, and H. Afzal, "Fog computing in internet of things: practical applications and future directions," *Peer-to-Peer Networking and Applications*, vol. 12, no. 5, pp. 1236–1262, 2019.

[78] T. Sherasiya, H. Upadhyay, and H. B. Patel, "A survey: intrusion detection system for internet of things," *International Journal of Computer Science and Engineering (IJCSE)*, vol. 5, 2006.

[79] M. Nikooghadam and H. Amintoosi, "Secure communication in CloudIoT through design of a lightweight authentication and session key agreement scheme," *International Journal of Communication Systems*, Article ID e4332, 2020.

[80] F. Kammüller, J. R. C. Nurse, and C. W. Probst, "Attack tree analysis for insider threats on the IoT using isabelle," in *Human Aspects of Information Security, Privacy, and Trust*, T. Tryfonas, Ed., vol. 9750, Springer International Publishing, Lecture Notes in Computer Science, pp. 234–246, 2016.

[81] G. Sharma and S. Kalra, "Identity based secure authentication scheme based on quantum key distribution for cloud computing," *Peer-to-Peer Networking and Applications*, vol. 11, no. 2, pp. 220–234, 2018.

[82] S.-Q. Cao, Q. Sun, and L.-L. Cao, "Security analysis and enhancements of a remote user authentication scheme," *IOP Conference Series: Materials Science and Engineering*, vol. 719, Article ID 012004, 2019.

[83] K. Mansoor, A. Ghani, S. A. Chaudhry, S. Shamshirband, S. A. K. Ghayyur, and E. Salwana, "Securing IoT based RFID systems: a robust authentication protocol using symmetric cryptography," *Mathematics & Computer Science*, vol. 19, Article ID 4752, 2019.

[84] S. Holger, "Insider threat report," *Cybersecurity Insiders*, Accessed: Aug. 13, 2020. [Online, 2019.

[85] I. B. M. X-Force® Research, *Cyber Security Intelligence Index*, Accessed: Jun. 02, 2017. [Online, 2016.

[86] N. Ye, Y. Zhu, R.-c. Wang, R. Malekian, and L. Qiao-min, "An efficient authentication and access control scheme for perception layer of internet of things," *Applied Mathematics & Information Sciences*, vol. 8, no. 4, pp. 1617–1624, 2014.

[87] R. Canzanese, M. Kam, and S. Mancoridis, "Toward an automatic, online behavioral malware classification system," in *Proceedings of the IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pp. 111–120, Philadelphia, PA, USA, September 2013.

[88] Y. Javed, A. S. Khan, A. Qahar, and J. Abdullah, "Preventing DoS attacks in IoT using AES," *J. Telecommun. Electron. Comput. Eng. JTEC*, vol. 9, no. 3–11, pp. 3–11, 2017.

[89] G. Sharma and S. Kalra, "A lightweight multi-factor secure smart card based remote user authentication scheme for cloud-IoT applications," *Journal of Information Security and Applications*, vol. 42, pp. 95–106, 2018.

[90] M. Azrour, Y. Farhaoui, and M. Ouanan, "A server spoofing attack on Zhang et al. SIP authentication protocol," *Int. J. Tomogr. Simulation*^{TM}, vol. 30, no. 3, pp. 47–58, 2017.

[91] M. Domb, "Smart home systems based on internet of things," *in Internet of Things (IoT) for Automated and Smart Applications*, IntechOpen, London, UK, 2019.

[92] T. Shah and S. Venkatesan, "Authentication of IoT device and IoT server using secure vaults," in *Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 819–824, New York, NY, USA, August 2018.

[93] O. Garcia-Morchon, S. Kumar, and M. Sethi, "Internet of Things (IoT) security: state of the art and challenges," *RFC Editor*, vol. RFC8576, 2019.

[94] M. Wu, J. Chen, and R. Wang, "An enhanced anonymous password-based authenticated key agreement scheme with formal proof," *IJ Network Security*, vol. 19, no. 5, pp. 785–793, 2017.

[95] A. Drissi and A. Asimi, "Behavioral and security study of the OHFGC hash function," *ReCALL*, vol. 1, no. 0, 2017.

[96] C.-W. Liu, C.-Y. Tsai, and M.-S. Hwang, "Cryptanalysis of an efficient and secure smart card based password authentication scheme," in *Recent Developments in Intelligent Systems and Interactive Applications*, F. Xhafa, S. Patnaik, and Z. Yu, Eds., vol. 541, pp. 188–193, Springer International Publishing, Cham, 2017.

[97] A. Jebrane, A. Toumanari, N. Meddah, and M. Bousseta, "A new efficient authenticated and key agreement scheme for sip using digital signature algorithm on elliptic curves," *Journal of Telecommunications and Information Technology*, vol. 2, pp. 62–68, 2017.

[98] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for internet of things applications: challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2018.

[99] V. Umadevi, R. Chezhian, and Z. U. Khan, "Security requirements in mobile ad-hoc networks," *Int J Adv Res Comput Commun*, vol. 1, no. 2, 2012.

[100] M. G. Samaila, M. Neto, D. A. B. Fernandes, M. M. Freire, and P. R. M. Inácio, "Challenges of securing internet of things devices: a survey," *Security and Privacy*, vol. 1, no. 2, e20 pages, 2018.

[101] D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "Totp: time-based one-time password algorithm," *Internet Req. Comments*, 2011.

[102] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, "Hotp: an hmac-based one-time password algorithm," *Internet Soc. Netw. Work. Group RFC4226*, 2005.

[103] B. Hamdane, A. Serhrouchni, A. Montfaucon, and S. Guemara, "Using the hmac-based one-time password algorithm for tls authentication," in *Proceedings of the 2011 Conference on Network and Information Systems Security*, pp. 1–8, La Rochelle, France, May 2011.

[104] S.-D. Park, J.-C. Na, Y.-H. Kim, and D.-K. Kim, "Efficient OTP (one time password) generation using AES-based MAC," *J. Korea Multimed. Soc.*vol. 11, no. 6, pp. 845–851, 2008.

[105] P.-A. Fouque, G. Leurent, and P. Q. Nguyen, "Full key-recovery attacks on HMAC/NMAC-MD4 and NMAC-MD5," in *Proceedings of the Annual International Cryptology Conference*, pp. 13–30, Santa Barbara, CA, USA, August 2007.

[106] E. Lee, D. Chang, J. Kim, J. Sung, and S. Hong, "Second preimage attack on 3-pass HAVAL and partial key-recovery attacks on HMAC/NMAC-3-pass HAVAL," in *Proceedings of the International Workshop on Fast Software Encryption*, pp. 189–206, Lausanne, Switzerland, February 2008.

[107] J. Kim, A. Biryukov, B. Preneel, and S. Hong, "On the security of HMAC and NMAC based on HAVAL, MD4, MD5, SHA-0 and SHA-1 (extended abstract)," in *Proceedings of the International Conference on Security and Cryptography for Networks*, pp. 242–256, Lecture Notes in Computer Science, Maiori, Italy, September 2006.

[108] X. Wang, H. Yu, W. Wang, H. Zhang, and T. Zhan, "Cryptanalysis on hmac/nmac-md5 and md5-mac," *Advances in Cryptology - EUROCRYPT 2009*, in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 121–133, Zagreb, Croatia, April 2009.

[109] Y. Lee and H. Kim, "Insider attack-resistant otp (one-time password) based on bilinear maps," *International Journal of Computer and Communication Engineering*, vol. 2, no. 3, pp. 304–308, 2013.

[110] V. L. Shivraj, M. A. Rajan, M. Singh, and P. Balamuralidhar, "One time password authentication scheme based on elliptic curves for Internet of Things (IoT)," in *Proceedings of the 2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*, pp. 1–6, Riyadh, Saudi Arabia, February 2015.

[111] M. Safkhani, N. Bagheri, S. Kumari, H. Tavakoli, S. Kumar, and J. Chen, "RESEAP: an ECC-based authentication and key agreement scheme for IoT applications," *IEEE Access*, vol. 8, pp. 200851–200862, 2020.

[112] S. Chatterjee and S. G. Samaddar, "A robust lightweight ECC-based three-way authentication scheme for IoT in cloud," in *Smart Computing Paradigms: New Progresses And Challenges*, pp. 101–111, Springer, Singapore, 2020.

[113] B. Hammi, A. Fayad, R. Khatoun, S. Zeadally, and Y. Begriche, "A lightweight ECC-based authentication scheme for Internet of Things (IoT)," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3440–3450, 2020.

[114] A. Lohachab and Karambir, "ECC based inter-device authentication and authorization scheme using MQTT for IoT networks," *Journal of Information Security and Applications*, vol. 46, pp. 1–12, 2019.

[115] M. A. Khan, M. T. Quasim, N. S. Alghamdi, and M. Y. Khan, "A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data," *IEEE Access*, vol. 8, pp. 52018–52027, 2020.

[116] S. N. Swamy, D. Jadhav, and N. Kulkarni, "Security threats in the application layer in IOT applications," in *Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 477–480, Palladam, India, February 2017.

[117] W.-q. Jiang, Z.-q. Huang, Y.-x. Yang, J. Tian, and L. Li, "ID-based authentication scheme combined with identity-based encryption with fingerprint hashing," *The Journal of China Universities of Posts and Telecommunications*, vol. 15, no. 4, pp. 75–120, 2008.

[118] O. Salman, S. Abdallah, I. H. Elhajj, A. Chehab, and A. Kayssi, "Identity-based authentication scheme for the internet of things," in *Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC)*, pp. 1109–1111, Messina, Italy, June 2016.

[119] S. A. Nauroze, J. G. Hester, B. K. Tehrani et al., "Additively manufactured RF components and modules: toward empowering the birth of cost-efficient dense and ubiquitous IoT implementations," *Proceedings of the IEEE*, vol. 105, no. 4, pp. 702–722, 2017.

[120] A. Hiltgen, T. Kramp and T. Weigold, Secure internet banking authentication," *IEEE Security and Privacy Magazine*, vol. 4, no. 2, pp. 21–29, 2006.

[121] S. Malani, J. Srinivas, A. K. Das, K. Srinathan, and M. Jo, "Certificate-based anonymous device access control scheme for IoT environment," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9762–9773, 2019.

[122] J. Choi, J. Cho, H. Kim, and S. Hyun, "Towards secure and usable certificate-based authentication system using a secondary device for an industrial internet of things," *Applied Sciences*, vol. 10, no. 6, 1962 pages, 2020.

[123] Q. Zhang, K. Zhao, X. Kuang et al., "Multidomain security authentication for the Internet of things," *Concurrency and Computation: Practice and Experience*, Article ID e5777, 2020.

[124] I. Ullah, N. U. Amin, M. A. Khan, H. Khattak, and S. Kumari, "An efficient and provable secure certificate-based combined signature, encryption and signcryption scheme for Internet

of Things (IoT) in Mobile Health (M-Health) system," *Journal of Medical Systems*, vol. 45, no. 1, pp. 1–14, 2021.

[125] Shuang and Y. Zhou, "A study of autonomous method of IoT component," in *Proceedings of the 5th International Conference on New Trends in Information Science and Service Science*, vol. 2, pp. 294–298, Macao, China, October 2011.

[126] P. Bagga, A. K. Sutrala, A. K. Das, and P. Vijayakumar, "Blockchain-based batch authentication protocol for internet of vehicles," *Journal of Systems Architecture*, vol. 113, Article ID 101877, 2021.

[127] W. Meng, W. Li, S. Tug, and J. Tan, "Towards blockchain-enabled single character frequency-based exclusive signature matching in IoT-assisted smart cities," *Journal of Parallel and Distributed Computing*, vol. 144, pp. 268–277, 2020.

[128] C.-H. Ling, C.-C. Lee, C.-C. Yang, and M.-S. Hwang, "A secure and efficient one-time password authentication scheme for WSN," *International Journal on Network Security*, vol. 19, no. 2, pp. 177–181, 2017.

[129] C. Tae-Ho and J. Garam-Moe, "A method for detecting man-in-the-middle attacks using time synchronization one time password in interlock protocol based internet of things," *Journal of Applied and Physical Sciences*, vol. 2, no. 2, pp. 37–41, 2016.

[130] H.-L. Yeh, T.-H. Chen, P.-C. Liu, T.-H. Kim, and H.-W. Wei, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.

[131] A. Maurya and V. N. Sastry, "Fuzzy extractor and elliptic curve based efficient user authentication protocol for wireless sensor networks and internet of things," *Information*, vol. 8, no. 4, 136 pages, 2017.

[132] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, "A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers," *The Journal of Supercomputing*, vol. 74, no. 12, pp. 6428–6453, 2018.

[133] M. Bayat, M. Beheshti-Atashgah, M. Barari, and M. R. Aref, "Cryptanalysis and improvement of a user authentication scheme for internet of things using elliptic curve cryptography," *IJ Netw. Secur*, vol. 21, no. 6, pp. 897–911, 2019.

[134] M. A. Jan, P. Nanda, X. He, Z. Tan, and R. P. Liu, "A robust authentication scheme for observing resources in the internet of things environment," in *Proceedings of the 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 205–211, Washington, DC, USA, September 2014.

[135] H. Al-Naji and R. Zagrouba, "CAB-IoT: continuous authentication architecture based on blockchain for internet of things," *J. King Saud Univ.-Comput. Inf. Sci.*, 2020.

[136] S. Lu and X. Li, "Quantum-resistant lightweight authentication and key agreement protocol for fog-based microgrids," *IEEE Access*, vol. 9, pp. 27588–27600, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1230593, 8 pages
https://doi.org/10.1155/2021/1230593

Research Article

A Reliable Network Intrusion Detection Approach Using Decision Tree with Enhanced Data Quality

Azidine Guezzaz,[1] **Said Benkirane**,[1] **Mourade Azrour**,[2] **and Shahzada Khurram**[3]

[1]*Computer Sciences Department, Technology Higher School, Cadi Ayyad University, Essaouira, Morocco*
[2]*Computer Sciences Department, IDMS Team, Faculty of Sciences and Technics, Moulay Ismail University, Errachidia, Morocco*
[3]*Computer Department of Information Security, Faculty of Computing, Islamia University, Bahawalpur, Pakistan*

Correspondence should be addressed to Azidine Guezzaz; a.guzzaz@gmail.com

Received 2 July 2021; Revised 26 July 2021; Accepted 13 August 2021; Published 21 August 2021

Academic Editor: Azeem Irshad

Due to the recent advancements in the Internet of things (IoT) and cloud computing technologies and growing number of devices connected to the Internet, the security and privacy issues are important to be resolved and protect the data and computer network. To provide security, a real-time monitoring of the network data and resources is needed. Intrusion detection systems have been used to monitor, detect, and alert an intrusion event in real time. Recently, the intrusion detection systems (IDS) incorporate several machine learning (ML) techniques. One of the techniques is decision tree, which can take reliable network measures and make good decisions by increasing the detection rate and accuracy. In this paper, we propose a reliable network intrusion detection approach using decision tree with enhanced data quality. Specifically, network data preprocessing and entropy decision feature selection is carried out for enhancing the data quality and relevant training; then, a decision tree classifier is built for reliable intrusion detection. Experimental study on two datasets shows that the proposed model can reach robust results. Actually, our model achieves 99.42% and 98.80% accuracy with NSL-KDD and CICIDS2017 datasets, respectively. The novel approach gives many advantages compared to the other models in term of accuracy (ACC), detection rate (DR), and false alarm rate (FAR).

1. Introduction

The computer security threats are becoming quite challenging with the growing capabilities of the adversaries, influencing the reliability of data communication and networks. The recent advancements in cloud computing and IoT technologies enabled new attack vectors for the adversary and even more prone to attacks [1–3]. The IoT applications enable the attacks not only focusing stealing the data but can also impacting human lives. For example, a hacked home utility smart heater can be used to automatically increase the temperature and indirectly impact the human beings living in the home [4, 5]. Hence, the main goal of security is to provide integrity, confidentiality, and availability by implementing various security tools and policies that can protect data and detect attacks targeting the IoT. [4, 6]. An intrusion tries to violate one of security objectives and infects systems. Hence, many tools and methods, such as IDS, are developed to secure networks and systems from intrusions [7–9]. Thereby, intrusion detection is a set of techniques implemented to detect undesirable activities by classifying data activity into normal or intrusion [6, 8]. The intrusion detection techniques detect and stop intrusions from outside or within a monitored network.

For this reason, two fundamental detection approaches can be used. The first one is called misuse detection; it is based on a known attack signature to detect intrusion. The second one is named anomaly detection or behavioral detection, based on a deviation from a normal model [1, 8, 10]. The hybrid detection approaches combine advantages of both misuse and anomaly detection and aim to increase detection rate and accuracy of IDS [9, 11, 12]. A considerable distinction is made between network IDS (NIDS) and host IDS (HIDS)

[1, 8]. Formally, an IDS can be software or hardware which detects malicious traffic, makes accurately automatic decisions, and interrupts intrusions quickly in real time with an automatic response [6, 8].

Despite their efficiency, the IDS suffers from a number of limitations, such as real-time analysis and detection, generated alarm, and data quality, that can decrease detection rate and accuracy performances [6, 8]. Therefore, intrusion detection is still an effective and dynamic research field.

Recently, ML methods have been integrated to enhance intrusion detection and reinforce computer security. Numerous research contributions explore how to incorporate ML techniques in intrusion detection to obtain reliable IDS with accurate performances by enhancing data quality and training [13–20]. The decision tree is an induction algorithm which has been used for classification in many issues. It is based on splitting features and testing the value of each one. The splitting process continues until each branch can be labelled with just one classification [21, 22]. The decision tree is more than equivalent representation to the training set. Hence, it can be used to predict the values of other instances not in the training set. The decision tree is widely used as a mean of generating classification rules because of the existence of a simple but very powerful algorithm called Top-Down Induction of Decision Trees (TDIDT). It is guaranteed to give a decision tree that correctly corresponds to the data provided by two of the best known being ID3 and C4.5 [22].

On the other side, the data is not always obtained in a structured form. For relevant analysis, the unstructured data have to be preprocessed. This operation is an essential stage which performed to enhance data quality and make accurate decisions. Data quality techniques are implemented before training and classification process [17, 23, 24]. Besides, feature selection is a desirable process aiming to select the useful features to both reduce the computational cost of modelling and to improve the performance of the predictive model [13, 24].

In this paper, we propose a novel network intrusion detection approach based on the decision tree method to train and build a binary classifier model and make accurate decisions. The features' engineering techniques were used to improve the data quality. Experimental results on the NSL-KDD dataset and CICIDS2017 dataset demonstrate that our proposed approach gives good performances in terms of accuracy DR and FAR. Two main contributions have been validated in this research work. Firstly, we implement feature selection using entropy decision technique to improve data quality. Secondly, we build a classifier model based on decision tree algorithm to achieve effective network intrusion detection approach.

The remainder of this paper is organized as follows. Section 2 presents related work on intrusion detection, especially which integrated ML techniques to improve IDS performances. Section 3 describes in detail the proposed solutions for the novel approach. In Section 4, we discuss experimental results, performance of the proposed model, and its comparison with other models. Finally, the conclusion and future works presented in Section 4.

2. Related Works

During the last decade, a set of contributions of intrusion detection were adopted in [8, 10, 11, 17, 21, 25, 26] to ensure computer security objectives. The research in intrusion detection is oriented towards on automatic response to increase effectiveness and capability of IDS [6]. Therefore, to obtain reliable IDS, the false positive (FP) and false negative (FN) rate should be low, but also, true positive (TP) and true negative (TN) rate should be high. Furthermore, including ML techniques in intrusion detection becomes an excited research domain [13–20]. Hence, intrusion detection based on ML is a classification task aiming to detect intrusions using labelled data by building a classifier able to distinguish between normal and abnormal activity [11, 16, 21, 27, 28]. Several ML techniques, such as decision tree [21], random forest [29], nearest neighbour [30], Naïve Bayes [26, 27], support vector machine [17], fuzzy clustering [15], reinforcement based learning [19], and deep learning methods [1, 6, 14, 18, 25, 26, 31, 32] have been integrated to enhance IDS by discovering knowledge from intrusion detection datasets [9, 31, 33, 34]. For more improvements, a set of feature engineering techniques, such as feature selection, are made to enhance data quality. They allow a relevant data process used to train and build effective classifier [13, 17, 23, 25, 35, 36].

In 2018, Karami [37] proposed an anomaly-based intrusion detection system using the fuzzy SOM method. In 2020, Tabash et al. [26] proposed an intrusion detection model which integrated NB and DL technique. The model implemented genetic algorithm for a good feature selection. In 2015, Ghazali et al. [27] proposed a detection model for intrusive communication. This research work tests five classification techniques: SimpleCart, NB, BFTree, PART, and Ridor. The performances' measures on NSL-KDD dataset demonstrate ACC 96.7%, DR 95.5%, and FAR 4.7%. In 2017, Kevric et al. [28] proposed a combining classifier approach using tree algorithm for network intrusion detection. The model is evaluated on NSL-KDD dataset ACC 89.24%. In 2018, Hadi [29] proposed a model based on random forest algorithm for selecting a significant feature. The model was evaluated using NSL-KDD. The results of the proposed model are ACC 99.33%, DR 0.993% TP, and FAR 0.001% FP. In 2019, Gu et al. [17] proposed a model of an ensemble SVM-based intrusion detection with LMDRT transformation as an effective method to enhance data quality. The performances' results on CICIDS2017 dataset are ACC 93.64%, DR 97.56%, and FAR 20.28%. In 2020, Elmasry et al. [32] developed a DL model for network intrusion detection using a double PSO metaheuristic. The model is evaluated on CICIDS2017 dataset and gives ACC 92.92%, DR 92.38%, and FAR 3.24%. In 2019, Prasard et al. [36] proposed new IDS which works on subset of features by extracting significant features using the probabilistic method. The BRS method is implemented to categorize samples into normal, intermediary, and abnormal category based on the rough set. The model is trained and tested on CICIDS2017 dataset and demonstrates ACC 97.6%, DR 96.38%, and FAR 3.00%. In 2019, Ahmim et al. [21] proposed

a hybrid IDS model which combines the classifier model based on decision tree, REP tree, JRIP algorithm, and forest PA. The performances of the novel model are evaluated using CICIDS2017 dataset and presented ACC 96.66%, DR 94.475%, and FAR 4.47%.

From the state-of-the-art literature survey, it is proven that the learning methods and data quality are two useful tasks which determine the robustness of IDS [6, 17, 26–29, 32, 36, 37]. These research works implement much of techniques for a high quality of data by not only reducing and selecting features but also building improved classifiers to better categorize data activities.

3. Novel Network Intrusion Detection Approach

In this section, we describe our methodology and proposed solutions aiming to implement and validate the novel approach. By enhancing feature engineering and classification techniques, we obtained reliable IDS with accurate performances.

3.1. Our Proposed Model. As depicted in Figure 1, the proposed model consists of three main components including data quality component, building of classifier component, and intrusion detection deployment component. The details of those three components are given in the following.

Part 1: data quality process.

The main goal of this component is collecting and preprocessing the data. Hence, the system executes the process that can gather and accumulate necessary data from networks. Once the data are collected, a specific data preprocessing is performed on gathered network traffic. The data preprocessing portion evaluates the data and ignores the incompatible data types. Furthermore, the data is sanitized and the resulting data is saved. In addition, the data is transformed and the features of network dataset are finalized. We used the entropy decision technique to select the features.

Part 2: building of the classifier.

Once the first part is completed, the second one is started. Generally, the objective of second part, as it is clear in its name, is to build a classifier model. The input here is the transformed data obtained in the data quality process part. In the classifier building part, we can distinguish between two main phases: model training phase and model validation phase. In the first phase, three portions of data are used for training a decision tree classifier implemented in our proposed approach. Then, in the second phase, the rest of data are used to validate our model.

Part 3: network intrusion detection deployment.

After building of the classifier model, the third part comes for deploying the network intrusion detection. At this point, actual tests are necessary to improve the performance of reliable IDS. Hence, we are in aptitude to check its capacity to classify activities in normal or abnormal. So, based on the classification results, the IDS can made accurate.

3.2. Description of Proposed Solutions. As we mentioned above, the first step which is made by our approach is to collect and transform data with feature selection according to needs of analysis and detection. The data quality is an important and essential task to train and build an accurate intrusion detection model. Hence, this step aims to prepare data for analysis and make accurate decision. We start first with data transformation by applying feature selection using entropy decision on original traffic collected within network traffic to obtain a good training set. In fact, it is a critical step aiming to improve accuracy of our approach. It aims also to overcome training complexity by reducing analysed data and obtain a great model with best performances in terms of accuracy, detection rate, and real-time detection. A particular preprocessing is applied on collected network traffic before the analysis step. Data normalization is performed. For this, we suggest and implement a particular coding to enumerate feature values and establish a pattern of activities facilitating the distinction between the activities. The goal of the feature extraction is to reduce the number of features in collected data from networks. It aims to summarize most of the information contained in this original data by creating new features. The feature selection aims instead to choose the important existing features in the original data and discard less important ones. For this reason, we use entropy decision technique for feature selection. The implementation of components that constitute our approach is described in Figure 2.

We obtain a transformed data by implementing proposed data quality techniques, aiming to increase our approach accuracy. This allows training and validating of an effective intrusion detection model based on the decision tree to make relevant decisions in real time. Moreover, intrusion detection is considered as a classification task aiming to classify incoming traffic in normal activity or intrusion. Hence, the main objective of this part is to predict a binary value to validate the classifier able to answer question with a yes or a no. Thus, we encoded both classes in numerical variable: +1 for normal activity and −1 for intrusion. We remember that the number of features must be fixed in advance. For the validation step of our model, there are various strategies used to split the data into a training and test set. In this case, we use the efficient and recommended one, k-fold [1].

According to standard components of an IDS mentioned in [8, 29], our approach is constituted by four parts: data collection part, preprocessing part, decision-making part, and response part. The proposed approach focuses on the preprocessing part by improving data quality technique used to train and build an accurate classifier which is able to discover intrusions within traffic network. It focuses also on enhancing the decision-making part by integrating the decision-tree classifier. A set of research works have been made in [6, 13, 24] to improve others parts of IDS, such as

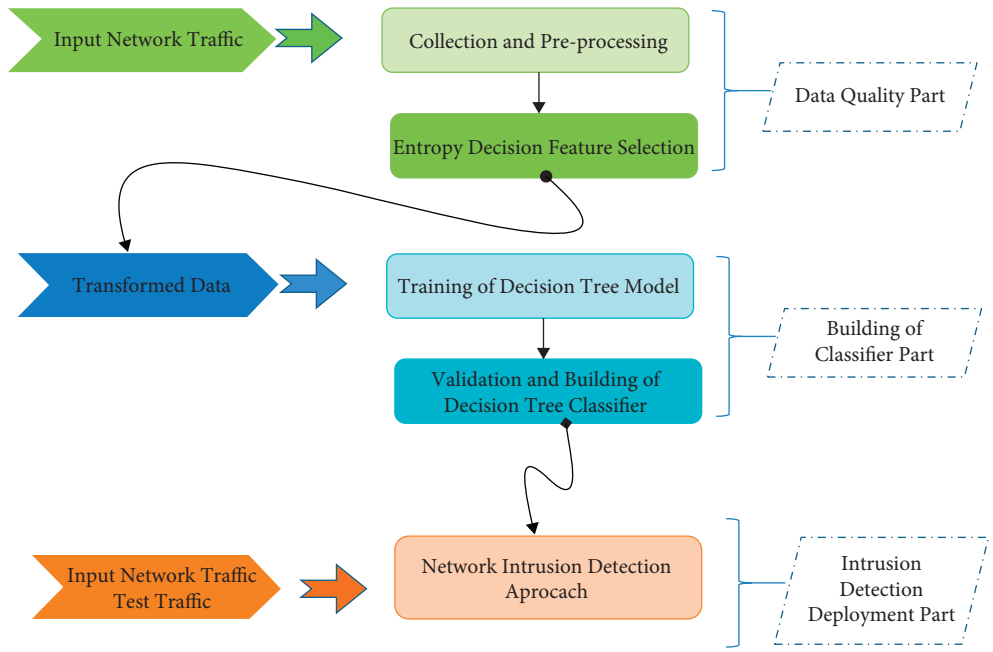


Figure 1: Proposed network intrusion detection model.

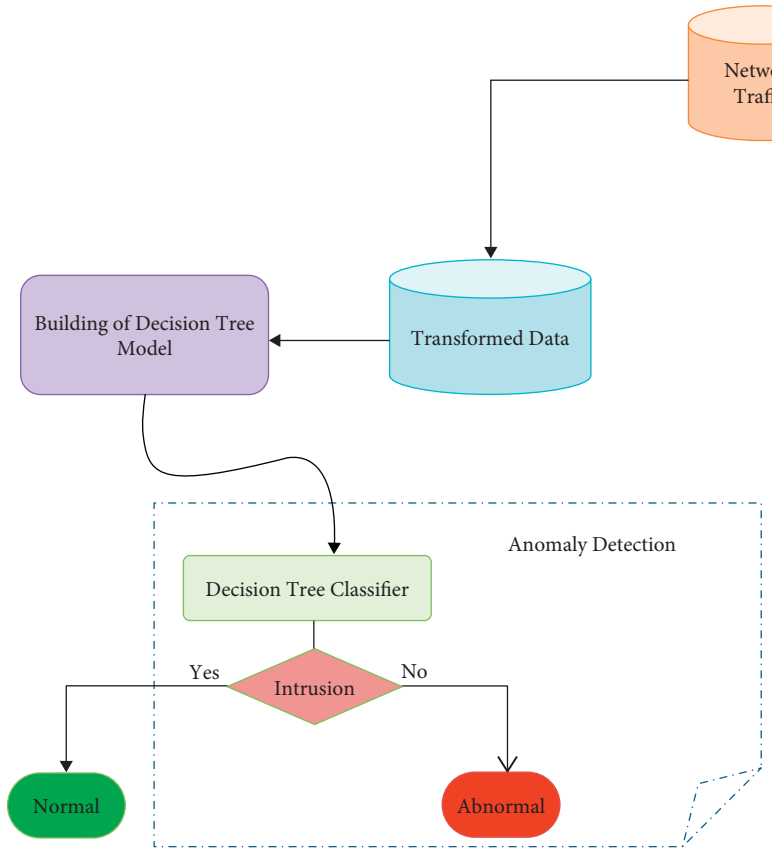


Figure 2: Procedure of validation and building classifier.

data collection, dimensionality reducing, and real-time response which are not taken into account in this research work.

4. Experimental Results and Discussion

4.1. Dataset Description. The assessment of datasets plays a vital role in validation of intrusion detection approaches. Therefore, for evaluating any IDS using ML techniques, one can select the desired dataset among a large number of appropriate and available datasets. For instance, numerous public datasets are available [9, 31, 33, 34] and can be used freely for evaluation proposed methods' capability. In our case, we have selected two types of datasets including NSL-KDD and CICIDS2017, which are used for training and performances' evaluation and validation of the proposed approach.

The NSL-KDD dataset was created from KDD cup 99 dataset [9, 27]. It contains 125,973 records of the training set and 22,544 for the test set. It has 22 training instances' attacks and 41 features in which 21 of them describe connection itself and 19 for nature of connection of the same host [33, 38]. The novelty and instances' volume of the NSL-KDD dataset make it very practical. On the contrary, the CICIDS2017 dataset was created from Canadian Institute for Cyber Security. It aims to overcome the limitations of the actual dataset and present an effective dataset for intrusion detection. It is a labelled dataset that comprises behavior and new malware attacks and is consisted of 8 files containing 2,830,743 instances. The CICIDS2017 dataset integrates 80 features' network flow captured at July 2017 from network traffic using CICFlowMeter tool [9].

Those two used datasets in this research work, NSL-KDD dataset and CICIDS2017 dataset, are available at [39, 40], respectively.

4.2. Experiments' Environment. The experimental setting of our research work is performed and evaluated on a computer with a Core-i7 2700K CPU@ 2.50 GHz and 32 GB of DDR3 running windows 7 professional 64 bits. The entropy feature selection and decision-tree model training are implemented using python version 3.8.0.

To validate our proposed intrusion detection model, we use the 10-fold cross-validation technique to obtain the training and test set. Hence, we split randomly full dataset into ten parts with the same size. Nine parts are used in the training and the last part in the test step. Finally, the performances of the model are presented by repeating this procedure ten times.

4.3. Data Transformation. In the implementation step, we propose to extract samples of dataset to avoid some drawbacks such as processing and big volume of data. The data extraction from each used dataset is given in Table 1.

Feature selection is a relevant technique included by our network intrusion detection approach. It is implemented and incorporated to select useful features for reliable detection and decision-making. For this, we implement entropy decision technique.

Table 1: Data extraction from NSL-KDD and CICIDS2017 datasets.

	Category	Original size	Extracted size
NSL-KDD dataset	Training	125,973	25,195
	Test	22,544	4,509
	Total	148,517	29,704
CICIDS2017 dataset	Benign	2,273,097	113,655
	Attack	557,646	27,883
	Total	2,830,743	141,538

The encoding step is performed to assign numeric values to categorical features for making relevant processing. To avoid undesirable influence problem of high weights, we normalize continuous features values. Equation (1) is used to find the new value. Hence, we make the values of each feature run from 0 to 1. If the lowest value of a given feature x is min and the highest value is max, we convert each value of x to

$$\frac{(\text{value}(x) - \min)}{(\max - \min)}. \tag{1}$$

Furthermore, all continuous features are in range [0, 1].

4.4. Metrics Evaluation and Discussion. The most obvious criterion to use for estimating the performances of a classifier is predictive accuracy. The proportion of a set of unseen instances that it correctly classifies. For numerical performances' evaluation of the proposed model, the following metrics are used.

These metric performances are not dependent on the size of the training and test set and can be really helpful in assessing the performance of the full model. Based on the confusion matrix (Table 2), the performances' metrics are calculated.

ACC is obtained from equation (2). It is the ratio of instances that are correctly predicted as normal or attack to the overall number of instances in the test set:

$$ACC = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{2}$$

DR is calculated using equation (3) and indicates the ratio of the number of instances that are correctly classified as attack to the total number of attack instances present in the test set:

$$DR = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3}$$

FAR is obtained from equation (4) and represents the ratio of instances which is categorized as attack to the overall number of instances of normal behavior:

$$FAR = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{4}$$

In this research work, we start with comparing detection assessment of our proposed model for novel approach and decision-tree model only. The results shown in Figures 3 and 4 demonstrate this comparison according to ACC, DR, and FAR on the NSL-KDD dataset and the CICIDS2017 dataset.

Table 2: Confusion matrix.

Actual class	Predicted class	
	Attack	Normal
Attack	TP	FN
Normal	FP	TN

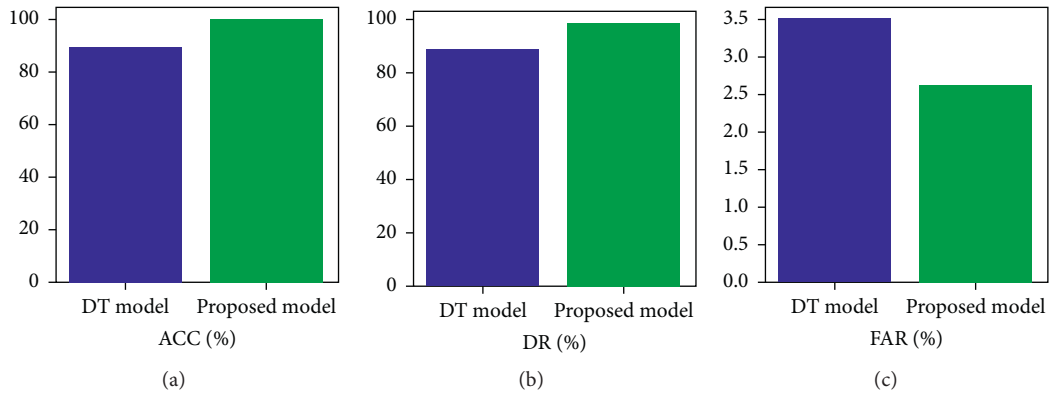


Figure 3: (a) ACC results of the DT model and our proposed model on the NSL-KDD dataset. (b) DR results. (c) FAR results.

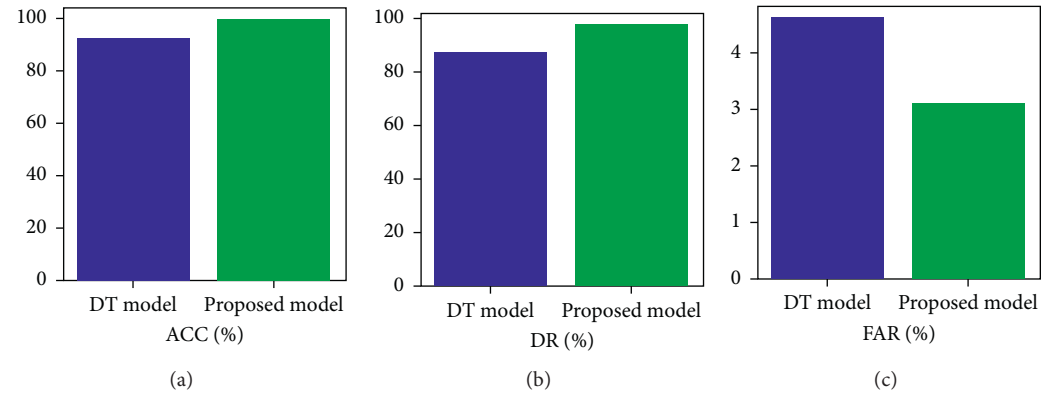


Figure 4: (a) ACC results of the DT model and our proposed model on the CICIDS2017 dataset. (b) DR results. (c) FAR results.

Figures 3(a) and 4(a) show that accuracy of the proposed model is specifically better than the model based on the decision tree only. Figures 3(b) and 4(b) demonstrate the DR of both IDS. It validates that the DR of the proposed IDS model is higher than the IDS based on the decision tree only on the NSL-KDD dataset and the CICIDS2017 dataset.

The results demonstrated above are summarized in Tables 3 and 4. They show that our proposed model can reach significant performances than the decision tree only. For the NSL-KDD dataset, the ACC of our proposed model achieves 99.42%, while the decision tree only exceeds 89%. In terms of DR and FAR, our proposed model obtains 98.2% and 2.64%, respectively, while the decision tree only presents DR 88.5% and FAR 3.5%. For the CICIDS2017 dataset, our proposed model indicates high performances in terms of ACC 98.8%, DR 97.3%, and FAR 3.10%. Besides, the decision tree only gives ACC 92%, DR 86.7%, and FAR 4.6%.

Table 3: Performances' metrics of the decision tree and the proposed model using the NSL-KDD dataset.

	ACC (%)	DR (%)	FAR (%)
Decision tree	89.00	88.50	3.50
Proposed approach	99.42	98.20	2.64

Table 4: Performances' metrics of the decision tree and the proposed model using the CICIDS2017 dataset.

	ACC (%)	DR (%)	FAR (%)
Decision tree	92.00	86.70	4.60
Proposed approach	98.80	97.30	3.10

The results obtained validate that our approach gives great detection capability in terms of ACC, DR, and FAR. Specifically, they demonstrate that the performances'

Table 5: Performances' comparison with other models on NSL-KDD.

	Method	Accuracy (%)	DR (%)	FAR (%)
Masdari and Khezri [12]	Five classification	96.70	95.50	4.70
Ahmim et al. [21]	Tree algorithm	89.24	—	—
Fang [16]	RF	99.33	0.993 TP	0.001FP
Proposed approach	DTE	99.42	98.20	2.64

Table 6: Performances' comparison with other models on CICIDS2017.

	Method	Accuracy (%)	DR (%)	FAR (%)
Chiba et al. [1]	DTRM	96.66	94.475	4.47
Alazzam et al. [13]	EnSVM	93.64	97.56	20.28
Ayo et al. [25]	BRS	97.96	96.38	3.00
Khraisat et al. [9]	DL	92.92	92.38	3.24
Proposed approach	DTE	98.80	97.30	3.10

metrics of our proposed model are higher *o* NSL-KDD dataset but low CICIDS2017 dataset. According to the evaluation performances, our proposed IDS model can reach great performances. The comparison with the model which uses the decision tree only indicates the effectiveness of our network intrusion detection approach.

Concretely, our proposed intrusion detection model is specified by high performances of ACC, DR, and FAR. Furthermore, we perform a comparison between our IDS and other recent intrusion detection approaches based on the NSL-KDD dataset and the CICIDS2017 dataset. Typically, the recent works that integrate ML techniques are tree algorithm, RF, DTRM, EnSVM, BRS, and DL. The comparison results are presented in Tables 5 and 6.

From the obtained results, we conclude that our proposed IDS approach is relevant, achieves important performances, and gives relevant training by implementing fast data quality techniques. Using the NSL-KDD dataset and the CICIDS2017 dataset, it is proven that our approach is reliable and reaches good results compared with other models. The novel approach can be integrated and used to secure various environments such as IoT environment and cloud computing.

5. Conclusion and Future Works

Intrusion detection is a set of enhanced techniques implemented to monitor systems and data to be more secure. In this paper, we present a reliable network intrusion detection approach based on decision-tree classifier and engineering feature techniques. According to heterogeneity of data, a preprocessing phase is setting up to increase detection rate and accuracy of IDS. Also, a feature selection technique based on the entropy decision-tree method is handled before building the model for high data quality. The validation of novel approach is achieved by proposed solutions that guarantee an efficient accuracy. The performances are evaluated on two datasets: NSL-KDD and CICIDS2017. Hence, the novel proposed network intrusion detection approach presents many advantages and provides high accuracy compared with other models. The future works will integrate other efficient ML techniques such as deep learning in various parts to empower detection rate and accuracy of our approach.

Data Availability

The assessments and experimental results, obtained using Anaconda 3 IDE, are available at https://sites.google.com/umi.ac.ma/azrour.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Z. Chiba, N. Abghour, K. Moussaid, A. El omri, and M. Rida, "Intelligent approach to build a deep neural network based IDS for cloud environment using combination of machine learning algorithms," *Computers & Security*, vol. 86, pp. 291–317, 2019.

[2] A. Irshad, S. A. Chaudhry, O. A. Alomari, K. Yahya, and N. Kumar, "A novel pairing-free lightweight authentication protocol for mobile cloud computing framework," *IEEE Systems Journal*, vol. 2020, Article ID 2998721, 2020.

[3] S. A. Chaudhry, I. L. Kim, S. Rho, M. S. Farash, and T. Shon, "An improved anonymous authentication scheme for distributed mobile cloud computing services," *Cluster Computing*, vol. 22, no. S1, pp. 1595–1609, 2019.

[4] A. Irshad, M. Usman, S. A. Chaudhry, H. Naqvi, and M. Shafiq, "A provably secure and efficient authenticated key agreement scheme for energy internet-based vehicle-to-grid technology framework," *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 4425–4435, 2020.

[5] S. A. Chaudhry, M. S. Farash, N. Kumar, and M. H. Alsharif, "PFLUA-DIoT: a pairing free lightweight and unlinkable user access control scheme for distributed IoT environments," *IEEE Systems Journal*, vol. 2020, Article ID 3036425, 2020.

[6] A. Guezzaz, Y. Asimi, M. Azrour, and A. Asimi, "Mathematical validation of proposed machine learning classifier for heterogeneous traffic and anomaly detection," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 18–24, 2021.

[7] G. Fernandes, J. J. P. C. Rodrigues, and L. F. Carvalho, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, pp. 447–489, 2019.

[8] A. Guezzaz, A. Asimi, Z. Tbatou, Y. Asimi, and Y. Sadqi, "A global intrusion detection system using pcapsocks sniffer and multilayer perceptron classifier," *International Journal on Network Security*, vol. 21, no. 3, pp. 438–450, 2019.

[9] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, 2019.

[10] S.-Y. Ji, B.-K. Jeong, S. Choi, and D. H. Jeong, "A multi-level intrusion detection method for abnormal network behaviors,"

Journal of Network and Computer Applications, vol. 62, pp. 9–17, 2016.

[11] Ü. Çavuşoğlu, "A new hybrid approach for intrusion detection using machine learning methods," *Applied Intelligence*, vol. 49, pp. 2735–2761, 2019.

[12] M. Masdari and H. Khezri, "A survey and taxonomy of the fuzzy signature-based intrusion detection systems," *Applied Soft Computing*, vol. 92, Article ID 106301, 2020.

[13] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer," *Expert Systems with Applications*, vol. 148, Article ID 113249, 2020.

[14] A. Aldweesh, A. Derhab, and Z. E. Ahmed, "Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues," *Knowledge-Based Systems*, vol. 189, Article ID 105124, 2020.

[15] M. Amini, J. Rezaeenour, and E. Hadavandi, "A neural network ensemble classifier for effective intrusion detection using fuzzy clustering and radial basis function networks," *The International Journal on Artificial Intelligence Tools*, vol. 25, no. 2, 2016.

[16] W. Fang, X. Tan, and D. Wilbur, "Application of intrusion detection technology in network safety based on machine learning," *Safety Science*, vol. 124, Article ID 104604, 2020.

[17] J. Gu, L. Wang, H. Wang, and S. Wang, "A novel approach to intrusion detection using SVM ensemble with feature augmentation," *Computers & Security*, vol. 86, pp. 53–62, 2019.

[18] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, pp. 386–396, 2020.

[19] K. Sethi, E. Sai Rupesh, R. Kumar, P. Bera, and Y. Venu Madhav, "A context-aware robust intrusion detection system: a reinforcement learning-based approach," *International Journal of Information Security*, vol. 19, no. 6, pp. 657–678, 2020.

[20] A. Sommer and V. Paxson, "Outside the closed world: on using machine learning for network intrusion detection,," in *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, pp. 305–316, Oakland, May 2010.

[21] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, "A novel hierarchical intrusion detection system based on decision tree and rules-based models,"pp. 228–233, https://ieeexplore.ieee.org/xpl/conhome/8790388/proceeding, Santorini, Greece, May 2019.

[22] C. Guo, Y. Ping, N. Liu, and S.-S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, 2016.

[23] K. Jeyakumar, T. Revathi, and S. Karpagam, "Intrusion detection using artificial neural networks with best set of features," *The International Arab Journal of Information Technology*, vol. 12, no. 6A, 2015.

[24] M. Rostami, K. Berahmand, E. Nasiri, and S. Forouzandeh, "Review of swarm intelligence-based feature selection methods," *Engineering Applications of Artificial Intelligence*, vol. 100, Article ID 104210, 2021.

[25] F. E. Ayo, S. O. Folorunso, A. A. Abayomi-Alli, A. O. Adekunle, J. B. Awotunde, and J. B. Awotunde, "Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection," *Information Security Journal: A Global Perspective*, vol. 29, no. 6, pp. 267–283, 2020.

[26] M. Tabash, M. Abd Allah, and B. Tawfik, "Intrusion detection model using naive bayes and deep learning technique," *The International Arab Journal of Information Technology*, vol. 17, no. 2, 2020.

[27] A. Ghazali, W. Nuaimy, A. Al-Atabi, and I. Jamaludin, "Comparison of classification models for Nsl-Kdd dataset for network anomaly detection," *Academic Journal of Science*, vol. 4, no. 1, pp. 199–206, 2015.

[28] J. Kevric, S. Jukic, and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Computing & Applications*, vol. 28, no. S1, pp. 1051–1058, 2017.

[29] A. Hadi, "Performance analysis of big data intrusion detection system over random forest algorithm," *International Journal of Applied Engineering Research*, vol. 13, no. 2, pp. 1520–1527, 2018.

[30] A. Topîrceanu and G. Grosseck, "Decision tree learning used for the classification of student archetypes in online courses," *Procedia Computer Science* in *Proceedings of the 21st International Conference on Knowledge Based and Intelligent Information and Engineering*, vol. 112, pp. 51–60, Marseille, France, September 2017.

[31] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, Article ID 102419, 2020.

[32] W. Elmasry, A. Akbulut, and A. H. Zaim, "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic," *Computer Networks*, vol. 168, Article ID 107042, 2020.

[33] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.

[34] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, pp. 108–116, Madeira, Portugal, January 2018.

[35] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

[36] M. Prasad, S. Tripathi, and K. Dahal, "An efficient feature selection based Bayesian and rough set approach for intrusion detection," *Applied Soft Computing*, vol. 2020, Article ID 105980, 2020.

[37] A. Karami, "An anomaly-based intrusion detection system in presence of benign outliers with visualization capabilities," *Expert Systems with Applications*, vol. 108, pp. 36–60, 2018.

[38] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 dataset," in *Proceedings of the Second 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, Ottawa, Canada, July 2009.

[39] https://web.archive.org/web/20150205070216/http://nsl.cs.unb.ca/NSL-KDD/ (2021, July 24).

[40] https://www.unb.ca/cic/datasets/ids-2017.html (2021, July 24).

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9965573, 30 pages
https://doi.org/10.1155/2021/9965573

Review Article

The Snowden Phone: A Comparative Survey of Secure Instant Messaging Mobile Applications

Christian Johansen,[1] Aulon Mujaj,[1] Hamed Arshad,[1] and Josef Noll[2]

[1]*Department of Informatics, University of Oslo, P.O. Box 1080 Blindern, 0316 Oslo, Norway*
[2]*Department of Technology Systems, University of Oslo, Postboks 70, 2027 Kjeller, Norway*

Correspondence should be addressed to Hamed Arshad; hamedar@ifi.uio.no

Received 12 March 2021; Accepted 24 June 2021; Published 10 July 2021

Academic Editor: Shehzad Ashraf Chaudhry

In recent years, it has come to attention that governments have been doing mass surveillance of personal communications without the consent of the citizens. As a consequence of these revelations, developers have begun releasing new protocols for end-to-end encrypted conversations, extending and making popular the old Off-the-Record protocol. New implementations of such end-to-end encrypted messaging protocols have appeared, and several popular chat applications have been updated to use such protocols. In this survey, we compare six existing applications for end-to-end encrypted instant messaging, namely, Signal, WhatsApp, Wire, Viber, Riot, and Telegram, most of them implementing one of the recent and popular protocols called Signal. We conduct five types of experiments on each of the six applications using the same hardware setup. During these experiments, we test 21 security and usability properties specially relevant for applications (not protocols). The results of our experiments demonstrate that the applications vary in terms of the usability and security properties they provide, and none of them are perfect. In consequence, we make 12 recommendations for improvement of either security, privacy, or usability, suitable for one or more of the tested applications.

1. Introduction

The trend to use mobile applications for communication has grown and become a standard method of communication between people. New messaging applications started to emerge and try to replace traditional SMS, but building them with security and privacy in mind was not important for the developers in the beginning. The popular messaging tools used in recent years did not support end-to-end encryption, only standard client-to-server encryption, which gives the service providers access to more private information than necessary. When Edward Snowden published the secret papers about NSA (https://en.wikipedia.org/wiki/Edward_Snowden#Global_surveillance_disclosures and two feature films on this topic are as follows: Oliver Stone's https://en.wikipedia.org/wiki/Snowden_(film) and Laura Poitras' https://en.wikipedia.org/wiki/Citizenfour), people finally understood that mass surveillance was an issue, and secure mobile messengers became more critical and popular. (Disclaimer: all the tests reported here were performed in the summer of 2017, and since applications in this area are very dynamic; some of the specific implementation recommendations and observations that we make may have already been treated by the developers. However, this work still should provide guidance for a new user on how to check which desired features are implemented by a specific application, even more so if the application is among the six surveyed here. More details for this paper can be found in the technical report [1] and the thesis of the second author [2].)

People are more prone to understand the privacy implications of mass surveillance [3]. Edward Snowden has sparked a heated debate throughout the world about individual privacy which is undermined by the mass surveillance that multiple countries have been doing for decades (https://en.wikipedia.org/wiki/List_of_government_mass_surveillance_projects). No

need to look further than the first quarter of 2017, when WikiLeaks (https://wikileaks.org/) leaked documents from the U.S. Central Intelligence Agency (CIA). The leak, codenamed "Vault 7" by WikiLeaks is the largest ever publication of confidential documents from the agency ("WikiLeaks Unveils "Vault 7": The Largest Ever Publication Of Confidential CIA Documents; Another Snowden Emerges," authored by Tyler Durden in the Zerohedge, March 2017, available at https://www.zerohedge.com/news/2017-03-07/wikileaks-hold-press-conference-vault-7-release-8am-eastern). The documents that leaked have information on how to get access to mobile phones or personal computers without the user's knowledge and how the CIA did mass surveillance.

Several companies started implementing secure messaging protocols and applications to counter the mass surveillance and offer an end-to-end encrypted messaging system that does not leak any information about the user's message content. However, the problem with new applications is their adoption. After a while, companies such as Google, Facebook, and Open Whisper Systems joined forces to implement protocols into already widely adopted applications such as WhatsApp, which has over one billion monthly active users (the statistics portal: "Number of monthly active WhatsApp users worldwide from April 2013 to December 2017 (in millions)" https://www.statista.com/statistics/260819/number-of-monthly-active-whatsapp-users/).

Instant messaging clients that did not provide asynchronous communication became uninteresting because of the rise of smartphones and applications that were not always online. The most mature secure messaging protocol, Off-the-Record, did not support asynchronous messaging, which motivated the development of new protocols with asynchronous communication built-in. The most notable is the Signal application with their protocol also called Signal. After a while, the new protocol became quite popular among developers and researchers [4–8]. Subsequently, the Signal protocol started to be implemented in other applications, which were supporting only client-to-server encryption until then.

Quite a number of new secure messaging applications exist (in 2017, we counted six, which we survey in this work) that offer end-to-end encrypted message conversations over mobile phones and computers, but these often sacrifice usability aspects for security. In the light of the above motivations one would probably prioritise privacy and security, but in order to attract most normal users, it should be possible to have the best of both worlds. Applications should give enough information for the users to know when or if a conversation is not secure anymore and the options to secure it once again. Moreover, the security controls should be intuitive and usable enough to be handled by a majority of people, not only for the technology inclined ones.

1.1. The Goals of This Study. The area of end-to-end encryption in instant messaging applications has become rather broad recently. It is difficult for a user to find digestible information sources, and even less when it comes to comparative integrated studies. Therefore, our first goal is as follows:

G1: provide comprehensible and comparative study of relevant approaches to end-to-end encrypted messaging applications.

A detailed analysis of the security and privacy properties provided by secure messaging protocols is not easy, and there are very few such studies (which we build upon). End-to-end encrypted messaging technologies should be both usable so to allow a wide adoption but also have rather strong security requirements. These two, i.e., usability and security, are usually conflicting, and a good balance is difficult to find. This leads to our second goal.

G2: overview the security and privacy properties provided by current end-to-end messaging technologies and to what extent existing applications achieve these properties.

The Signal application (and protocol) is one of the most used end-to-end messaging technologies currently available for smartphones and desktop PCs. Moreover, the Signal protocol is employing state-of-the-art encryption and key establishment techniques.

G3: describe for nonexperts the security mechanisms behind the Signal protocol.

There is little research in the area of usability vs. security in secure messaging applications. Schroder et al. [9] were the first to look at the usability issues for end-to-end encrypted messengers, doing a user study of the usability of Signal's security features and proposing fixes to the issues they found with users failing to detect and deter man-in-the-middle attacks. In this paper, we look at the same types of potential attack as in [9], but we also look at the application interface and various interactions a user has with these applications. Moreover, we extend to five more applications than Signal (namely, we look also at WhatsApp, Wire, Viber, Riot, and Telegram) and check several new application usability properties (summarised in Table 1 from Section 4) for all the six applications under the test. We supplement the fixes proposed in [9] with 12 more recommendations for improvement and in Section 5, applicable to one or more of the tested applications.

Unger et al. [4] did a comprehensive study of secure messaging protocols, looking at security properties related to trust establishment, conversation security, and transport privacy. Since the audience of this paper may include also people without much technical or security skills, we make here an accessible summary of the findings from [4] about the protocols that are implemented by the applications that we study. In order to make this paper self-contained and easier to understand, we also provide an in-a-nut-shell description for each of the three major end-to-end encrypted messaging protocols, i.e., OTR, Signal, and Matrix.

1.2. Main Contributions.

(i) We make a comprehensive analysis of applications that implement secure messaging, by performing five testing scenarios to study their essential security and

Table 1: Overview of the results from the analysis of secure messaging applications test scenarios.

Test scenario and properties	Application					
	Signal	WhatsApp	Wire	Viber	Riot	Telegram
Setup and registration						
Phone registration	●	●	●	●	○	●
E-mail registration	○	○	●	○	●	○
Access SMS inbox	●	●	○	●	○	●
Contact list upload	●	●	●	●	●	●
Verification by SMS	●	●	●	●	○	●
Verification by phone call	●	●	●	●	○	●
Initial contact						
Trust-on-first-use	●	●	○	○	○	○
Notification about E2E encryption	○	●	○	○	●	●
Message after a key change						
Notification about key changes	●	●	○	○	●	○
Blocking message	●	○	○	○	○	○
Key change while a message is in transit						
Re-encrypt and send message	○	●	○	○	○	○
Details about transmission of message	●	●	●	○	●	●
Verification process						
QR-code	●	●	○	○	○	●
Verify by phone call	●	●	●	●	●	○
Share keys through 3rd party	●	●	○	○	○	○
Verified check	○	○	●	●	●	○
Other security implementations						
Two-step verification	○	●	○	○	○	●
Passphrase/code	●	○	○	○	○	●
Screen security	●	○	○	○	○	●
Clear trusted contacts	○	○	○	●	○	○
Delete devices from account	○	○	●	●	●	●

●: has the property; ○: does not have the property.

usability properties. We also provide suggestions for improvements.

(ii) We provide an (updated) overview of conversation security in secure messaging protocols, following [4]. Subsequently, we describe the inner security workings of the latest versions of two major protocols (OTR and Signal), striving to make these understandable for a general audience.

The rest of the paper is organised as follows. Section 2 presents a systematisation of knowledge about three secure end-to-end encrypted messaging protocols, with a discussion of their security properties. Section 3 presents the study testing six mobile phone applications that support either the secure messaging protocols presented before or their own variants which are not open source applications. Section 4 summarises the results from the test scenarios in a unified and comparative manner. Section 5 discusses the applications as a whole also providing recommendations for improvements. Finally, Section 6 concludes the paper.

2. Background on Secure Messaging Protocols

This section provides background on secure messaging protocols that are implemented by the applications analysed in this paper. First, we present the attacker models that we consider and assumptions that we make about the user applications, and then we review basic properties relevant for end-to-end encrypted messaging.

This section builds on the comprehensive survey [4], as well as on various other resources regarding these protocols. Most of the resources for the two new protocols Signal and Matrix are online, since these protocols have not come out of academia. However, both are built on the good foundation laid by the OTR protocol, which has been well studied in academia [10–16], and also complemented by significant online resources.

We relegate the more detailed information on the Off-the-Record and Signal protocols to Appendix. Appendix Section A surveys Off-the-Record (OTR) that is the baseline for the two other protocols, Signal (surveyed in Appendix Section B) and Matrix, which are new protocols that are actually implemented (or copied) by the current popular secure messaging applications.

2.1. Relevant Threat Models. We assume the following adversaries:

(i) Active adversaries: man-in-the-middle attacks are possible on both local and global networks by adding a proxy between the applications and servers

handling the messages. These are under the usual assumptions of a Dolev–Yao model [17].

(ii) Passive adversaries: these adversaries log everything that is sent to and from a user and could potentially use that information to keep track of who users talk to and when. Passive adversaries could also log information such as messages and keys, even though the contents of the messages are encrypted.

(iii) Service providers: the messaging systems that require centralized infrastructure (such as Signal and Matrix) need to keep the information about users secure. The service operators could at any time become a potential adversary.

We assume the endpoints of the messaging applications, e.g., an app on a smartphone, are secure and that the devices do not have malware that could exploit the messaging application.

2.2. Security Principles Relevant for End-to-End Encrypted Messaging. Different secure messaging protocols capture different security principles in various degrees and are important when comparing specifications of applications implementing them. Most security and privacy features that we review here are also found in [4].

End-to-end encryption: communication encryption protocols such as Transport Layer Security (TLS) [18] are designed to secure communications between a client and server. Messaging applications that allow two parties to communicate to each other through a server can use TLS to secure their communication against network attackers. Messages sent to the server are decrypted by the server, which means that it can read, store, or edit the message before encrypting again and sending it to the other user. Often servers cannot be trusted, as they can be hacked by an adversary, or may be contacted by law enforcement to give information sent by clients through the server [19]. End-to-end encryption ensures that the endpoints do the encryption while the servers only transmit the messages without network attackers nor a corrupted server being able to see the content.

Confidentiality: confidentiality ensures that the necessary level of secrecy is enforced at each junction of data processing, preventing unauthorized disclosure [20]. Confidentiality can be provided by encrypting data while it is stored and transmitted. In cryptographic protocols, confidentiality is essential to ensure that keys and other data are available only as intended [21]. Attackers try to break confidentiality by stealing password files, breaking encryption schemes, etc. Users, on the other hand, can intentionally or accidentally disclose sensitive information by not encrypting it before sending it to another person, or by falling prey to a social engineering attack [20].

Integrity: integrity ensures that no one throughout the transmission modifies the messages. Hardware, software, and communication mechanisms must work in concert to maintain, process, and move data to intended destinations, without unexpected alterations. Systems that enforce and provide this security property ensure that attackers, or mistakes by users, do not compromise the integrity of systems or data [20]. This can be achieved through the use of hash functions in combination with encryption, or by use of a message authentication code (MAC) to create a separate check field. Data integrity is a form of integrity that is essential for most cryptographic protocols to protect elements such as identity fields or nonces [21].

Authentication: authentication is meant to identify the parties in a conversation. Message authentication is also called *data-origin authentication* and protects the integrity of the sender of the message [22–27]. Message authentication codes can provide assurance about the source and integrity of a message. A message authentication code is computed by using the message and a shared secret between the two parties [28]. If an adversary changes the message, then the computed MAC would be different as well, and moreover, an adversary cannot produce a valid MAC because only the sender and receiver have the shared secret.

Perfect forward secrecy: a key establishment protocol provides forward secrecy if a compromise of long-term keys of a set of principals does not compromise the session keys established in previous protocol runs involving those principals [29–31]. Typical examples of protocols which provide forward secrecy are key agreement protocols where the long-term key is only used to authenticate the exchange. Key transport protocols in which the long-term key is used to encrypt the session key cannot provide forward secrecy [21, 32].

Future secrecy: future secrecy, as it is called by Open Whisper Systems [33] (sometimes also called backward secrecy), is the guarantee that the compromise of long-term keys does not allow subsequent ciphertexts to be decrypted by passive adversaries [4]. A protocol supports future secrecy when it can provide the "self-healing" aspect of the Diffie–Hellman ratchet, which is described in Appendix Section A, because if any ephemeral key is compromised or found to be weak at any time, the ratchet will heal itself and compute new ephemeral keys for the future messages sent during the conversation [33].

Deniability: deniability is a property common to new secure messaging protocols, where it is not possible for others to prove that the data were sent by some particular conversation party. If Bob receives a message from Alice, he can be sure it was Alice that sent it but cannot prove to anyone else that. To provide deniability, usually secure messaging protocols have a mechanism to allow anyone to forge messages, after a conversation, to make them look like coming from someone in the conversation. Deniability also includes authenticity during the conversation so that the

participants are assured that the messages they see are authentic and are not modified by anyone [34]. Deniability can be divided into three different parts:

(1) Message unlinkability: if a judge is convinced that a participant authored one message in the conversation, it does not provide evidence that they authored other messages.
(2) Message repudiation: given a conversation transcript and all cryptographic keys, there is no evidence that a given message was authored by any particular user. We assume the accuser has access to the session keys, but not the other participants' long-term secret keys.
(3) Participation repudiation: given a conversation transcript and all cryptographic key material for all but one accused (honest) participant, there is no evidence that the honest participant was in a conversation with any of the other participants.

Synchronicity: there are two types of communication, synchronous and asynchronous. Synchronous protocols require all participants to be online for them to receive or send messages. Chat applications are traditionally synchronous communications. Alternatively, asynchronous messaging means that the participants do not need to be online to receive messages, such as SMS text messaging or e-mails, since there is a third party, like a server, to save the information until the recipient gets online again. Modern chat protocols do not use synchronous protocols, usually because of social or technical constraints, such as device battery, limited reception, or other social happenings which do not allow people to be constantly online to receive messages. That is why the majority of instant messaging (IM) solutions provide an asynchronous environment by having a third-party server to store the messages until the other participant gets online to receive it.

Group chat properties: group conversations are popular nowadays, e.g., using Facebook Messenger (https://www.messenger.com), Slack (https://slack.com), or other popular messaging applications (https://www.engadget.com/2016/09/30/12-most-used-messaging-apps/). Security properties in the context of group chats include the following:

(1) Computational equality: whether the participants share an equal computational load when talking to each other.
(2) Trust equality: no single participant has more trust or responsibility, within the group, than any other.
(3) Subgroup messaging: participants can send messages to only a subgroup without generating a new conversation.
(4) Contractible membership: no need to restart the security protocol when a member leaves the conversation.
(5) Expandable membership: there is no need to restart the security protocol when adding a new member after the group has been generated.

It is important to be able to change the cryptographic keys when a new user joins the secure group conversation, since then the new users will not have the ability to decrypt previously exchanged messages. New cryptographic keys should also be exchanged when a user leaves the conversation. Changing the keys can easily be done by restarting the protocol, but this is often computationally expensive. Protocols which offer contractible and expandable memberships usually achieve these features without restarting the protocol.

Other security properties: a protocol or application for end-to-end secure IM may implement any (if not all) of the following:

(1) Participant consistency: at any point when a message is accepted by an honest party, all honest parties are guaranteed to have the same view of the participant list.
(2) Destination validation: when a message is accepted by an honest party, they can verify that they were included in the set of intended recipients for the message.
(3) Anonymity preserving: any anonymity features provided by the underlying transport privacy architecture (such as the Tor (https://www.torproject.org/) network [35,36]) are not undermined (e.g., if the transport privacy system provides anonymity, the conversation security level does not deanonymize users by linking key identifies).
(4) Speaker consistency: all participants agree on the sequence of messages sent by each participant. A protocol might perform consistency checks on blocks of messages during the protocol, or after every message is sent.
(5) Causality preserving: implementations can avoid displaying a message before messages that causally precede it.
(6) Global transcript: all participants see all messages in the same order. When this security feature is assured, it implies both speaker consistency and causality preserving are assured.

2.3. Usability and Adoption Principles for End-to-End Encrypted Instant Messaging. Various aspects need to be taken into account when looking at usability and adoption of a secure IM application:

(1) Out-of-order resilience: if a message is delayed in transit but eventually arrives, its contents are accessible upon arrival.
(2) Dropped message resilient: messages can be decrypted without receipt of all previous messages. This is desirable for asynchronous and unreliable network services.
(3) Asynchronous: messages can be sent securely to devices which are not connected to the Internet at the time of sending.

(4) Multidevice support: A user can connect to the conversation from multiple devices at the same time and has the same view of the conversation as the others.

(5) No additional service: the protocol does not require any infrastructure other than the protocol participants. Specifically, the protocol must not require additional servers for relaying messages or storing any kind of key material.

2.4. Overview of Protocols for End-to-End Encrypted Instant Messaging. According to [4] and as shown in Table 2, none of the secure messaging protocols such as the Off-the-Record, Signal, and Matrix protocols can give the users every security property (for more information about the definition of the properties used in Table 2, please refer to [4]). In this section, we briefly comment on each of these, including for completeness also the Matrix (Matrix protocol having several IM implementations, at https://matrix.org) as a major protocol.

While the Off-the-Record protocol does not need any additional services or servers, it cannot provide group conversation (in the current version and implementations). There have been research works investigating group conversations on top of OTR [14–16], but they have not received enough attention from the developers mainly because these do not support asynchronous chat conversations. While Signal supports desktops through the Chrome Extensions, it does not support native desktop application. Moreover, it only allows for one device to be used; that is, multiple mobile phones cannot be added to a user's account. This could be achieved using the same functionality for group conversations, but efficiency could be a problem.

The Matrix protocols and application (see Section 2.4.3 for details) support multiple devices, without affecting the efficiency of the conversations. However, it does not achieve full forward and backward secrecy in the protocols, but the implementation does. The Signal protocol has been audited by two research groups in 2017 [6, 7] and since it is open source, the community can improve it. The Matrix protocol has also been audited [37]. This indicates that researchers are taking these protocols seriously and want to strengthen their credibility.

2.4.1. Off-the-Record. OTR uses an encrypt-then-MAC approach to protect messages (see Appendix Section A for details) which provides confidentiality, integrity, and authentication. The SIGMA protocol (a variant of authenticated Diffie–Hellman key exchange) ensures participation consistency for the key exchange [38]. Forward secrecy is ensured by the fact that message keys are regularly replaced with new key material during the conversation. Backward secrecy is ensured by the fact that message keys are computed by new DH values which are advertised by the sender with each sent message. Anonymity preservation is ensured by the fact that the long-term public keys are never observed, neither during the key exchange nor during the conversation. Causality preservation is only partially achieved, as messages implicitly reference their causal predecessors based on which keys they use [4]. Speaker consistency is only partially achieved since an adversary cannot drop messages without also dropping all future messages, for otherwise the recipients would not be able to decrypt subsequent messages [4]. The aftermath of the speaker consistency is that the recipient needs to save out-of-order messages because if they do not come in order, the message will be encrypted with an unexpected key, and at the same time the window of compromises enlarges, and the OTR would end up only partially providing the forward secrecy. Out-of-order and dropped messages are only partially provided because if a message is out-of-order or dropped during the transmission, the protocol can store the decryption key until the participant receives that message. The problem of storing the decryption key is that it raises the possibility of successful attacks on the client side.

The OTR protocol signs the messages with the shared MAC keys and not the long-term keys. To strengthen the message unlinkability and message repudiation features, OTR uses malleable encryption and the MAC keys are published after each message exchange [10]. OTR only signs the ephemeral keys and not every parameter during the key

Table 2: Comparison of secure messaging protocols (reproduced from [4]).

Properties	Protocol/client		
	OTR Pidgin	Signal Signal	Matrix Riot
Security and privacy			
Confidentiality	●	●	●
Integrity	●	●	●
Authentication	●	●	●
Participant consistency	●	●	●
Destination validation	●	●	●
Forward secrecy	⋈	●	⋈
Backward secrecy	●	●	⋈
Anonymity preserving	●	◷	◷
Speaker consistency	⋈	●	●
Causality preserving	⋈	●	●
Global transcript	◷	◷	◷
Message unlinkability	●	●	●
Message repudiation	●	●	●
Participation repudiation	⋈	●	●
Usability and adoption			
Out-of-order resilient	⋈	●	●
Dropped message resilient	⋈	●	●
Asynchronicity	◷	●	●
Multidevice support	◷	⋈	●
No additional service	●	◷	◷
Group chat			
Computational equality	◷	●	●
Trust equality	◷	●	●
Subgroup messaging	◷	●	●
Contractible membership	◷	●	●
Expandable membership	◷	●	●

●: provides the property; ⋈: partially provides the property; ◷: does not provide the property.

exchange, which provides only partial participation repudiation since the conversation partners can use the signed ephemeral keys to forge transcripts. The OTR protocol is intended for instant messaging and thus does not provide asynchronous messaging. However, the synchronous only requirement allows OTR to not rely on additional services for establishing a connection between two participants.

2.4.2. Signal. The design of the Signal protocol extends OTR, thus maintaining the same security features, while in some cases adding stronger or new features as well. Forward secrecy is provided because of the use of thee KDF ratchets, whereas backward secrecy is provided because even when KDF keys are comprised, they are soon replaced by new keys. The X3DH handshake (Appendix Section B.3) provides the same level of authentication as the SIGMA from OTR, but X3DH achieves full participation repudiation since anybody can forge a transcript between two parties [4]. However, Signal fails to provide anonymity preserving because X3DH uses the long-term public keys during the initial key agreement. The prekeys are used to provide an asynchronous messaging system by sending a set of prekeys to a central server, and then a sender can request the next prekey for the receiver to compute encryption keys. By using a central server to keep the prekeys, the Signal protocol loses the no additional service property. Out-of-order and dropped messages are fully supported on one-to-one conversations asynchronously by the use of prekeys.

Group conversation is achieved by using multicast encryption, in which when sending a single encrypted message to the group, it is sent to a server and then relays it to the other participants while the decryption key is sent as a standalone message to each member of the group conversation. The group conversation provides asynchronous messaging, speaker consistency, and causality preservation, by attaching message identifiers, of the messages before, to the new message [4], but it cannot guarantee participant consistency. Multidevice is partly provided, in the sense that only an extra computer can join in a conversation by using the Signal Desktop application (https://whispersystems.org/blog/signal-desktop/), which is only a Chrome Extension (https://en.wikipedia.org/wiki/Browser_extension) and not an own application.

The Signal protocol provides computational and trust equality, subgroup messaging, and contractible and expandable membership properties. By using pairwise group messaging and multicast encryption, Signal has the ability to push group management into the client apps, which makes it easier for the users to change the group, expand it, or shrink it in size, without having to restart the whole group conversation and protocol. When users want to send a group message, they send a message to each of the users that are participating and adding a parameter to the header marking that it is meant for the specific group chat. The Signal server does not know about the group conversation, since the messages are encrypted using their normal public key. The pairwise group messaging also makes the computation of new cryptographic keys and trust equality as computationally demanding as if there was only a one-to-one conversation.

2.4.3. Matrix. The Matrix protocol consists of two different algorithms, the Olm (https://matrix.org/docs/spec/olm.html) for one-to-one conversations and Megolm (https://matrix.org/docs/spec/megolm.html) for group conversations between multiple devices. The Olm algorithm is based on the Signal protocol, which means they achieve the same security properties as Signal does, while the Megolm algorithm is a new AES-based cryptographic ratchet developed for group conversations. Multiple devices are possible with Matrix because Megolm implements a separate ratchet per sending device that is participating in a group conversation (Matrix.org Launches Cross-platform Beta of End-to-End Encryption Following Security Assessment by NCC Group https://pr.blonde20.com/matrix-e2e/). The protocol does not restart when the ratchet is replaced with a new one, which provides computational and trust equality, subgroup messaging, and contractible and expandable membership properties.

The NCC Group has audited both algorithms [37] and found that Megolm has some security flaws about forward and future secrecy. If an attacker manages to compromise the key to Megolm sessions, then it can decrypt any future messages sent to the participants in a group conversation. The Matrix SDK, which is used in the applications that have implemented the Matrix protocol such as Riot (https://about.riot.im/); ensures that the Megolm keys get refreshed after a certain amount of messages. Forward secrecy is only partially provided since the Megolm maintains a record of the ratchet value which allows them to decrypt any messages sent in the session after the corresponding point in the conversation. The Matrix developers have stated that this is intentionally designed [39] but also said that it is up to the application to offer the user the option to discard old conversations.

3. Analysis of Applications Implementing Secure Messaging

This section surveys applications (mostly smartphone apps) that advertise secure messaging conversation capabilities between one-to-one and/or many-to-many users. We investigate a set of usability properties relevant for secure messaging. This is in contrast to the previous works [4] which looked at the protocols that are underlying some of the applications that we are evaluating. Here, we introduce and test more properties than in [9] that are specific for applications.

3.1. Test Scenarios. We first explain in this section the test scenarios that we carried and the security and usability properties we are looking for. The test scenarios are the same for each application, and screenshots were taken during the testing phases to gather enough information for later analysis. In each test scenario, we are going to study a set of properties regarding the security and usability of each of the

six applications. The results from testing the 21 properties described here will be summarised in Section 4 and Table 1.

3.1.1. Initial Setup. This test scenario includes two stages: "setup and registration" and "initial contact." Stage one is the first process a user needs to go through after installing an application. Here, we test how the applications handle the registration process, what the user needs to do to register a new account, and whether there are multiple ways to register. The properties of interest are as follows:

(i) Phone registration: user can register an account with a phone number.

(ii) E-mail registration: user can register an account with an e-mail address.

(iii) Verification by SMS: receive verification code through SMS.

(iv) Verification by phone call: receive verification code through a phone call.

(v) Access SMS inbox: the app requires access to the SMS inbox in order to read the verification code automatically.

(vi) Contact list upload: the app requires to upload contacts to a server in order to see if others are using the same application.

Stage two examines how applications handle the first message sent from one participant to another, whether the participants are informed of the secure messaging capabilities or whether the app shows how the cryptographic keys are used. Properties are as follows:

(i) Trust-on-first-use: automatically verify each other on initiation.

(ii) Notification about E2E encryption: the app presents notifications to explain to the user that messages are encrypted.

3.1.2. Message after a Key Change. This scenario tests how the application handles changes of cryptographic keys after Bob deletes the application in the middle of a conversation with Alice. After Bob has reinstalled his application, Alice sends him a new message and examines if the application gives Alice any information about the key changes. When a user deletes a secure messaging application, the cryptographic keys are normally deleted from the device to strengthen the security of the messages the participant has already sent. When a participant then reinstalls the application, a new set of cryptographic keys are generated. Properties of interest are as follows:

(i) Notification about key changes: notifying Alice that Bob has changed cryptographic keys.

(ii) Blocking message: blocking new messages from being sent until Alice and Bob verify each other.

3.1.3. Key Change While a Message Is in Transit. Cryptographic key changes while a message is in transit are similar to the test scenario before; however, we are interested in what happens when a message is lost before new keys are generated. Bob deletes his application without telling Alice; she then sends Bob a message, but the message is lost in transit. Properties of interest are as follows:

(i) Re-encrypt and send message: does the application re-encrypt the message and send it again after the receiver has generated new cryptographic keys or is the message lost forever?

(ii) Details about transmission of message: users can see the difference between sent and delivered messages.

3.1.4. Verification Process between Participants. In a conversation, Alice and Bob want to verify each other to ensure that they are having a conversation with honest participants. This test scenario looks at how the verification process works and if it is a secure and usable method of doing it:

(i) QR-code: verify each other through a QR-code.

(ii) Verify by phone call: call each other with E2E-encrypted phone call and read keys out loud.

(iii) Share keys through 3rd party: share the keys through other applications.

(iv) Verified check: users can check later if a specific user is already verified.

3.1.5. Other Security Implementations. Each application may have additional security and privacy features meant to protect from various intrusions or attacks.

(i) Passphrase/code: possibility of a passphrase/code that only the user knows and enters to access the application.

(ii) Two-step verification: when registering after a reinstall or new device, then a second passphrase/code is needed which only the specific user knows.

(iii) Screen security: the user is not allowed to screenshot within the application.

(iv) Clear trusted contacts: can all verified contacts be cleared, which means the user needs to verify each contact again?

(v) Delete devices from account: if the application allows multiple devices, is there an option to delete devices which are not in use anymore.

3.2. Running the Different Test Cases. For our tests, we used two separate smartphones as described in Table 3. Both phones have their personal phone number; the Sony phone has the contact information of the Nexus phone named Bob, while the Nexus phone has the contact details of the Sony phone named Alice. The reason behind the contact details is

Table 3: The phone models involved in the testing.

Phone	Alice	Bob
Model	Sony Xperia Z5	Google Nexus 5X
OS	Android 7.0	Android 7.1.2
Security patch	December 1st, 2016	January 5th, 2017
Kernel	3.10.84-perf-gda8446	3.10.73-gbc7f263
CPU	Qualcomm MSM8994 Snapdragon 810	Qualcomm MSM8992 Snapdragon 808
Memory	3 GB	2 GB

to quickly find each other when initiating a conversation during the testing.

The applications used during the testing phase are locked to one version number and did not get updates, in order to keep the tests consistent. Both devices have installed the same applications under test with an identical version number.

3.2.1. Case 1: Signal. Signal is an instant messaging application as well as a voice calling application, for both Android and iOS. What sets the Signal application apart from the other applications, except for Riot, is the fact that it is completely open source. This reassures people that it does what the developers claim, since anyone can audit the source code as well as the employed cryptographic protocols.

(1) Initial Setup. An account can be registered to one device at a time, which means that using the same number on a second device will automatically deactivate the first device, to strengthen the security and to keep the private cryptographic keys on one device only. Figure 1(a) shows the first view a user sees when opening the app for the first time. Twilio is used for handling the SMS verification process with the Signal server when registering an account. Contact information is transmitted to the server but is not stored.

Figure 1(b) explains the different steps the Signal app goes through to register and verify a new user account. The verification code is sent as an SMS, and the app reads the SMS automatically to verify the new user. After the verification, the app generates new device cryptographic keys. At the end, the app registers the account within the Signal server. If the user does not give the application access to their SMS inbox, then it has to wait for the SMS verification timer to time out, as shown at the bottom of Figure 1(b), after which the Signal application calls the user and gives out a verification number to be typed in manually.

(2) Message after a Key Change. This test scenario analyses what happens when cryptographic keys change, e.g., when a user in a conversation deletes and then reinstalls the Signal app. Figure 2(a) shows the first two messages that Alice sends to Bob. The double checkmark shown on each message indicates that it has been received and read by Bob. The lock indicates that the message is encrypted from one end to the other, and nobody in between can read it. Figure 2(b) shows when Alice sends Bob another message after he has deleted and reinstalled his Signal app. The application notifies Alice that the message has not been delivered with a red notification icon on the left. It also gives information that by pressing on the message, the user can get more details about the notification.

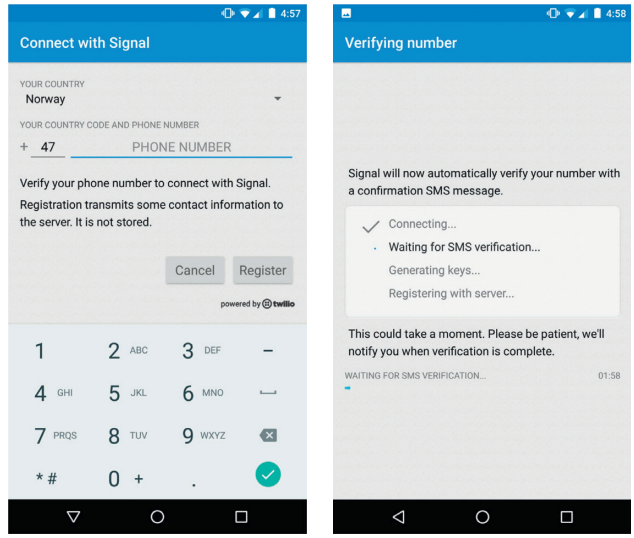
Figure 2(c) shows the view the user sees when pressing the message that was not delivered. Alice is presented with information that Bob has a new security number (cryptographic keys), and she needs to verify the new keys to get the ability to send any new message to Bob. After the verification process between Alice and Bob is done, they can continue the conversation, and a notification is posted in the conversation that Bob has changed his security number, as shown in Figure 2(d).

(3) Key Change While a Message Is in Transit. This test scenario is mostly the same as the previous one, with the difference that here we want to check what does the Signal app do when a message is sent before Bob has managed to reinstall his Signal application, i.e., handling of messages lost in transit.

Figure 3(a) shows the initialization of the conversation. Figure 3(b) shows the conversation after a couple of messages from Alice to Bob. The second message is sent after Bob has deleted his application, and there is only a single checkmark on that message, which means the message has been sent, but not received. The icons on the third message indicate that Bob has finally reinstalled, but he never received the second message which was sent before he reinstalled. After Alice and Bob verify their new security number, all new messages are received and encrypted by both sides, but the second message is never received. The reason for never receiving the second message in Figure 3(b) is because the Signal app never stores messages that are encrypted after they are sent to the server, and the messages are never re-encrypted by Alice when Bob has changed his cryptographic keys.

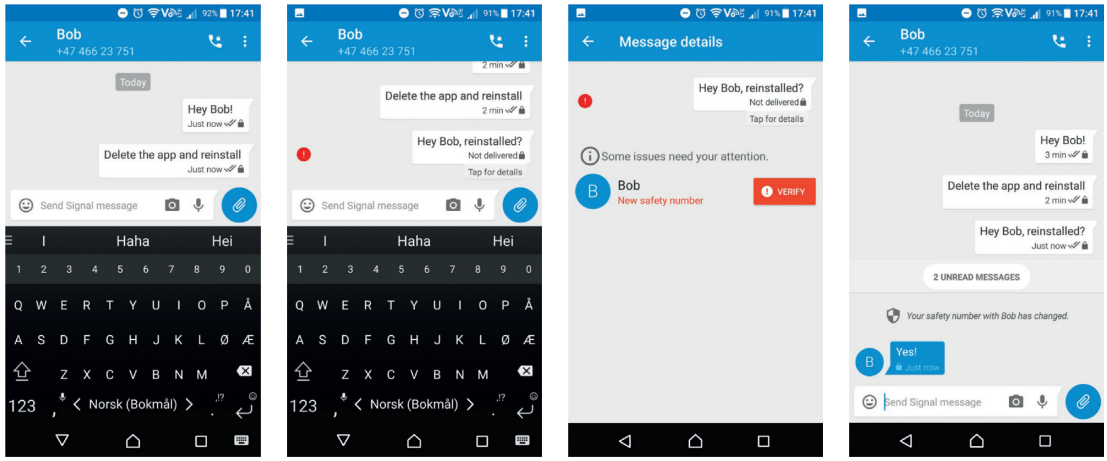
(4) Verification Process between Participants. Signal supports three different methods for the users to verify each other. The first verification process uses the built-in calling option of Signal which is end-to-end encrypted and then read out loud to the other participant the security numbers that are shown in Figure 3(c). If the Signal calling is not regarded as secure enough, users can meet in person and show the numbers to each other.

The second method, shown in Figure 3(c), uses the built-in QR-code scanner to scan the other participants QR-code to verify it is the same person in the chat. The third option to verify the other user is meant to be used when the users do



(a) (b)

Figure 1: Signal: registration process. (a) Phone number registration. (b) Verifying the phone number.



(a) (b) (c) (d)

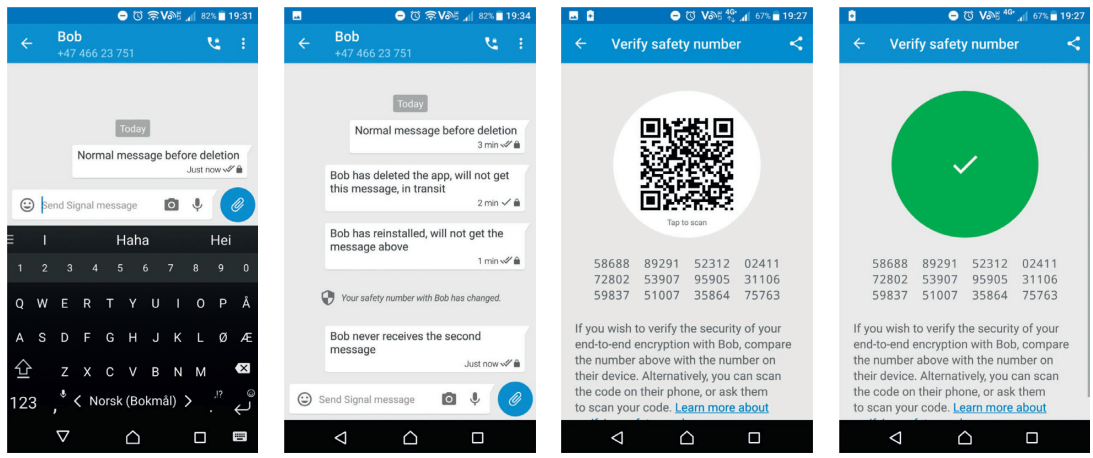
Figure 2: Signal: message after a key change. (a) Alice's first message. (b) Message after reinstall. (c) Verifying Bob again. (d) Message after verification.

not trust the Signal application for handling the verification process. It is possible to share the security numbers to other applications on the user's phone. The user may have PGP (https://en.wikipedia.org/wiki/Pretty_Good_Privacy) [40,41] enabled e-mail on their phone, and they trust it more than the Signal application; then, this method is a better way of verifying the other user.

(5) Other Security Implementations. The Signal application has extra privacy settings. The first is the "Safety numbers approval" as shown in Figure 4(a). The setting is activated by default, which is important. When a user changes the safety numbers (cryptographic keys) by deleting and reinstalling the app, the device keys will also change. When the device keys change, the messages will not be shown to the receiver on a new device, until the new safety numbers are approved. The second privacy setting is "Screen security," which does not allow the user to take screenshots as long as they are inside the Signal application.

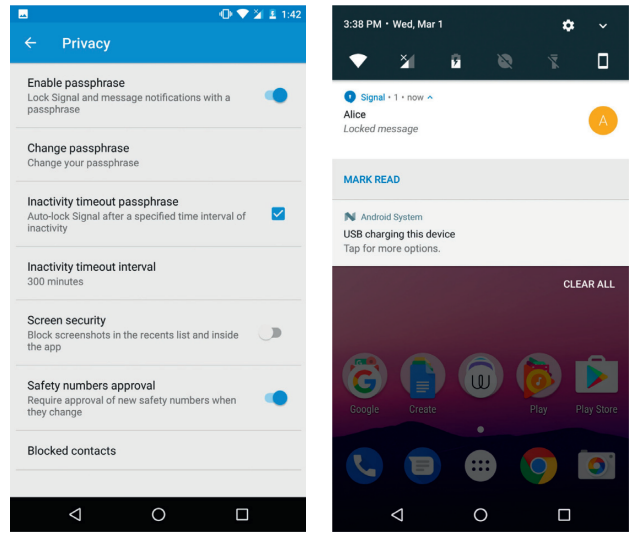
The last privacy setting is the ability to enable a passphrase. The passphrase locks the Signal application and all message notifications. It is possible to add an inactivity timeout passphrase which locks the application after some given time. Figure 4(b) shows the notification which is locked and when the user tries to open the application, she needs to enter the passphrase they chose when the setting was activated.

3.2.2. Case 2: WhatsApp. WhatsApp started as a small company in 2009, bought by Facebook in 2014 when it had 465 million monthly active users, and in 2017 that number



(a) (b) (c) (d)

Figure 3: Signal: key change while a message is in transit and verification process. (a) Message before key change. (b) Message after key change. (c) Verification page. (d) Verified.

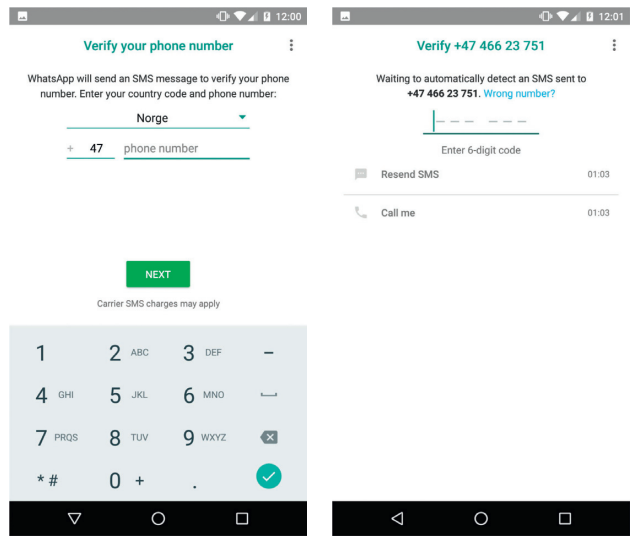


(a) (b)

Figure 4: Signal: other security implementations. (a) Privacy settings. (b) Notification locked.

has grown to 1.5 billion (the statistics portal: "Number of monthly active WhatsApp users worldwide from April 2013 to December 2017 (in millions)" https://www.statista.com/statistics/260819/number-of-monthly-active-whatsapp-users/). WhatsApp initially was only a cross-platform nonsecure instant messaging, but by the end of 2014 they announced that every user was going to start sending end-to-end encrypted messages using the Signal protocol ("Open Whisper Systems partners with WhatsApp to provide end-to-end encryption," announced by Moxie Marlinspike from Open Whisper Systems on November 18 2014, at https://whispersystems.org/blog/whatsapp/). This was an important step for the Signal protocol and Open Whisper Systems since now the most popular instant messaging application would use their protocol. In April 2016, a complete transition was made from nonsecure messaging to fully end-to-end encryption ("WhatsApp's Signal protocol integration is now complete", announced by Moxie Marlinspike from Open Whisper Systems on April 05 2016, at https://whispersystems.org/blog/whatsapp-complete/).

(1) Initial Setup. Establishing an account on WhatsApp is done in the same way as for the Signal application, where the account only works on one device at a time. Figure 5(a) shows the first page a user sees when starting the application for the first time. WhatsApp uses its own infrastructure to handle the SMS verification process instead of a third party. Figure 5(b) shows the verification page after the user has entered her phone number. WhatsApp automatically enters the verification code that is sent to the user's SMS inbox, but if the user has not given the app access to the inbox, she can enter the verification code manually. If for some reason the



(a) (b)

Figure 5: WhatsApp: registration process. (a) Phone number registration. (b) Verifying the phone number.

verification code does not arrive, the user has the options to either resend the SMS or ask WhatsApp to call the user to receive the verification code through voice.

(2) Message after a Key Change. Figure 6(a) shows when Alice sends her first and second messages to Bob in order to initiate a conversation. The yellow notification box at the top of the conversation is shown to both participants stating that the conversation is end-to-end encrypted and more information can be obtained by pressing the box. Each message is shown with a double checkmark, as in Signal.

Figure 6(b) shows a new notification box appearing on Alice's conversation page after Bob has reinstalled his application. WhatsApp automatically checks if new cryptographic keys (security code) are changed even though she has not sent him any message. When Alice taps the notification box, a popup (Figure 6(c)) informs Alice why Bob's cryptographic keys have changed and the option to verify him before she sends new messages. Alice sends a new message to Bob after verifying new cryptographic keys of Bob and the message is labeled (Figure 6(d)) with a double checkmark meaning that everything went well.

(3) Key Change While a Message Is in Transit. This test scenario starts as the previous one, but here we look at how WhatsApp handles messages sent before Bob finishes to reinstall. Figure 7(a) shows Alice sending a second message to Bob after he has deleted his application. The single checkmark on the message means that it has been sent but not received and read by Bob. When Bob finishes the reinstallation of the application, both the second message Alice sent and the same yellow notification box are added to the conversation. Figure 7(b) displays the conversation after Alice sends a third message, showing that Bob receives the second message that was sent before he reinstalled. This means that WhatsApp re-encrypts messages when the receiver generates new cryptographic keys and the sender does not verify the new keys.

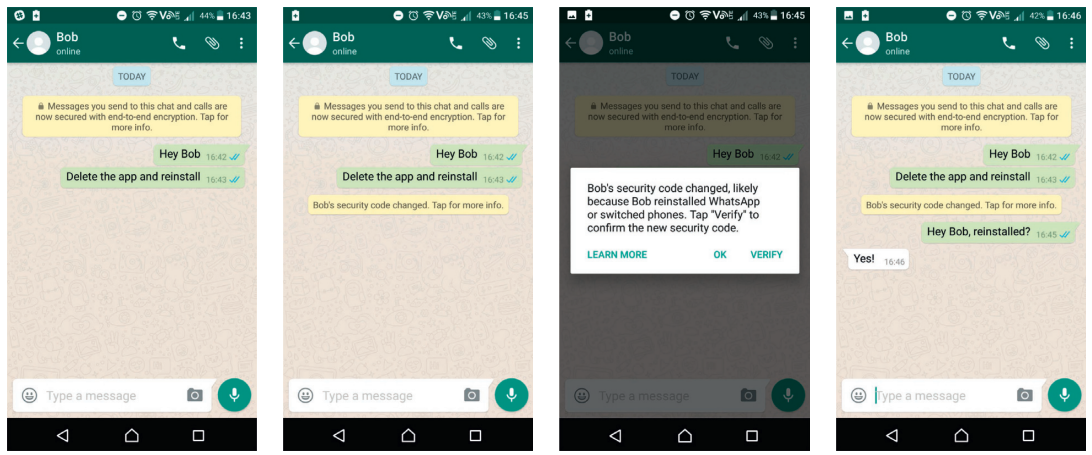
(4) Verification Process between Participants. WhatsApp has implemented the same verification process as Signal. It uses the Signal numerical format for verification, a QR-code for scanning with the built-in scanner, and the user can choose if they want to copy the security numbers outside of the WhatsApp application. The reason for this may be that when they decided to implement the Signal end-to-end security protocol, they implemented every single step of the Signal implementation to uphold the specifications. WhatsApp also features end-to-end encrypted calling, which enables users to call each other and verify the security code.

(5) Other Security Implementations. As shown in Figure 8(a) (setting page), there are options for changing the number of the account or even delete the account.

If Alice chooses the "Security" menu item and then activates the "Show security notification" option (shown in Figure 8(b)), then when Bob reinstalls the application or receives a new device the application shows a notification to Alice. Otherwise, if the option is turned off, then Alice does not receive any notification.

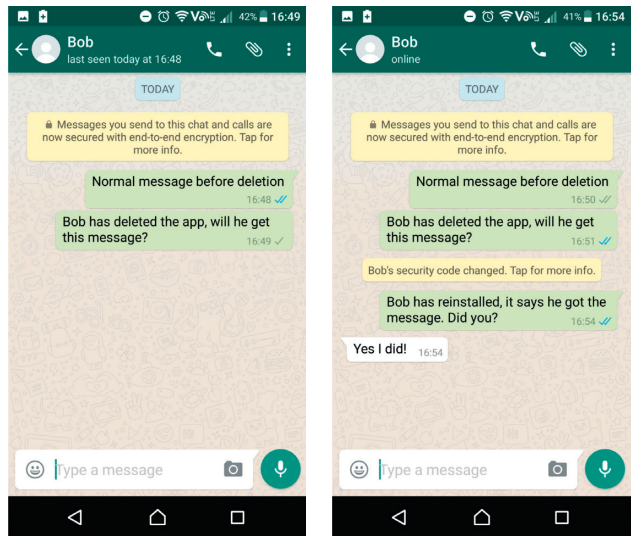
Figure 8(c) shows the two-step verification settings that WhatsApp has implemented, where the user needs to enter an additional passphrase when registering the account with the same number on a new device or after a fresh reinstall.

3.2.3. Case 3: Wire. Wire is an application that implements end-to-end encryption using the Proteus protocol, which is heavily based on the Signal protocol, but reimplemented in-house (Proteus Protocol, by Wire Swiss GmbH available at https://github.com/wireapp/proteus). Wire was started in 2012 by developers who previously worked at Microsoft and Skype, and finally, they released their own instant messaging



(a) (b) (c) (d)

FIGURE 6: WhatsApp: message after a key change. (a) Alice's first message. (b) After Bob has reinstalled. (c) Info about Bob's new keys. (d) Message after verification.



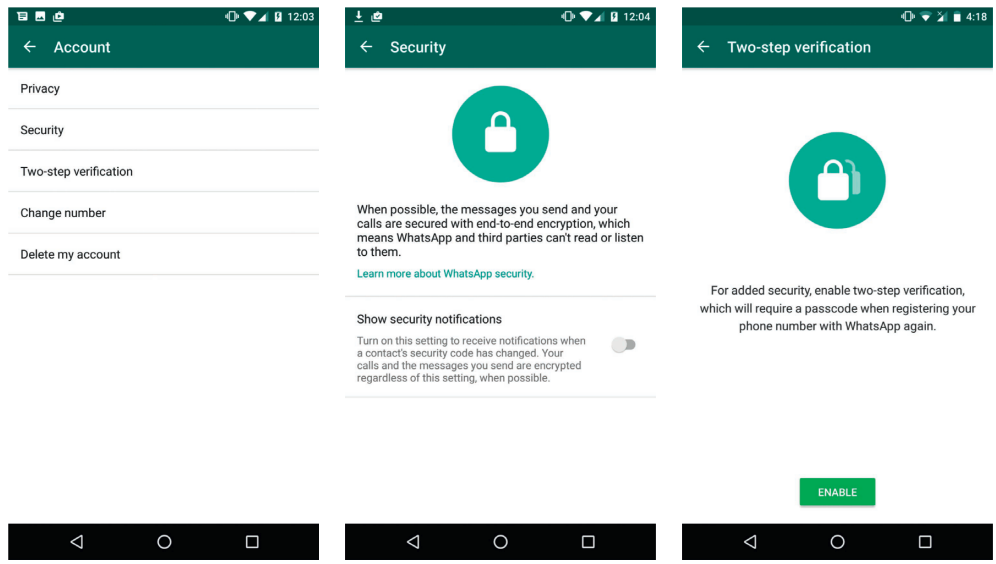
(a) (b)

FIGURE 7: WhatsApp: key change while message is in transit. (a) Bob deletes his app. (b) Bob reinstalled.

application in 2014 ("Skype Co-Founder Backs Wire, A New Communications App Launching Today On iOS, Android And Mac, by Sarah Perez in Tech Crunch on December 2014, available at https://techcrunch.com/2014/12/02/skype-co-founder-backs-wire-a-new-communications-app-launching-today-on-ios-android-and-mac/). The first version did not offer end-to-end encryption until March 2016, when they launched the encryption on instant messaging and their video calling feature [42]. Wire offers the same features as the other applications, such as text, video, voice, photo, and music messages, is supported on multiple platforms, from smartphones to personal computers, and is also open sourced (https://github.com/wireapp).

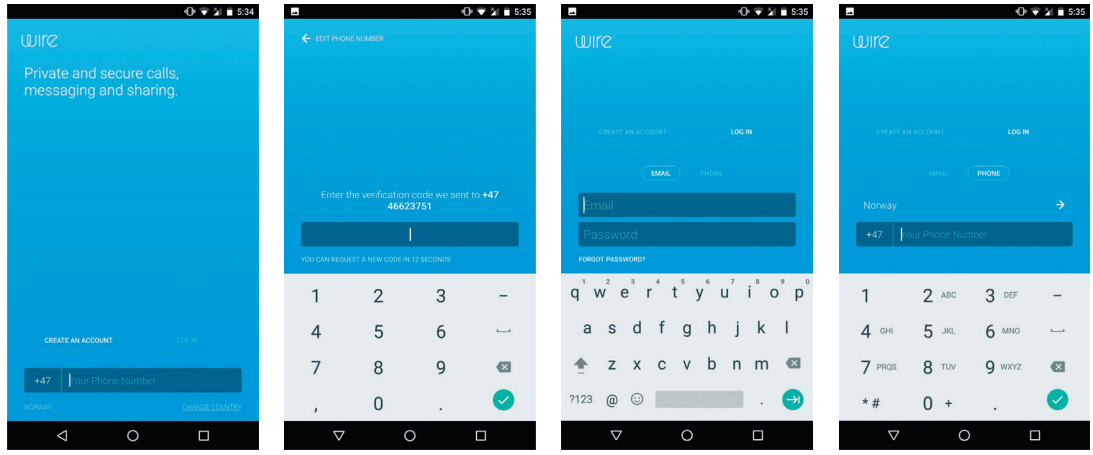
(1) Initial Setup. The Wire app has a different registration process than the other applications. The first page, as shown in Figure 9(a), asks to register a phone number. However, one can also create an account through the Wire web application by using just an e-mail address. Figure 9(b) shows the verification process, where the user needs to enter the verification code (which is received in an SMS) manually. If the user never receives the verification code, she can ask Wire to call the user to receive it. Wire application does not read the code received in the SMS automatically. Figures 9(c) and 9(d) demonstrate options to log in with an e-mail and/or a phone number, respectively. When a user reinstalls the application or changes her device, she does not need to go through the registration again.

(2) Message after a Key Change. Figure 10(a) shows Alice's initial contact with Bob. Wire uses a text under each message to explain if the message is delivered to the



(a) (b) (c)

Figure 8: WhatsApp: other security implementations. (a) Privacy settings. (b) Security notification. (c) Two-step verification.

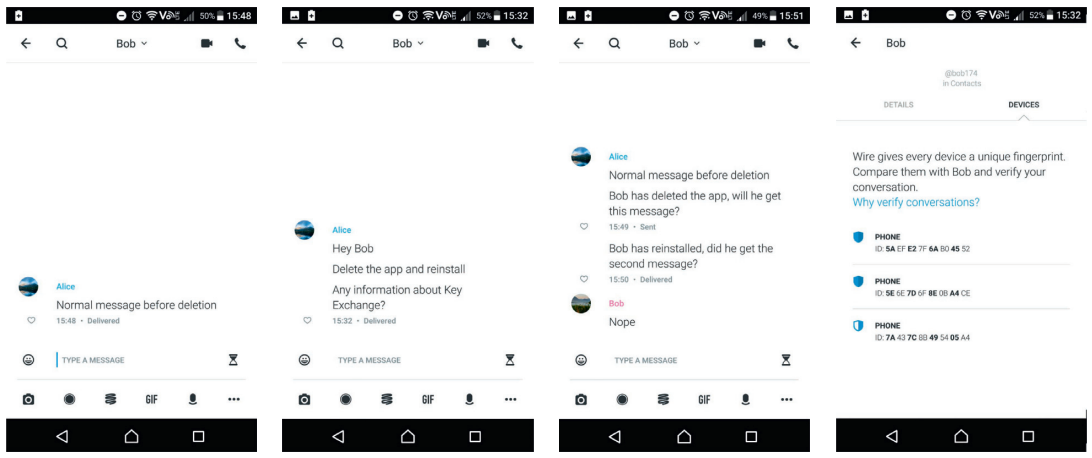


(a) (b) (c) (d)

Figure 9: Wire: registration process. (a) Phone number registration. (b). Phone verification. (c) User login with e-mail. (d) Login with phone number.

recipient or not. If the receiver reinstalls his application (which results in changing his cryptographic keys), then the sender (i.e., Alice) will not be notified by the Wire application about the new cryptographic keys of the receiver. As shown in Figure 10(b), Alice sends two messages to Bob, where Bob has reinstalled his application, but Alice does not get any notification by Wire that Bob has new cryptographic keys; it may look to Alice that Bob has the same keys as before. Alice can check Bob's account information to see if Bob has got new cryptographic keys. Figure 10(d) shows Bob's device keys under his account, for three different devices, because Wire allows multiple devices to be associated to one account. This means that Alice needs to verify each device to know that the conversation is secure with end-to-end encryption. The two top devices have a full blue shield which means they are verified, while the bottom device only has a half shield because it has not been verified yet.

(3) Key Change While a Message Is in Transit. Figure 10(a) shows the initial message from Alice to Bob. If Alice sends a message when Bob has deleted the app, the message will be labeled by just "sent" and not delivered (Figure 10(c)). However, if Alice sends another message when Bob has reinstalled the app, then the message will be labeled as "delivered." This test demonstrates that Wire does not notify Alice about Bob's new keys and at the same time does not deliver messages encrypted with old cryptographic keys to devices with new keys.



(a) (b) (c) (d)

Figure 10: Wire: key change while a message is in transit and message after a key change. (a) Alice's first message before deletion. (b) After Bob has reinstalled. (c) No notification after Bob reinstalls. (d) Bob's device keys.

(4) Verification Process between Participants. Wire's verification process does not offer the same options for verifying each participant as the other applications. However, because Wire allows several devices to be associated with one account, and each user has access to the whole list of devices of another conversation party. When Alice wants to verify one of Bob's devices, she can see Bob's profile and particularly the tab displaying information about all his devices (Figure 10(d)). If Alice taps on one of the devices from the list, as shown in Figure 11(a), she can get some information about the phone's ID number and the public keys of that device. Alice can either call Bob over the phone or meet him in person and then verify the keys. When the verification is done, Alice needs to toggle the "not verified" switch to specify that this particular device is verified.

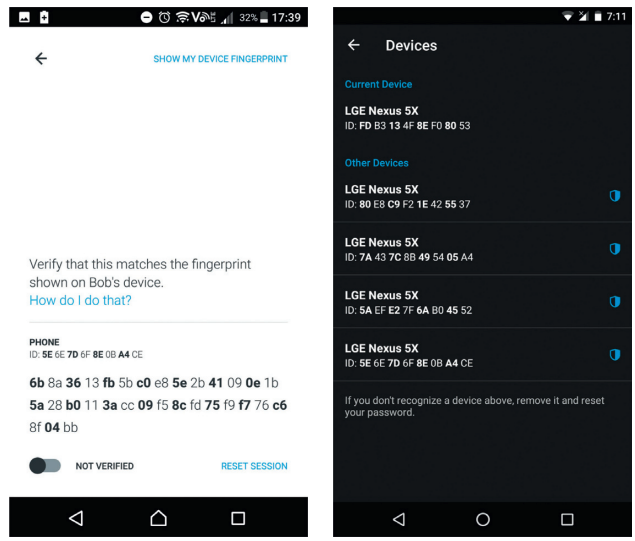
(5) Other Security Implementations. Wire does not have the extra security implementations that Signal or WhatsApp has. The few options include a way to change how the message conversation looks and the possibility to add an e-mail to the account for easier log in. The user can look at the devices which have been used with her account (as shown in Figure 11(b)), and if there are any devices which the user does not own or recognize, she can delete that specific device. After deleting a device, the user is prompted to change her password.

3.2.4. Case 4: Viber. Viber is another instant messaging application that was launched in 2010 and has become quite popular, with 800 million overall users and 266 million monthly active users [43]. Viber has properties similar to the other applications, where users are capable of forming groups, send messages, call each other, and send pictures, videos, or voice messages to other users of Viber (Viber, by Rakuten Inc. https://www.viber.com/en/about). Viber works on smartphones and personal computers, which makes it cross-platform. Viber did not have end-to-end encryption in the beginning, but it is introduced in April 2016 for both one-to-one and group conversations ("Giving Our Users Control Over Their Private Conversations," by Michael Schmilov from Viber on April 19 2016, at https://www.viber.com/en/blog/2016-04-19/giving-our-users-control-over-their-private-conversations). Viber does not use the Signal protocol but implements its own protocol, which has the same concepts as the double ratchet protocol used by Signal (as stated by its developers).

(1) Initial Setup. The user registration process of Viber is the same as that in the previous applications. Figure 12(a) shows the user input for the user's phone number in the registration screen. Figure 12(b) shows the activation process of the user account. The user can either give Viber access to the SMS inbox to enter the verification code automatically or do it manually otherwise. If the SMS with the verification code does not arrive within one minute, the user can ask the application to either resend a new verification code or get the code through a phone call.

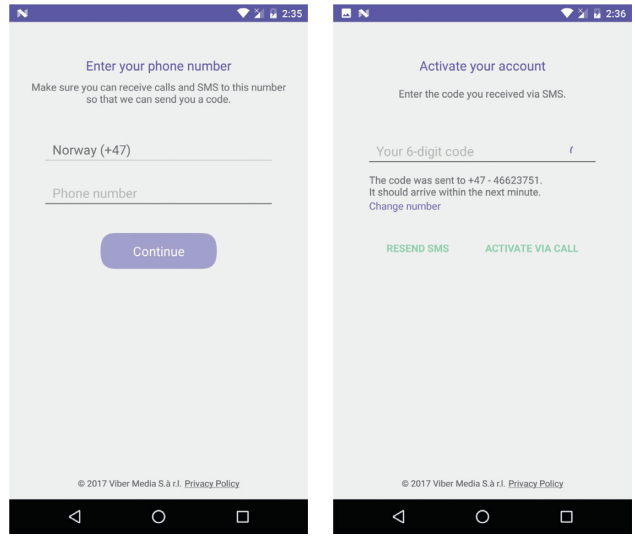
(2) Message after a Key Change. Viber does not notify the participants when the cryptographic keys change during a conversation. The first two messages in Figure 13(a) show Alice initiating the conversation with Bob. After Bob has reinstalled the application, Alice sends him another message. However, Figure 13(a) (third message) shows that Viber does not give any notification to Alice that Bob has generated new cryptographic keys. The only way for Alice to find out this is by checking the details of the conversation, by swiping from right to left. As shown in Figure 13(b), if the "Trust this contact" tab has changed to "Re-trust this contact," then Alice can infer that Bob has new cryptographic keys that should be verified.

(3) Key Change While a Message Is in Transit. Key changes in transit are handled the same as key changes after the reinstall of the application explained before. Figure 13(c)shows that Alice sends a second message to Bob before he has reinstalled his application, and there is no information given to



(a) (b)

Figure 11: Wire: verification process. (a) Bob public keys for a device. (b) Other security features.



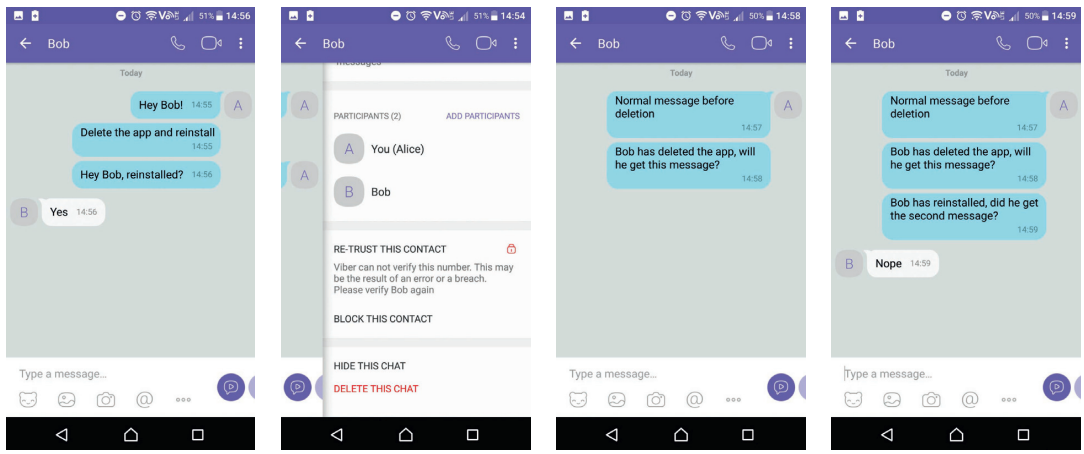
(a) (b)

Figure 12: Viber: registration process. (a) Registration with phone number. (b) Verification of phone number.

Alice if the message is sent or read by Bob. Figure 13(d) shows the third message from Alice to Bob after he has reinstalled his application. Alice never receives any notification from Viber that Bob has new cryptographic keys nor that he has not received the second message, and Viber does not re-encrypt and resend messages later on.

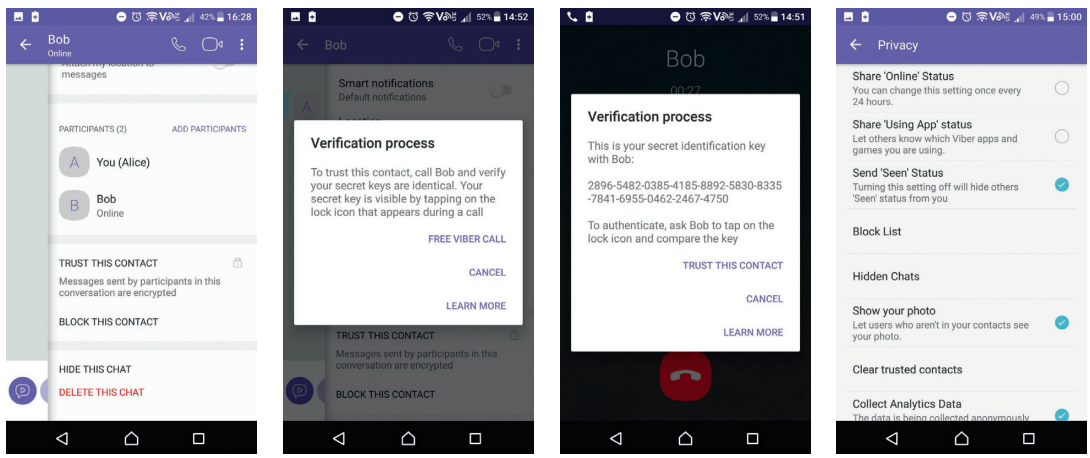
(4) Verification Process between Participants. The process of verifying a contact in Viber is quite straight forward. If Alice wants to verify Bob, she goes to one of their conversations, swipes (Figure 14(a)) to get the information tab, and then goes to the "Trust this contact" option. Figure 14(b) shows the popup notification box after Alice clicks the "Trust this contact" option. The only verification option Alice can use to verify Bob is by calling Bob and then read the cryptographic keys over the phone. Figure 14(c) shows when Alice calls Bob and wants to verify, the popup message displays the cryptographic keys that both Alice and Bob share. When they have verified each other, they press the "Trust this contact" button.

(5) Other Security Implementations. Viber does not have extra security implementations. Figure 14(d) shows the privacy settings where the only security implementation is the "Clear trusted contacts" which clears all the contacts that Alice has verified throughout the time she had the account.



(a) (b) (c) (d)

FIGURE 13: Viber: message after a key change and key change while a message is in transit. (a) No notification about key changes. (b) Bob needs to be retrusted. (c) Messages after Bob has deleted. (d) Bob did not get the second message.



(a) (b) (c) (d)

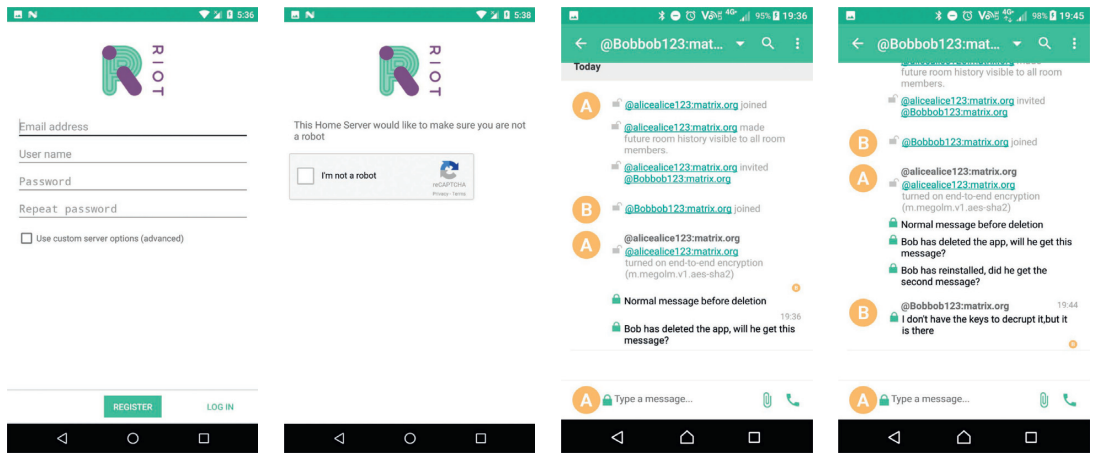
FIGURE 14: Viber: verification process and privacy settings. (a) Conversation info. (b) Verify a contact. (c) Alice verifying Bob. (d) Viber: privacy settings.

3.2.5. Case 5: Riot. Riot is a new chat client that is built on top of the Matrix (https://matrix.org/) protocol for its end-to-end encrypted capabilities. Matrix is an open standard for decentralized communications which uses bridged networks and cross-platform possibilities plus full end-to-end encryption that is based on the Double Ratchet protocol from Signal.

Riot uses servers (the same as Signal), but one does not need to rely on servers under the control of the Matrix team (unlike Signal). Riot and Matrix are open source, which means anyone can set up their own servers with the Matrix implementation and use its end-to-end encryption. This is good for companies that want to have secure chat between employees but do not want to rely on anything outside their own network. Riot also provides group chat, voice (VoIP) and video calling, file transfer, and integration with other applications such as Slack (https://slack.com/) or IRC (https://www.wikiwand.com/en/Internet_Relay_Chat).

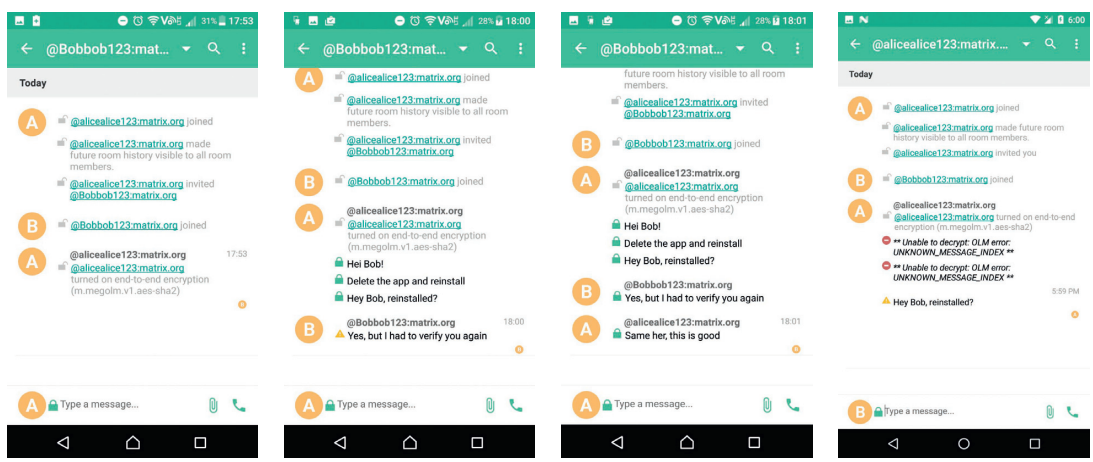
(1) Initial Setup. Riot is the only messaging client that does not rely on a phone number, but a user registers an account with an e-mail address (Figure 15(a)). When a user registers through the app, Riot sends a confirmation link to the entered e-mail address and the user has to click on that link, after which a Captcha verification will be requested for an extra layer of security as shown in Figure 15(b).

(2) Message after a Key Change. Riot is not the typical instant messaging application such as Signal or WhatsApp. Their vision is to make an application which works in the same way as Slack or IRC, where there are chat rooms to join and talk to others. Therefore, Alice starts a chat room, invites Bob, and then activates end-to-end encryption. It should be noted that end-to-end encryption is still in beta form, and thus, it is not turned on by default. Figure 16(a) shows the chat room, which in the beginning has open locks on each of the messages from Alice and Bob that have been sent before



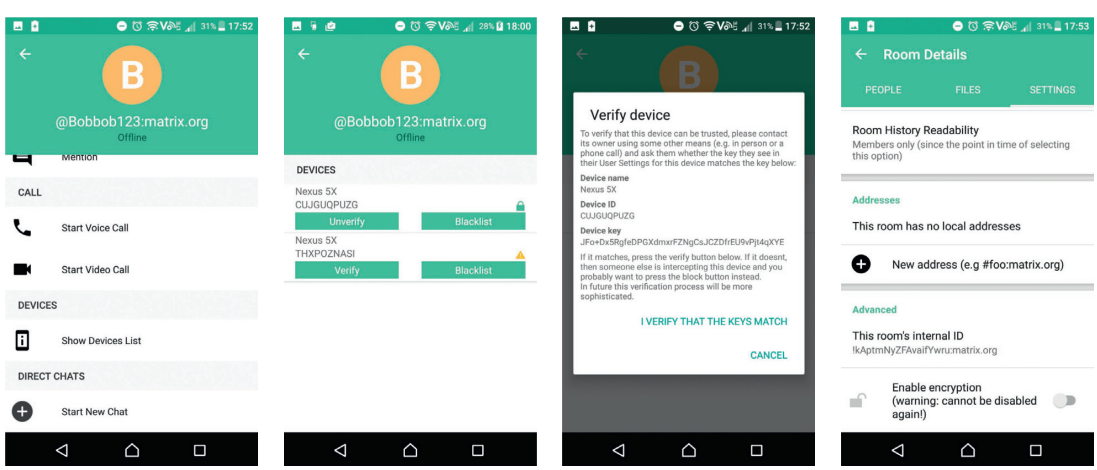
(a) (b) (c) (d)

Figure 15: Riot: registration process and key change while a message is in transit. (a) Registration by e-mail. (b) Captcha code. (c) Message before reinstall. (d) Bob's message after reinstall.



(a) (b) (c) (d)

Figure 16: Riot: message after a key change. (a) Initial conversation, Alice's view. (b) Message to Bob after he reinstalled. (c) New messages are verified. (d) Bob's view of previous message.



(a) (b) (c) (d)

Figure 17: Riot: verification process and other security features. (a) Profile details. (b) Verifying Bob the 2nd time. (c) Verifying Bob. (d) Other security features.

the encryption was toggled on. How the end-to-end encryption is toggled on is shown in the "other security implementations" part (Figure 17(d)). When Alice sends her initial message to Bob, the lock is changed to closed (Figure 15(c)) since E2E encryption is on. Figure 16(b) shows that if Bob reinstalls his application, he has to reverify Alice because his device keys are changed. Alice can also see that she has to reverify Bob as Bob's message has a yellow notification triangle. When Alice verifies Bob, the messages are then listed with a correct closed lock, which means that the messages are encrypted correctly (Figure 16(c)). In Riot, when a new user (or existing user with new keys) enters the chat room, she cannot read/access the previous messages (see Figure 16(d)).

(3) Key Change While a Message Is in Transit. Riot handles key changes in the same way as the previous section regardless of whether the message is in transit or not. When Bob deletes the application, Alice sends the second message to Bob (Figure 15(c)). Alice also sends a third message to Bob when Bob reinstalls the application. As shown in Figure 15(d), Bob can read just the third message but not the second. This shows that Riot handles the key changes in the same way as before, and Bob can see there were some messages sent but cannot decrypt since he lost the old keys.

(4) Verification Process between Participants. The verification process is rather easy in the Riot application. Moreover, Riot gives considerable amounts of information to the user about the users they interact with. When Alice wants to verify Bob's devices, she needs to look at his profile account to find Bob's list of devices by clicking on the "Device" tab, as shown in Figure 17(a). One of Bob's devices in Figure 17(b) has the yellow notification triangle, then that specific device has not been verified by Alice yet. She can either verify the device or put it into the blacklist, as in Figure 17(c), which means that specific device is no longer able to send any messages or invites to Alice.

If Alice decides to verify Bob's device, she clicks on the verify button and sees the verification popup from Figure 17(c). The popup is informative for users, but for some end-users, it may have too much information and could look cluttered. The popup states that the verification information and process will become more sophisticated in future versions when the application starts to reach the end of the beta period. For the verification, Alice can either call Bob or meet him in person and then exchanges the device keys. Finally, Alice needs to press the "I verify that the keys match" button.

(5) Other Security Implementations. Riot does not have that many extra security implementations, but since the application is only in beta, they may be implemented in future versions. Figure 17(d) shows the settings page providing details about a chat room. The administrator of the room (the one who initialized the room) is the only user who can change the settings of the room. The last setting shown in the figure is the option to enable encryption in that specific room. Once encryption is enabled, it cannot be disabled throughout the conversation.

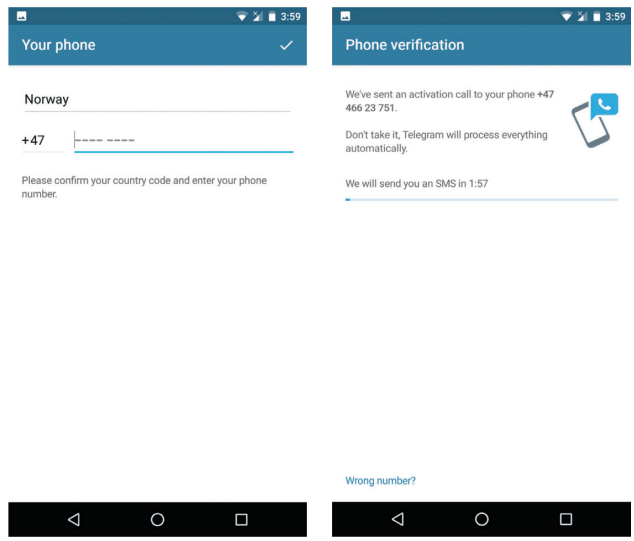
3.2.6. Case 6: Telegram. Telegram is an instant messaging platform which was started in 2013 after the NSA scandal. It has been developed for smartphones, tablets, and even computers (Telegram FAQ https://telegram.org/faq). Telegram allows one-to-one and group communications and the possibility to send files to people in the contact list. The difference between Telegram and the other secure IM applications is that it only offers opt-in secure messaging, while normal conversations are cloud chats that are not end-to-end encrypted. Their motivation is to offer seamless cloud chat synchronization between all connected devices (Seamless chat cloud synd, tweet by Pavel Durov in 2015 at https://twitter.com/durov/status/678305311921410048).

For secure chatting, Telegram implements its own cryptographic protocol called the MTProto Protocol (MTProto Protocol, by Nikolai Durov in Telegram Documentation, available at https://core.telegram.org/mtproto). The same protocol is also used for normal cloud chats to encrypt the communication between the server and the client.

For end-to-end encrypted chats, it is not allowed to screenshot inside the secret chat conversation. Hence, in order to provide images for different test scenarios, we used an external camera.

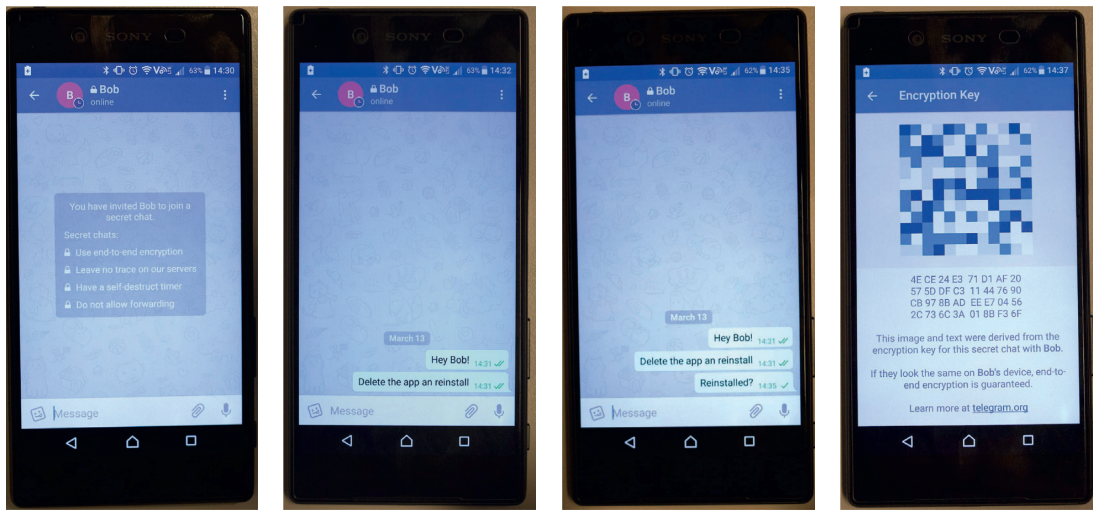
(1) Initial Setup. The initial setup of the Telegram application and user registration is the same for the other applications. Figure 18(a) shows that the user needs to enter her phone number for the registration process. As shown in Figure 18(b), Telegram sends an activation code through SMS, which can either be input manually or give Telegram access to take it automatically. If the verification message is not received in two minutes, a new SMS will be sent. The user also can ask Telegram to call her and activate it through phone call.

(2) Message after a Key Change. The end-to-end encryption is not enabled in Telegram by default. Normal messages, which are called cloud chats on Telegram, are not encrypted. Figure 19(a) shows the first view Alice sees when initiating a secret conversation with Bob, which says that the chat is end-to-end encrypted and the messages cannot be forwarded for security reasons. Figure 19(b) shows the first messages that have been sent from Alice to Bob, and the double checkmarks illustrate that Bob has received and read the message (a single checkmark means that the message has been sent). If Bob reinstalls his application and meanwhile Alice sends a message to Bob, then Bob will never receive that message (even after finishing the reinstallation of the application) as shown in Figure 19(c). Thus, Telegram does not use the previous device keys after reinstalling the application by one of the participants. Hence, Alice needs to start a new secret chat with Bob and send the previous undelivered messages again. Telegram does not store keys, or any other information that could reveal that two users have ever had a secret chat. Therefore, Telegram cannot check whether one of the users has reinstalled the application or not.



(a) (b)

Figure 18: Telegram: registration process. (a) Phone number registration. (b) Verification of phone number.



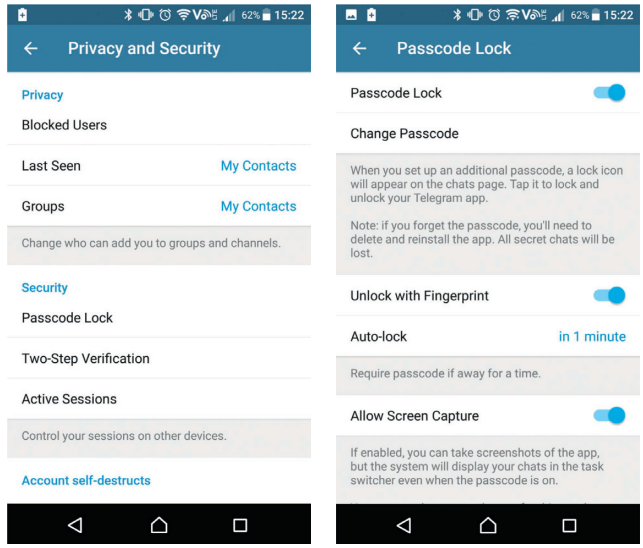
(a) (b) (c) (d)

Figure 19: Telegram: message after a key change. (a) Initial contact. (b) Alice: initial message. (c) Message after reinstall. (d) Telegram: verification process.

(3) Key Change While a Message Is in Transit. As described in the last scenario, Telegram does not store any information about Alice or Bob having a secret chat; all is done by the client and nothing is sent to the cloud of Telegram. Therefore, there makes no sense to test this scenario, as it will have the same outcome as the one before.

(4) Verification Process between Participants. The verification process between Alice and Bob is rather difficult when using secret chats in Telegram. If Alice wants to verify Bob's encryption keys, she needs to open the specific secure chats settings page and then click on the "Encryption Key" button. Telegram just supports messaging and does not support calling for the verification. Figure 19(d) shows the verification page, with an image, which is derived from the encryption key, and the encryption key below. There is no way for Alice to arrive to the conclusion that it is the right image for the conversation.

(5) Other Security Implementations. Telegram supports a few other security features. Inside the settings page, there is one option to look at the "Privacy and Security" settings for the application (Figure 20(a)) Telegram supports two-step verification, i.e., when a user wants to log in on another device or after a reinstall, then they need to write a second, personally chosen, password after the activation code received by SMS. "Active sessions" is a list of devices the user has logged into. The last option, "Account self-destructs," is



(a) (b)

Figure 20: Telegram: other security implementations. (a) Privacy and Security settings. (b) Passcode settings.

a security measurement where if the user has not used their account in the last six months, the account gets deleted by Telegram. The length of the counter for self-destructing can be changed to one month, three months, six months, or one year.

Figure 20(b) shows options under the "Passcode Lock." This function locks the whole application with a passcode that the user chooses. Telegram has implemented the possibility to unlock the application by fingerprint if the user has added a fingerprint in the operating system. A user has the chance to change when the application should autolock, from one minute to five hours. The last option shown is the "Allow screen capture," which (if enabled) allows users to screenshot anything inside the application. However, for secure chats, it is never allowed to screenshot.

4. Summary of Results

The results of the tests are summarised in Table 1, listing what properties each application provides and which are missing.

From this overview, one can conclude that all applications provide mostly the same properties related to the setup and registration phase. All applications except for Riot support registration with the user's phone numbers, whereas Riot needs an e-mail address. Wire supports both phone number and e-mail address, which can also be used later to log in. Access to the SMS inbox is not mandatory in any of the applications, but it is set up as default to make it easier for the user, for otherwise one would need to enter the verification code manually. As Riot does not use a phone number for verification, it does not need access to SMS. Wire also does not have access to the SMS inbox because they believe that it is easy for the user to enter the verification code by hand.

Uploading the contacts list to a server is required by all applications because it enables to find if any of the contacts is already using the application. However, if a user, in order to remain anonymous, does not want to upload her contacts list, she needs instead to only give out her phone number to particular persons in order to communicate.

All the applications (except for Riot that does not use a phone number) have the same properties when it comes to verification by SMS and phone call. They all first give the user the option to verify by reading the SMS and if the user never receives the SMS, then they can ask the application to call them.

Signal and WhatsApp have a trust-on-first-use method, where users trust the other participants in a conversation without verifying first. The other applications ask users to verify each other first in order to be assured that the conversation is secure.

A notification at the start of the conversation would be useful to a new user who does not know what end-to-end encryption is, and having this only in the beginning would not bother the users. Only half of the applications have this notification implemented.

The differences in the way applications are handling key changes are quite big. Only the Signal application had both blocking messages and showed a notification that the other user in the conversation did not have the same cryptographic keys after a reinstall. The blocking message functionality would not allow the sender to send a message before they verify the new cryptographic keys of the receiver if the receiver has generated new cryptographic keys during the conversation. Applications that do not give any notification or block sending of messages could be target for man-in-the-middle attacks, since one of the participants would never get the notification of key changes and thus could not detect any inconsistencies in the hijacked conversation. Wire and Viber are particularly vulnerable to this. The secret chats of

Telegram do not work if cryptographic keys change because the application does not store any information about secret chats. The participants would need to restart the conversation.

The only application that re-encrypts the messages and sends them again after the receiver gets new cryptographic keys was WhatsApp. This is a useful usability property, which we hope it would be implemented by the rest of the applications as well. There is one problem with the way WhatsApp re-encrypts and sends the message again: it never asks the user if it is the correct receiver because the keys have changed.

The "details about the transmission of a message" property questions whether the applications show to the sender that the message is either sent, delivered, or seen by the receiver. If the message is never delivered because of changes to the receivers cryptographic keys, then the message is only tagged as "sent" for the sender, but if the message is re-encrypted and sent correctly, then the message details should also indicate "seen" by the receiver. The only application that does not show any information about whether past messages are sent or delivered is Viber.

Signal and WhatsApp have the easiest verification process, using a QR-code in conjunction with a built-in scanner. However, both have shortcomings since they do not have a check for revealing whether the particular user is already verified. Wire, Viber, and Riot confirmed when a user is already verified, but they did not have the useful QR-code nor any way of sharing the keys outside the application. Telegram was the only application which only offered a QR-code but no way of actually scanning the code. Users had to read the secret keys that are shared between them, whereas the image encoding has no technical way of comparing it, besides by only looking at it. WhatsApp and Telegram have two-step verification capabilities, which means that whenever a user reinstalls the application or changes devices, they need to enter a second password after the normal verification code from the provider, in order to gain access to their account on the new device.

Signal and Telegram both had a passphrase or code that the user had to enter in order to gain access to the application after some specific timeout expires. Both applications have also implemented screen security to not give potential intruders the ability to screenshot conversations. There is a setting to toggle the security off, but it is on by default in both Signal and Telegram.

The only application which had a list of verified contacts, and the option to delete them, was Viber. Clients such as Wire and Riot, which have a verified check on each contact within a conversation, do not offer this option even if it is not difficult to implement since they already know which contacts and their devices are verified.

The "delete devices from account" is only interesting for those applications that support multiple devices. All the applications which supported multiple devices also had a list of devices such that the user could delete a device which is not in use anymore.

5. Discussion and Recommendations

Instead of focusing on one test scenario at a time, like in the previous section, this section discusses and evaluates each application as a whole. Moreover, based on the knowledge gained from the test scenarios, some possible improvements for each application are provided, which have to be verified critically using modelling and verification techniques (besides standard software testing) in order to ensure that an improvement does not break other security properties.

5.1. Signal. The experiments conducted in Section 3.2.1 demonstrate that the Signal app does not have major weaknesses. However, there are several potential improvements that we discuss in the following. Signal showed good understanding and care for the user experience, with an easy verification process. The users can employ QR-codes for the verification purpose and/or can call each other and they can do the verification process through end-to-end encrypted phone calls.

When a party sends a message after changing the keys, the application blocks the message until the sender and receiver verify each other again. This is a useful property, but the application does not reveal the notification immediately when one of the participants in the conversation changes her keys (for example, due to the reinstallation of the application).

Overall, the application has both good security when it comes to end-to-end encryption and useful user experience properties which would not cause problems for new users.

The following provides recommendations for improvement that are applicable to the Signal app. We first state the feature in general terms and then explicit it for the specific app if needed:

(A) Re-encrypt and send lost messages: give the user an option to re-encrypt a lost message and resend it after finishing the reverification process, so that messages would not get lost during a conversation. In the Signal application, the sender of a message can know the status of her messages (i.e., sent or delivered/read) due to the existence of checkmarks on each message, making the implementation of this feature easy.

(B) Notification about key changes: it is recommended for an application to show a notification message immediately after each change of cryptographic keys. If the keys of a party in a conversation change, then the Signal application does not show any notification message to the participants immediately. It only gives the notification (that the keys are changed) when one party wants to send a message to another one (after changing the keys).

(C) Notification on end-to-end encryption: giving a notification at the beginning of each conversation stating that it is now end-to-end encrypted and the possibility to read more about it could help educate the end-users about what E2E encryption is and why they should care about it.

(D) Verified check: offer visual cues so the user can easily know that a party should be verified again to regain the trust properties. As demonstrated by our experiments, in the Signal application, there is no way to know if a user is already verified or not. However, if the application already keeps the list of verified contacts, then when one of the participants changes her cryptographic keys, it should be easy to implement this feature.

(E) Two-step verification: the security of an application would be improved by adding a "two-step verification" option, which, e.g., when a user wants to change her device or reinstall the application, she has to enter a previously chosen password, besides the code received in the SMS.

5.2. WhatsApp. The results of the experiments do not show major weaknesses in the WhatsApp application. However, the WhatsApp application can be improved in several ways as discussed below. WhatsApp takes great care to strengthen the security around the user's account and messages; however, it may suffer from impersonation attacks. Recommendations applicable to WhatsApp are as follows.

(F) Block messaging until verification: in order to prevent sending private messages to an impersonator, the participants should not be able to send any message before verifying each other. The WhatsApp application immediately re-encrypts a lost message when it finds out that the cryptographic keys have changed and the receiver never received the message. Hence, an adversary may impersonate a legitimate contact and consequently WhatsApp re-encrypts and sends lost messages to the adversary who would thus receive private messages in an unauthorized way. Since this process is automatically controlled by the app, even the sender cannot stop re-encrypting and resending of lost messages after a key change. In order to overcome this weakness, the application must suspend the process of re-encrypting and resending of lost messages (after a key change) until the new cryptographic keys are verified.

(G) Locking the application using a passphrase/code: adding an option that requires a passphrase or code before opening the application would enhance the security of the user's account from unauthorized access. If an adversary somehow gets access to a user's phone, she would be unable to access the application messages as she does not know the passphrase/code.

WhatsApp recently has updated its privacy policy, which has caused concern among the users of this messaging application. According to WhatsApp's statements (https://faq.whatsapp.com/general/security-and-privacy/answering-your-questions-about-whatsapps-privacy-policy), their new Terms of Service and Privacy Policy is related to the managing of businesses on WhatsApp, which is an optional feature. It is claimed that all personal communications with friends, family, and so on are still protected by end-to-end encryption and neither WhatsApp nor Facebook can access them. It is also claimed that WhatsApp not only does not maintain logs of calls and messages but also it does not have access to the shared locations. Besides, groups will remain private and neither group data nor contact lists will be shared with Facebook or other apps offered by Facebook.

However, all communications with a WhatsApp business account may be used by Facebook to improve the marketing by means of displaying personalized advertisements on apps offered by Facebook, e.g., WhatsApp, Instagram, and Facebook. For example, Facebook may show an advertisement with a button to communicate with the related business through WhatsApp.

5.3. Wire. The Wire application features several useful security and usability properties. However, there are some properties which are not provided and might cause serious security problems. In this application, if a user uses a new device, Wire does not notify the other participants in a conversation. Thus, an impersonator may join the conversation and receive all the exchanged messages. Recommendations applicable to Wire are (A), (note that the Wire application uses a text under each message to show the status of the message (e.g., delivered or sent), which makes easy to implement this feature) (E), (F), and (G) from above, as well as the following new ones:

(H) Notify users regarding verification: when the application does not verify participants, it should notify users that they have to verify each other manually when initiating a new conversation, to prevent impersonation attacks.

(I) Notification about the verification of new devices: notify the other participants that a user added a new device and they have to verify the new device before sending any more messages (as the new device will also receive these messages). The Wire application allows a user to use the same account on multiple devices. However, if a user (in a conversation) adds a new device, the other participants (in that conversation) will not be notified about this change.

(J) More verification options: provide several different ways of performing the verification. For example, a QR-code or sharing keys with a third-party application are some possible options. Managing keys and authentication credential could greatly benefit from a secure device attached to the smart phone such as the OffPAD [44], in conjunction with a proxy that knows how to use such a device as in the OTDP architecture of [45]. Wire provides only calling a person every time, which may become cumbersome for users.

(K) Screen security: the privacy can be improved by providing a "screen security" option, which does not allow screenshots to be taken within the

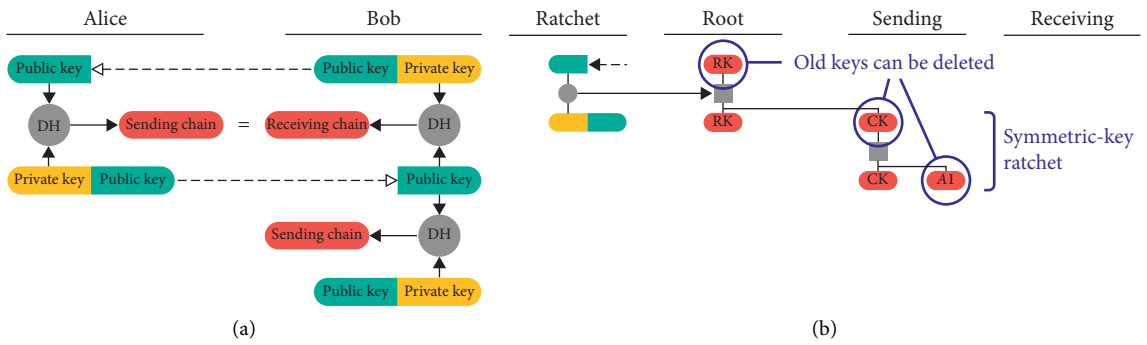


FIGURE 21: Signal Double Ratchet key derivation function (KDF) chains (reproduced from [53]). (a) Sending and receiving chains. (b) First double ratchet message key $A1$ for Alice.

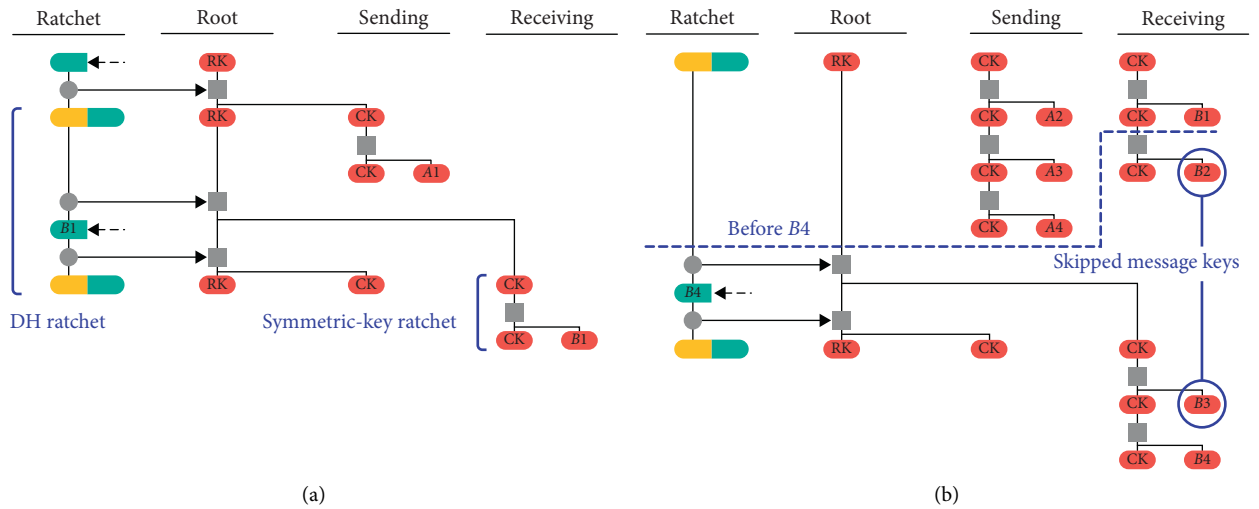


FIGURE 22: Signal more advanced aspects of the double ratchet protocol (reproduced from [53]). (a) First receiving double ratchet message key $B1$ computed by Alice. (b) Handling of out-of-order messages.

conversations. It is worth to mention that the screen security is useful if all participants in a conversation enable it because if a user does not enable this option, then she can take a snapshot and thus breaches the privacy of the other participants in the conversation.

5.4. Viber. The Viber application provides some good usability and security properties. However, there are several questionable aspects which make us reluctant in recommending this application. If the cryptographic keys of a user in a conversation change (e.g., because of reinstallation), Viber does not notify the other participants in the conversation about such changes. In addition, if a user sends several messages while the receiver is reinstalling Viber, then the sender cannot know whether the previous messages have been received or not. Many of the observations for Viber are easy to fix or implement, in our opinion, and would drastically improve the application. These include (A), (B), (E), (F), (G), (H), (J), and (K) from above, as well as the following new one:

(L) Labeling the status of messages: it is important for all messages in a conversation to have a label stating their status (e.g., sent, delivered, and read). In Viber, only the last message in a conversation is labeled with status information.

5.5. Riot. The Riot application is still in beta stage, and despite its usability and security properties, it can still be improved. When cryptographic keys change during a conversation, the previous messages are locked for the user (who is reinstalling the application). However, there is an option for the sender to send the locked messages (encrypted with new keys) again. In Riot, the users do not receive any notification message regarding the key changes (and the need to reverify each other). Moreover, Riot does not use end-to-end encryption by default. In addition, Riot does not provide an easy way for the verification of the users. Recommendations applicable to Riot are (E), (G), (J), and (K) from above.

5.6. Telegram. The telegram application has some useful security properties, but the usability features are rather

lacking and confusing for people who are not tech savvy. The biggest flaw in a secure messaging application is that the end-to-end encryption is not on by default. Hence, an explicit secret chat needs to be initiated every time users want to communicate. We think that this should become a norm these days for an application that advertises encrypted conversations.

In Telegram, if a user sends a message while the cryptographic keys have changed, the intended receiver does not get that message because secret chats are locked to one set of keys. Hence, if a user generates new keys, the participants need to start a new secret chat and cannot continue the previous secret chat. This is good for security, but the problem arises since Telegram does not notify users about key changes.

Telegram provides just one way for the verification of users, showing QR-codes in person, which is not good enough for a secure messaging application. We believe that Telegram can be improved by providing more options, such as calling or sharing keys through third-party applications. Recommendations applicable to Telegram are (B), (D), (J), and (K) from above.

6. Conclusion and Further Work

We have presented the testing and analyses that we have conducted on six secure messaging applications. As a prerequisite, in Section 2, we have started with giving a gentle introduction (following the recent article by Unger et al. [4]) to three secure messaging protocols that offer end-to-end encryption and the types of security and privacy properties they provide. This review of protocols is important for understanding our analysis of the applications implementing them, which is our main contribution presented in Section 3. The Signal and Matrix protocols are both secure messaging protocols that manage end-to-end encryption well, but none of them could offer every security property. The Signal protocol does not fully support multiple devices, while the Matrix protocol does this well. On the other hand, the Matrix protocol does not fully provide forward and future secrecy in the protocol and leaves it up to the implementation to support it. The Signal protocol is designed to use a server for achieving the asynchronous messaging property. Even if messages are encrypted end-to-end, there is still important metadata that is being manipulated and stored on the server. Therefore, one would wish to secure better the server side, especially when deployed in a cloud infrastructure or in a country with legislation that disregards privacy. Initial results in this direction have been presented in [8] using the recent technology of Intel SGX. This is also applicable to Matrix.

The contribution of this paper is the research experiment described in long Section 3 where we have performed five sets of tests on each of the six instant messaging applications that implement one of the two secure messaging protocols Signal or Matrix. We have thus tested 21 properties related to the usability and security of the applications Signal, WhatsApp, Wire, Viber, Riot, and Telegram. The results for the 21 properties on each of the six applications are summarized in Section 4 and listed in Table 1. In conclusion, these applications offer useful usability together with security, and we would strongly recommend to the general public to adopt one of them (i.e., the one most suited for their needs, after reading through our analysis). We believe that even for lay people without special technical skills, it would be rather easy to adopt one of these applications; that is, one would not encounter more difficulties (though this word is rather strong) than with any other chat application, whereas the features related to encryption would not add significant inconvenience. However, several of the tested applications could still benefit from improvements. We have provided 12 recommendations, in Section 5, each one is applicable to one or more of the six applications in order to harden their security while maintaining their useful usability properties.

Appendix

A. Off-the-Record in a Nut Shell

Intuitively, the Off-the-Record (OTR) protocol [10–16] wants to provide for online conversations the same features that reporters want when talking with a news source. Take a scenario where Alice and Bob are alone in a room. Nobody can hear what they are saying to each other unless someone records them. No one knows what they talk about, unless Alice and Bob tell them, and no one can prove that what they said is true, not even themselves. A good thing about an Off-the-Record conversation (in reality) is the legal support behind it since it is illegal to record conversations without participants knowing. It also applies to conversations over the phone, since by law, it is illegal to tap phone lines. There are however no similar laws for communications over the web. OTR-like protocols aim to provide this using cryptography techniques and thus need to provide at least perfect forward secrecy and deniability/repudiation. Full details can be found in [4], (Section 2.4).

Step 1: authenticated key exchange

The latest version of OTR [46] uses a variation of Diffie–Hellman key exchange called SIGMA [38]. Alice and Bob also have long-term authentication public keys pub_A and pub_B, respectively. The point is to do an unauthenticated Diffie–Hellman key exchange to set up an encrypted channel, and inside that, the channel does mutual authentication. The plain Diffie–Hellman key exchange is vulnerable to man-in-the-middle attacks which would break the authentication that OTR needs [12]. Therefore, OTR implements a signature-based authenticated DH exchange, named SIGMA, which solves this weakness [38].

The SIGMA acronym is short for "SIGn-and-MAc," because SIGMA decouples the authentication of the DH exponentials from the binding of key and identities. The former authentication task is performed using digital signatures while the latter is done by computing a MAC function keyed via the common DH secret and applied to the sender's identity [38]. OTR uses a four

message variant known as SIGMA-R, since it provides defence both against active attacks on the responder's identity and passive attacks on the initiator's identity.

Step 2: message transmission

The message transmission step, before sending, performs encryption and authentication of messages using AES [47] in counter mode [48] using message authentication codes (HMAC) [49] for authentication. Using AES in counter mode provides a malleable encryption scheme which allows deniability. Malleability allows to transform a ciphertext into another ciphertext which then decrypts to a related plaintext [12]. This means that a valid ciphertext cannot be connected with neither Alice nor Bob since anyone can create a ciphertext that can be decrypted correctly and then compute a valid MAC from the ciphertext, because old MAC keys are published (more about this in Step 4). Entire new messages, or full transcripts, can thus be forged.

For sending messages, Alice needs to compute an encryption key and a MAC key. The encryption key is used to encrypt the message while the MAC key is used to ensure the authenticity of the message. This method is called encrypt-then-MAC, where usually the encryption key is a hash of the shared secret, EK = Hash (SS), and then the encryption key is hashed a second time to compute the MAC key (MK = Hash (EK)). After the encryption and MAC key are computed, Alice encrypts first the message, $\text{Enc}_{\text{EK}}(M)$and then MACs, the encrypted message, $\text{MAC}(\text{Enc}_{EK}(M), \text{MK})$. Bob computes the same EK and MK from the common secret, used to verify the MAC and decrypt the message.

Step 3: Re-Key

Off-the-Record changes the keys every time the conversation changes directions, to make the duration of vulnerability to attacks as short as possible. Once the new key is established, it will be used to encrypt and authenticate new messages, while the previous ones are erased [12]. After establishing new secrets and keys, the partners erase the old secret SS and encryption key EK, so that no one can forge or decrypt the messages that have been sent. The reason to securely erase this information is to get perfect forward secrecy. The MK key is not erased, but published in the next step.

Step 4: Publish MK

The next step of OTR is to publish the old MAC keys by adding them to the next message that Alice or Bob sends to each other, in plaintext. Alice and Bob both know that they have moved over to MK'; hence, if one of them receives a message with the old MK, they will know that the message has been forged. Publishing MK allows others to forge transcripts of conversations between Alice and Bob. This is useful since it provides extra deniability to both parties [50]. In short, Alice's secrecy relies on Bob deleting the encryption keys, whereas Alice's deniability relies only on Alice publishing her MK.

A.1. Socialist Millionaire Protocol. The problem with secure instant messaging is that there is no way to tell if there has been a man-in-the-middle attack. Therefore, the parties need to make sure they have the same secret which is done using the Socialist Millionaire Protocol (SMP) [51,52]. Intuitively, SMP allows two millionaires who want to exchange information to see whether they are equally rich, without revealing anything about the fortunes themselves. Between Alice and Bob, the SMP allows to know whether $\text{ss}^A = \text{ss}^B$, i.e., the respective computed secrets, without revealing these secrets to anyone [13].

B. Signal in a Nut Shell

Signal is a new end-to-end encryption protocol which has recently seen larger adoption than the Off-the-Record protocol. OTR had an original feature, i.e., refresh the message encryption keys often, which has become known as ratcheting and adopted in Signal as well [6]. The Signal protocol is designed by Moxie Marlinspike and Trevor Perrin from Open Whisper Systems (https://whispersystems.org/) to work in both synchronous and asynchronous messaging environments (advanced ratcheting, by Moxie Marlinspike at Open Whisper Systems, November 26, 2013, https://whispersystems.org/blog/advanced-ratcheting/). The goals of Signal include end-to-end encryption and advanced security properties such as forward secrecy and future secrecy [6]. Initially, Signal was divided into two different applications, TextSecure (https://whispersystems.org/blog/the-new-textsecure/) and RedPhone (https://whispersystems.org/blog/low-latency-switching/). The former was for SMS and instant messaging, while the latter was an encrypted VoIP (https://www.voip-info.org/wiki/view/What+is+VOIP) application. TextSecure was based on the OTR protocol, extending the ratcheting into a Double Ratchet, combining OTR's asymmetric ratchet with a symmetric ratchet [6], and naming it *Axolotl Ratchet*. Open Whisper Systems later combined TextSecure and RedPhone to form the new Signal application that implements the protocol with the same name.

In recent years, the Signal protocol has been adopted by numerous companies, such as WhatsApp (https://whatsapp.com) by Facebook, the Messenger (https://messenger.com) also by Facebook, and Google's new messaging app, Allo (https://allo.google.com).

Signal uses the Double Ratchet algorithm [53] to exchange encrypted messages based on a shared secret key that the two parties have. To agree on the shared secret key, Signal uses the X3DH Key Agreement [54] protocol (standing for extended triple Diffie–Hellman (https://whispersystems.org/docs/specifications/x3dh/)) which we describe later in Appendix Section B.3. Full details can be found in [4] (Chapter 3).

B.1. The Double Ratchet Algorithm. The Double Ratchet Algorithm uses key derivation function chains (KDF chains) [53] to constantly derive keys for encrypting each message,

and it combines two different ratchet algorithms, a symmetric-key ratchet for deriving keys for sending and receiving messages and a Diffie–Hellman ratchet used to provide secure inputs to the symmetric-key ratchet.

A KDF chain is a sequencing of applications of a key derivation function which returns one key used as a new KDF key for the next chain cycle as well as an output key for messages. The KDF chain has the following important properties [53]:

(i) Resilience: the output keys appear random to an adversary without knowledge of the KDF keys, even if the adversary has control of the KDF inputs.

(ii) Forward security: output keys from the past appear random to an adversary who learns the KDF key at some future point.

(iii) Break-in recovery: future output keys appear random to an adversary who learns the KDF key at some point in time, provided the future inputs have added sufficient entropy.

The Double Ratchet generates and maintains keys of each party for three chains: a root chain, a sending chain, and a receiving chain (Alice's sending chain matches Bob's receiving chain, and vice versa; see Figure 21(a)). While the parties exchange messages, they also exchange new Diffie–Hellman public keys. The secrets from the Diffie–Hellman ratchet output become the inputs to the root chain of the KDF chain, and then the output keys from the root chain become new KDF keys for the sending and receiving chains, which need to advance for each sending and receiving message. This is called the symmetric-key ratchet.

The output keys from the symmetric-key ratchet are unique message keys which are used to either encrypt or decrypt messages. The sending and receiving chains ensure that each message is encrypted or decrypted with a unique key which can be deleted after use. The message keys are not used to derive new message keys or chaining keys. Because of this, it is possible to store the message key without affecting the security of other keys, and only the message that belongs to the particular message key may be read if this key is compromised. This is useful for handling out-of-order messages because a participant can store the message key and decrypt the message later when they receive the respective message. See more about out-of-order messages in Appendix Section B.2.

The Double Ratchet is formed by combining the symmetric-key ratchet and the Diffie–Hellman ratchet. If the Double Ratchet did not use the Diffie–Hellman ratchet to compute new chain keys for the sending and receiving chain keys, an attacker that can steal one of the chain keys can then compute all future message keys and decrypt all future messages [53].

Each party generates a DH key pair, a public and a private key, which will be their first ratchet key pair. When a message is sent, the header must contain the current public key. When a message is received, the receiver checks the public keys that are given with the message and do a DH ratchet step to replace the receiver's existing ratchet key pair with a new one [53].

The result is a kind of "ping-pong" behaviour as the two parties take turns replacing their key pairs. An attacker that compromises one message key has little use of it since this will soon be replaced with a new, unrelated key [53]. The DH ratchet produces as output the sending and the receiving chain keys, which have to correspond; that is, the sending chain of one party is the same as the receiving chain of the other party (see Figure 21(a)). Using a KDF chain here improves the resilience and break-in recovery.

Combining the symmetric-key ratchet and Diffie–Hellman ratchet is as follows: (a) when a message is sent or received, a symmetric-key ratchet step is applied to the sending or receiving chain to derive the message key and (b) when a new ratchet public key is received, a DH ratchet step is performed prior to the symmetric-key ratchet to replace the chain keys.

Figure 21(b) shows the information used by Alice to send her first message to Bob. The sending chain key (CK) is used on a symmetric-key ratchet step to derive a new CK and a message key, $A1$, to encrypt her message. The new CK is stored for later use, while the old CK and the message key can be deleted. To ensure that the secrecy throughout the Double Ratchet is upheld, the old root key (RK) is deleted after it has been used to derive a new RK.

Figure 22(a) shows the computations that Alice does when receiving a response from Bob, which includes his new ratchet public key. Alice applies a new DH ratchet step to derive a new receiving and sending chain keys. The receiving CK is used to derive a new receiving CK and a message key, $B1$, to decrypt Bob's message. Then, she derives a new DH output for the next root KDF chain with her new ratchet private key to derive a new RK and a sending CK.

B.2. Out-of-Order Messages. The Double Ratchet handles lost or out-of-order messages by including in each message header the message's number in the sending chain (N) and the length (number of message keys) in the previous sending chain (PN) [53]. This allows the receiver to advance the keys to the relevant message key, while still storing the skipped message keys in case they receive an older message at a later time. Consider the example from Figure 22(b) where we assume that Alice has already received message $B1$, and now she receives message $B4$ from Bob, with the PN = 2 and $N = 1$. Alice sees that she would need to do a DH ratchet step, but first, she calculates how many message keys she needs to store from her current receiving chain (Bob's previous sending chain). Since PN = 2 and her current receiving chain length is 1, the number of stored keys from the current receiving chain is 1 message key (i.e., $B2$). Then, she does a DH ratchet step where a new receiving chain is derived. Because the length of her new receiving chain is 0, she needs to store a message key from her new receiving chain (i.e., $B3$). After Alice has stored $B2$ and $B3$, she can derive the last message key to decrypt message $B4$.

B.3. The X3DH Key Agreement Protocol. For Signal, the X3DH is designed for asynchronous settings where one user, Bob, is offline but has published information to a server, and

another user, Alice, wants to use that information to send encrypted data to Bob [54]. The extended triple Diffie–Hellman key agreement protocol (X3DH) thus establishes a shared secret between two parties who mutually authenticate each other based on public keys and at the same time provides both forward secrecy and cryptographic deniability.

To provide asynchrony, a server is used to store messages from Alice and Bob which later can be retrieved, and the same server keeps the sets of keys for Alice and Bob to retrieve when needed [54]. The X3DH protocol has three different phases:

(1) Bob publishes his elliptic curve public keys to the server: (i) Bob's identity key IK_B, (ii) Bob's signed prekey SPK_B, and (iii) a set of Bob's one-time prekeys (OBK_B^1, OBK_B^2, OBK_B^3, ...). Identity keys need to be uploaded to the server once, while the other keys such as new one-time prekeys can be uploaded again later if the server is getting low. The server will delete a one-time prekey each time it sends it to another user.

(2) Alice fetches from the server Bob's identity key, signed prekey, prekey signature, and optionally a single one-time prekey. If the verification of the prekey signature fails, the protocol is aborted. Otherwise, Alice generates an ephemeral key pair with her public key EK_A and will use the prekey to calculate several DH keys with the purpose to provide mutual authentication and forward secrecy (see details in [54]) used to generate the secret key for encryption (SK). After calculating the SK, Alice will delete her ephemeral private key and the DH outputs to preserve secrecy.

Alice uses the key to send an initial message to Bob containing: (i) Alice's identity key IK_A; (ii) Alice's ephemeral key EK_A; (iii) identifiers stating which of Bob's prekeys Alice used; and (iv) an initial ciphertext encrypted with some AEAD encryption scheme [55] using AD as associated data and using an encryption key which is either SK or the output from some cryptographic pseudorandom function keyed by SK. Alice's initial ciphertext is typically used as the first message in a post-X3DH communication protocol, such as the Double Ratchet protocol in the case of Signal.

(3) Bob receives and processes Alice's initial message. Bob will load his identity private key and the private key (s) corresponding to the signed prekey and one-time prekey that Alice used [54]. Bob repeats the same steps with DH and KDF calculations to derive his own SK and then deletes the DH values, the same as Alice did. Afterwards, he tries to decrypt the initial ciphertext. The decryption is the only difference between what Bob does and what Alice did on her side. If the decryption fails, Bob will delete the SK and the protocol aborts, and the participants need to restart the protocol [54]. If the decryption is successful, he gets the information that Alice had encrypted, and the protocol is complete for Bob. He deletes any one-time prekey private key that was used during the protocol in order to uphold the forward secrecy.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] J. Christian, M. Aulon, A. Hamed, and J. Noll, *Comparing Implementations of Secure Messaging Protocols (Long Version)*, Vol. 475, Department of Informatics, University of Oslo, Oslo, Norway, 2017.

[2] M. Aulon, "A comparison of secure messaging protocols and implementations," Master's thesis, Department of Informatics at the Faculty of Mathematics and Natural Sciences of the University of Oslo2017, Oslo, Norway, 2017.

[3] S. Bruce, *Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World*, W.W. Norton & Company, New York, NY, USA, 2015.

[4] N. Unger, D. Sergej, B. Joseph et al., "SoK: secure messaging," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, pp. 232–249, IEEE, San Jose, CA, USA, May 2015.

[5] F. Tilman, M. Christian, B. Christoph, B. Florian, S. Jörg, and T. Holz, "How secure is TextSecure?" in *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 457–472, IEEE, Saarbrucken, Germany, March 2016.

[6] K. Cohn-Gordon, C. Cas, D. Benjamin, G. Luke, and S. Douglas, "A formal security analysis of the signal messaging protocol," in *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*, p. 466, Paris, France, April 2017.

[7] N. Kobeissi, K. Bhargavan, and B. Blanchet, "automated verification for secure messaging protocols and their implementations: a symbolic and computational approach," in *Proceedings of the European Symposium on Security and Privacy (EuroS&P)*, pp. 435–450, Paris, France, April 2017.

[8] S. Kristoffer, J. Christian, and B. Sergiu, "Securing the endpoints of the signal protocol using intel SGX based containers," in *Proceedings of the 5th Workshop on Hot Issues in Security Principles and Trust (HotSpot 2017)*, pp. 40–47, University of Stuttgart, Uppsala, Sweden, April 2017.

[9] S. Schröder, M. Huber, D. Wind, and R. Christoph, "When SIGNAL hits the fan: on the usability and security of state-of-the-art secure mobile messaging," in *Proceedings of the 1st European Workshop on Usable Security*, Darmstadt, Germany, July 2016.

[10] B. Nikita, I. Goldberg, and E. Brewer, "Off-the-record communication, or, why not to use PGP," in *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, pp. 77–84, ACM, Washington, DC, USA, October 2004.

[11] S. Ryan, K. Yoshida, and I. Goldberg, "A user study of off-the-record messaging," in *Proceedings of the 4th Symposium on Usable Privacy and security*, pp. 95–104, ACM, Pittsburgh, PE, USA, July 2008.

[12] D. R. Mario, G. Rosario, and K. Hugo, "Secure off-the-record messaging," in *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, pp. 81–89, ACM, Alexandria, VA, USA, November 2005.

[13] A. Chris and I. Goldberg, "Improved user authentication in off-the-record messaging," in *Proceedings of the 2007 ACM Workshop on Privacy in Electronic Society*, pp. 41–47, ACM, New York, NY, USA, October 2007.

[14] B. Jiang, R. Seker, and T. Umit, "Off-the-record instant messaging for group conversation," in *Proceedings of the IEEE International Conference on Information Reuse and Integration*, pp. 79–84, Las Vegas, NV, USA, August 2007.

[15] H. Liu, Y. Vasserman Eugene, and H. Nicholas, "Improved group off-the-record messaging," in *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, pp. 249–254, ACM, New York, NY, USA, November 2013.

[16] I. Goldberg, U. Berkant, D. Van Gundy Matthew, and H. Chen, "Multi-party off-the-record messaging," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, pp. 358–368, ACM, Chicago, IL, USA, November 2009.

[17] D. Danny and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[18] D. Tim, "The transport layer security (TLS) protocol Version 1.2," 2008, https://rfc-editor.org/rfc/rfc5246.txt.

[19] E. Justin and M. Cara, *Secure Messaging for Normal People*, NCC Group, New Delhi, India, 2015, https://www.nccgroup.trust/uk/our-research/secure-messaging-for-normal-people/.

[20] S. Harris, *CISSP All-in-One Exam Guide*, McGraw-Hill Education, New York, NY, USA, 6th edition, 2012.

[21] C. Boyd and M. Anish, *Protocols for Authentication and Key Establishment*, Information Security and Cryptography-Springer, New York, NY, USA, 2003.

[22] C. S. Ashraf, Y. Khalid, Al-T Fadi, and Y. Ming-Hour, "A secure and reliable device access control scheme for IoT based sensor cloud systems," *IEEE Access*, vol. 8, pp. 139244–139254, 2020.

[23] C. S. Ashraf, F. Mohammad Sabzinejad, N. Kumar, and H. Alsharif Mohammed, "Pflua-DIoT: a pairing free lightweight and unlinkable user access control scheme for distributed IoT environments," *IEEE Systems Journal*, vol. 99, pp. 1–8, 2020.

[24] I. Azeem, C. S. Ashraf, A. Osama Ahmad, Y. Khalid, and N. Kumar, "A novel pairing-free lightweight authentication protocol for mobile cloud computing framework," *IEEE Systems Journal*, vol. 99, pp. 1–9, 2020.

[25] A. Hamed, N. Morteza, A. Sara, and N. Mahboubeh, "Design and FPGA implementation of an efficient security mechanism for mobile pay-TV systems," *International Journal of Communication Systems*, vol. 30, no. 15, Article ID e3305, 2017.

[26] J. Black, H. Shai, K. Hugo, K. Ted, and R. Phillip, "UMAC: fast and secure message authentication," in *Proceedings of the Annual International Cryptology Conference*, pp. 216–233, Springer, Santa Barbara, CA, USA, August 1999.

[27] J. Black and R. Phillip, "CBC MACs for arbitrary-length messages: the three-key constructions," in *Proceedings of the Annual International Cryptology Conference*, pp. 197–215, Springer, Barbara, CA, USA, August 2000.

[28] D. Gollmann, *Computer Security*, Wiley, Hoboken, NJ, USA, 2011.

[29] I. Azeem and A. Chaudhry Shehzad, "Comment on "ElGamal cryptosystem-based secure authentication system for cloud-based IoT applications"" *IET Networks*, vol. 1880, 2021.

[30] I. Azeem and A. Chaudhry Shehzad, "Comment on "SFVCC: chaotic map-based security framework for vehicular cloud computing"" *IET Intelligent Transport Systems*, vol. 14, no. 12, p. 1723, 2020.

[31] I. Azeem, K. Saru, L. Xiong, W. Fan, A. Chaudhry Shehzad, and A. Hamed, "An improved SIP authentication scheme based on server-oriented biometric verification," *Wireless Personal Communications*, vol. 97, no. 2, pp. 2145–2166, 2017.

[32] A. Chaudhry Shehzad, "Correcting "PALK: password-based anonymous lightweight key agreement framework for smart grid"" *International Journal of Electrical Power & Energy Systems*, vol. 125, Article ID 106529, 2021.

[33] M. Moxie, *Advanced Ratcheting*, Open Whisper Systems, San Francisco, CA, USA, 2013, https://whispersystems.org/blog/advanced-ratcheting/.

[34] I. Goldberg, "Off-the-record messaging," 2020, https://otr.cypherpunks.ca/.

[35] D. Roger, M. Nick, and S. Paul, *Tor: The Second-Generation Onion Router*, Naval Research Lab Washington DC, Washington, DC, USA, 2004.

[36] McC. Damon, K. Bauer, G. Dirk, T. Kohno, and S. Douglas, "Shining light in dark places: understanding the tor network," in *Privacy Enhancing Technologies*, pp. 63–76, Springer, Berlin, Germany, 2008.

[37] A. Balducci and J. Meredith, *Olm Cryptogrpahic Review*, NCC Group PLC, Manchester, UK, USA, 2016, https://www.nccgroup.trust/us/our-research/matrix-olm-cryptographic-review/.

[38] H. Krawczyk, "SIGMA: the "SIGn-and-MAc" approach to authenticated diffie-hellman and its use in the IKE protocols," in *Proceedings of the 23rd Annual International Cryptology Conference—CRYPTO*, pp. 400–425, Springer, Santa Barbara, CA, USA, August 2003.

[39] H. Matthew, "Encrypting matrix: building a universal end-to-end encrypted communication ecosystem with matrix and olm," in *Proceedings of the Free and Open Source Software Developers' European Meeting*, Buenos Aires, Argentina, May 2017.

[40] R. Zimmermann Philip, *The Official PGP User's Guide*, MIT Press, Cambridge, MA, USA, 1995.

[41] S. Garfinkel, *PGP: Pretty Good Privacy*, O'Reilly Media, Inc., Newton, MA, USA, 1995.

[42] A. Eric, *Go Ahead, Make Some Free, End-To-End Encrypted Video Calls on Wire*, Reuters, London, UK, 2016, http://www.reuters.com/article/us-dataprotection-messaging-wire-idUSKCN0WC2GM.

[43] L. Ingrid, "Viber follows messenger, launches public accounts for businesses and brands," 2016, https://techcrunch.com/2016/11/09/viber-follows-messenger-with-the-launch-of-public-accounts-for-businesses-and-brands/.

[44] M. Denis, J. Christian, and J. Audun, "DEMO: Off-PAD—offline personal authenticating device with applications in hospitals and e-banking," in *Proceedings of the 23rd Conference Computer and Communication Security*, pp. 1847–1849, ACM, Vienna, Austria, October 2016.

[45] M. Denis, J. Christian, and J. Audun, "Offline trusted device and proxy architecture based on a new TLS switching technique," in *Proceedings of the International Workshop on Secure Internet of Things (SIOT) IEEE*, Oslo, Norway, September 2017.

[46] I. Goldberg, D. Goulet, A. Jacob, and B. Jurre, *Off-the-Record Messaging Protocol Version 3*, University of Waterloo, Ontario, Canada, 2012, https://otr.cypherpunks.ca/Protocol-v3-4.0.0.html.

[47] J. Daemen and R. Vincent, *The Design of Rijndael: AES—The Advanced Encryption Standard*, Springer, Berlin, Germany, 2002.

[48] J. Dworkin Morris, *Recommendation for Block Cipher Modes of Operation: Methods and Techniques. SP 800-38A 2001 Edition*, National Institute of Standards & Technology, Gaithersburg, MD, USA, 2001.

[49] B. Mihir, C. Ran, and K. Hugo, "Keying hash functions for message authentication," in *Proceedings of the 16th Annual International Cryptology Conference—CRYPTO*, pp. 1–15, Springer, Santa Barbara, CA, USA, August 1996.

[50] M. Moxie, *Simplifying OTR Deniability*, Open Whisper Systems, San Francisco, CA, USA, 2013, https://whispersystems.org/blog/simplifying-otr-deniability/.

[51] J. Markus and Y. Moti, "Proving without knowing: on oblivious, agnostic and blindfolded provers," in *Proceedings of the 16th Annual International Cryptology Conference—CRYPTO*, pp. 186–200, Springer, Santa Barbara, CA, USA, August 1996.

[52] C. Yao Andrew, "Protocols for secure computations," in *Proceeings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pp. 160–164, IEEE, Chicago, IL, USA, November 1982.

[53] M. Moxie and T. Perrin, *The Double Ratchet Algorithm*, Open Whisper Systems, San Francisco, CA, USA, 2016, https://whispersystems.org/docs/specifications/doubleratchet/.

[54] M. Moxie and T. Perrin, *The X3DH Key Agreement Protocol*, Open Whisper Systems, San Francisco, CA, USA, 2016, https://whispersystems.org/docs/specifications/x3dh/.

[55] R. Phillip, "Authenticated-encryption with associated-data," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 98–107, ACM, Washington, DC, USA, November 2002.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9997771, 20 pages
https://doi.org/10.1155/2021/9997771

Review Article

Security in Vehicular Ad Hoc Networks: Challenges and Countermeasures

Jabar Mahmood,[1] **Zongtao Duan**,[1] **Yun Yang**,[1] **Qinglong Wang**,[1] **Jamel Nebhen**,[2] **and Muhammad Nasir Mumtaz Bhutta**[3]

[1]*School of Information and Engineering, Chang'an University, Xi'an 710064, China*
[2]*Prince Sattam Bin Abdulaziz University, College of Computer Engineering and Sciences, P.O. Box 151, Alkharj 11942, Saudi Arabia*
[3]*Department of Information Systems, College of Computer Sciences and Information Technology, King Faisal University, Al Ahsa, Saudi Arabia*

Correspondence should be addressed to Yun Yang; yangyun@chd.edu.cn

Received 10 March 2021; Accepted 21 June 2021; Published 30 June 2021

Academic Editor: Azeem Irshad

Recently, vehicular ad hoc networks (VANETs) got much popularity and are now being considered as integral parts of the automobile industry. As a subclass of MANETs, the VANETs are being used in the intelligent transport system (ITS) to support passengers, vehicles, and facilities like road protection, including misadventure warnings and driver succor, along with other infotainment services. The advantages and comforts of VANETs are obvious; however, with the continuous progression in autonomous automobile technologies, VANETs are facing numerous security challenges including DoS, Sybil, impersonation, replay, and related attacks. This paper discusses the characteristics and security issues including attacks and threats at different protocol layers of the VANETs architecture. Moreover, the paper also surveys different countermeasures.

1. Introduction

Aiming at ensuring the safety and facilitating the passengers and driver, the VANETs are getting much popularity and attention from the researchers [1–3]. VANETs are the networks of vehicles communication and road infrastructures to extend road safety and infotainment [4]. The wireless sensors are fitted within vehicles, accompanied with positioning devices and maps. Through On-Board Unit (OBU), the vehicles are connected with road-side units (RSUs) to share intervehicle and vehicle to RSU, the safety related and otherwise information [5, 6]. The VANETs consist of short-range communication infrastructure. Therefore, the source and destination share information through intermediate nodes. Like OBU, RSU, the trusted authority (TA) is also an entity of the VANETs architecture and is responsible for controlling and supervising the whole network [7, 8].

The remaining paper is ordered as follows: Section 2 explains the VANTEs overview in detail and describes the characteristics of VANETs. Section 3 is divided into two parts. The first part provides detailed security issues in VANETs, the security attacks on the physical layer; the second part presents other security attacks on different layers of VANETs and also describes the protocol layers threat. Section 4 describes the various challenges and solutions in VANETs, and Section 5 concludes the article.

2. Overview of VANETs

The VANETs architecture contains the OBU, RSU, and TA. There are two types of communication technologies in VANETs architecture, i.e., (1) vehicle to vehicle (V2V) and (2) vehicle to infrastructure (V2I) communication as shown in Figure 1. V2V contact vehicles converse with one another and exchange the traffic-related information inside the

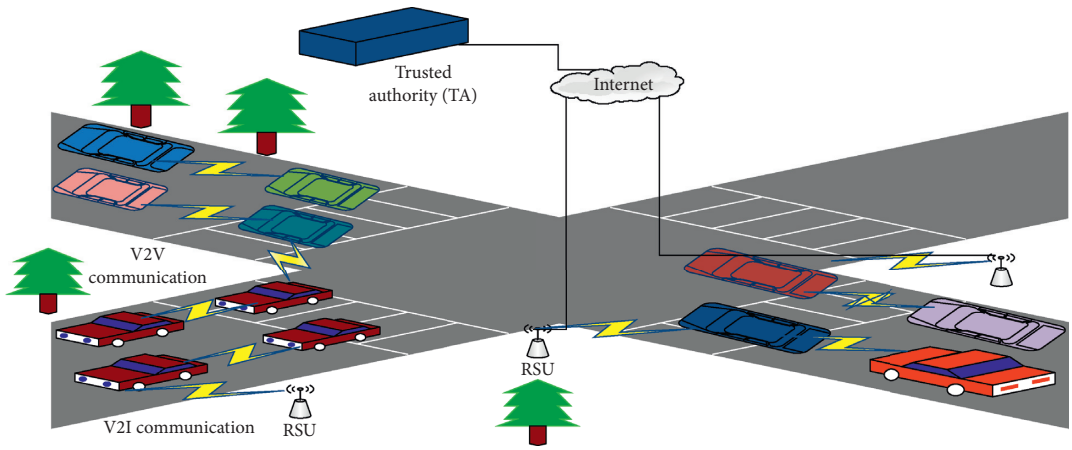


Figure 1: VANETs architectures.

wireless network range [3, 9, 10]. In such networks, when any unforeseen incident happens, such as accident or traffic blockage on the road, instantly a vehicle sends an alert signal to the other nodes or vehicles in the network suggesting to avoid that particular road or area. The vehicle, employing V2I communication, shares the information with RSU which is part of infrastructure installed on the road. The V2I-based communication notifies the driver about traffic and weather updates to keep an eye on the nearby environment [3, 9, 11]. RSU and OBU are registered by a trusted authority [12, 13], which is used to keep up and supervise the VANETs system. The road-side unit positions itself on the road for authentication and communication between TA and OBU. With the use of dedicated short-range communication (DSRC) [6], the OBU fitted in each vehicle can transmit traffic information to nearby vehicles and RSU [10].

2.1. Characteristics of VANETs. VANETs is a dynamic ad hoc network that enables the vehicles converse with one another using fixed and mobile nodes offering numerous services, however with narrow access to the network's infrastructure. Compared to the MANETs, the VANETs have high mobility features and normally vary in topology [10]. In VANETs, vehicles or nodes move arbitrarily in the network, and their movement transforms the network topology. VANETs topology is complex and dynamic because of the strong mobility factor of nodes [9]. The features of VANETs are mentioned below.

2.1.1. High Mobility. Because of the high mobility, the VANETs have good versatility relative to MANETs, and they play a significant role in modelling VANETs protocol. In VANETs, every node moves quickly; thus, vehicles' mobility minimizes the communication time in the network [10, 14].

2.1.2. Driver Protection. The VANETs might get better driver protection, improve traveller console, and support a better flow of traffic. The core benefit of VANETs is that nodes communicate straight to everyone [10].

2.1.3. Vibrant Network Topology. In VANETs, the topology design is vibrant because the vehicle speed of mobility is very high. Therefore, the forecast of node position is very tough to compute. The high speed of vehicle networks is extra weak to attacks, and it is incredibly complicated to identify intruders and vehicles if something is wrong in a network [10].

2.1.4. Variable Network Density. Due to high-speed mobility vehicles, traffic congestion or even lousy weather, the network may experience frequent or intermittent disconnection among nodes. In this situation, the nodes may receive proper guidance from the V2I infrastructure [9, 10].

2.1.5. The Medium of Transmission. Due to open wireless nature, these kinds of networks inherit all security vulnerabilities as posed to other traditional wireless networks [10].

2.1.6. No Power Limits. In MANETs, power is a grave problem; however, in VANETs the power is not a big problem since the OBU in vehicular entities bear sufficient battery and power resources necessary to carry out its communicative tasks [9, 14].

2.1.7. Restriction of Transmission Power. Wireless Access to Vehicle Environment (WAVE) limits the transmission power, which varies from 0 to 28.8°dBm with the corresponding coverage distance ranging from 10°m to 1 km. Thus, the narrow power transfer may change the distance from the VANTS coverage [10].

2.1.8. Network Strength. The signal strength of the network in VANETs depends on the traffic congestion, since it might gain more strength if there is no congestion or less traffic on the road. On the other hand, in case of traffic jams the signal quality might experience degradations [10].

2.1.9. Extensive Scale Network. In VANETs, the network is highly scalable, since such kind of networks may experience highways, downtown areas, point of entries, and exit in the

cities; hence, a massive number of nodes can be dynamically added or adjusted into the system [9, 10].

2.1.10. Extensive Computational Processing. In VANETs, a large number of resources such as processors, colossal memory GPS, and antenna are embedded in vehicles. Such resources may require a massive computational capabilities and guidance to provide enhanced and trustworthy wireless communication for getting accurate information, i.e., live location, speed, and route of the vehicle [9, 10].

3. Security Issues in VANETs

The security issue is very crucial in VANETs which ensures safety for the drivers as well as passengers. This is obligatory to design essential algorithms to assure safety and protection. The security challenges as posed to VANETs are availability, authentication, integrity, confidentiality, nonrepudiation, pseudonymity, privacy, mobility, data and location verification, access control, and key management issues [9, 10, 15, 16].

3.1. Security Issues. In this section, we provide details about various security issues in VANETs.

3.1.1. Availability. Availability [17] is considered a significant factor in VANETs security. This ensures that all resources are accessible forever in a network in the face of vulnerabilities and denial of service attack-based attempts. Cryptography and trust-based algorithms and protocols are helpful to protect the VANETs from these attacks [9, 10, 17, 18].

3.1.2. Authentication. Authentication enables the right participants to enter the network after dual verification. It also ensures that the sender or user who sends a message is not an intruder. Besides, the privacy of the user is preserved using pseudonyms [17–19].

3.1.3. Integrity. Integrity or data integrity ensures that there is no change in the original data packets sent by the sender. Alternatively, it must be protected from the adversary on the way. Data accuracy is one of the fundamental security issues in VANETs. Digital signature, public key infrastructure, and cryptography revocation mechanism may be employed to ensure the integrity between the sender and receiver [9, 10].

3.1.4. Confidentiality. Confidentiality means to hide data from adversaries. In confidentiality we make sure only authenticated users access the data with the help of encryption and decryption. In this way the data remains confidential, while the other unauthorized users may not access this confidential information [9, 20].

3.1.5. Nonrepudiation. This feature ensures that the source of the originating message may not deny the fact that it has generated a particular message. Alternatively, this feature binds the content with the originator of a particular message. [9, 10, 19].

3.1.6. Pseudonymity. The pseudonymity refers to hiding the original identity. The legal participants may use pseudonyms instead of using original identities. In this manner, the legitimate entities may communicate anonymously without revealing their true identities. This ensures protected privacy for the subscribers [18].

3.1.7. Privacy. In VANETs, the privacy refers to concealing driver identity as well as the location's information from other unauthorized users in the network [9, 18, 21].

3.1.8. Scalability. The capability of the network to respond to the dynamically changing requirements is termed as scalability. The frequently changing topology of the vehicular network is another challenge for the researchers [18].

3.1.9. Mobility. Mobility is ubiquitous in VANETs because nodes communicating in VANTEs change their location very quickly and frequently in a network. VANETs nature is dynamic because every second, the node position is changed. This mobility factor focuses on the need of more secure and dynamic algorithms maintaining quality of service requirements [18].

3.1.10. Data Verification. It is used to eliminate malicious messages in the network. This ensures to test the accuracy of data and verify the legitimacy of participating nodes [9].

3.1.11. Access Control. Access control is used to monitor and check the policy rights and roles for all participating nodes in the network [9, 15].

3.1.12. Key Management. Key management refers to the key used in encryption or decryption process during communication between the nodes. The key management and issuance are resolved during the designing of security protocols for the network [18, 22].

3.1.13. Location Verification. A reliable mechanism for the verification of location is required in VANETs, because this is necessary to protect from various attacks during communication and is also helpful in the data validation process [18].

3.2. Attacks on the Physical Layer of VANETs (Security Attacks in VANETs). This section on attacks in VANETs can be divided into three parts. In the first part, we discuss the attackers based on their nature, behaviour, and efficiency. In the second part, we discuss the various attacks on physical layer, and the third part focuses on the rest of the attacks in VANETs. Now we discuss the types of the attackers according to nature, behaviour, and efficiency:

(i) Active vs. passive: in the case of an active assault, the assailant gets the information from the network, changes the original message's information, and forwards it to the receiver. Usually, in an active assault, the assailant wants to decrease the network's efficiency or get access to the network for unauthorized services [23]. In the case of the passive assault, the assailant does not send or receive any message on a network by eavesdropping the wireless network and collecting information about the network or seeking potential vulnerabilities [24, 25].

(ii) Insider vs. outsider: insider attacker means that the authorized member who is part of the network has full information about the network and can access network efficiently. On the other side, outsider attackers are intruders who are not authorized and cannot access the network directly. That is, if they want to initiate an attack they must collect knowledge about the network first and then attack [24, 26, 27].

(iii) Malicious vs. rational: the attacker's intention is to attack the network and gain personal benefits. A malicious attacker may upset the network's performance with an objective to affect the legal users of the network [23, 28]. On the other hand, a rational attacker may intentionally launch an attack on the network to get some information in order to damage the network [24, 26].

(iv) Local vs. extended: in the case of local attackers, they launch attacks on a limited scope and cover the limited area or region like some RSU and node [27]. However, extended attackers cover bigger region or area comparatively. The extended attacker aims to degrade the network's performance or shut down the whole network [25].

3.2.1. Eavesdropping Assault. Eavesdropping assault is a type of passive assault and is done in the privacy of the network. Assailant collects the secret information, and the attacker secretly monitors the traffic flow of the network or the existing location and actions of a specific vehicle. This type of assault cannot be detected easily because the attacker performs its activity without any kind of reaction [25, 29]. Figure 2 shows that Car C regularly monitors ATM's cash van's facts and leaks such information to the intruder. ID revelation assault is a subcategory of eavesdropping where the assailant exposes the identity vehicle and uses it to track the under-attack vehicle.

3.2.2. Denial of Service Assault. In DoS-based assault [30–32], the assailant attacks the service provider's services. In this attack, even the legitimate users are unable to acquire services in the network. The assailant may initiate this attack any time and jam the communication channel. This kind of assault can be launched in two ways. On the first hand, the attacker may engulf the resource with numerous requests, while that resource may not be able to respond to legal user requests. This type of attack can be extended by sending a large number of requests for messages and jamming the communication. Therefore, RSU cannot accommodate several requests that OBU might have submitted [29, 33, 34]. In Figure 3, a DOS attack is demonstrated where auto F is an attacker in the car who denies access to RSU services for users of Cars A, C, D, E, and H.

3.2.3. Distributed Denial of Service Assault. Distributed Denial of Service (DDOS) [35] could be more damaging for the ad hoc vehicular environment since the attacker may attack the network in a distributed manner. An attacker may use various time slots for different vehicles to submit a message. The only objective of the assailant is to bring the network down [27, 29]. In Figure 4, a Distributed Denial of Service attack is demonstrated where the two cars, i.e., Car Q and U, attack the services provided by RSU, while the Cars M, N, O, and P denied the attackers Q and R, S and T, deprived of access to RSU services by the car in the attacker.

3.2.4. Illusion Assault. It involves deception with the manipulation of vehicle's inside information, for instance, speed and location, by tampering the hardware physically. By providing the wrong information of vehicles using internal devices or sensors, it misguides the other network nodes. For instance, it may show another person by cloning the location of the other vehicle [25].

In the case of in-transit traffic tampering assault, a malicious node may deliberately cause delay, corruption, replay, or alteration of a message to spoil the VANETs communication. This type of replay assault [36, 37] includes message replay where the assailant records the message received from certified nodes and then resends after sometime to create some misunderstanding or disturb the traffic. In Figure 5, it is shown that the attacker spoofs the message and sends back to the node; the original message was created by "M" assailant to create misunderstanding and replacing it as "tn." This assault could be launched in two ways, one is using an on-board unit by using a particular part of the hardware. The duplicate messages remain unsuccessful in locating the neighbor 's accurate driving status, for example, speed, location, direction, etc. [25, 38].

3.2.5. Message Modification/Alteration. In a message modification attack, the attacker changes the information of the vehicle integrated into a message (for example, speed, position, or direction) for its own benefit. It is a potential hazard for the security of the other nodes in the network [25].

3.2.6. Jamming Assault. An assailant intentionally generates large amount of messages in a network and creates congestion on wireless channel that might affect the performance of network [25]. The assailant may initiate jamming attack by transmitting a strong radio signal to interrupt the entire communication by declining the signal to noise ratio.

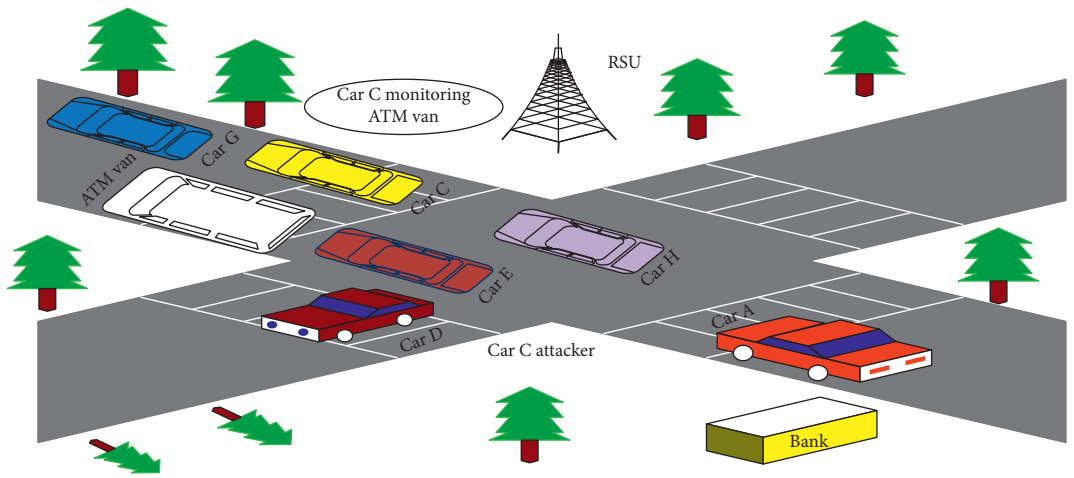


Figure 2: Eavesdropping assault.

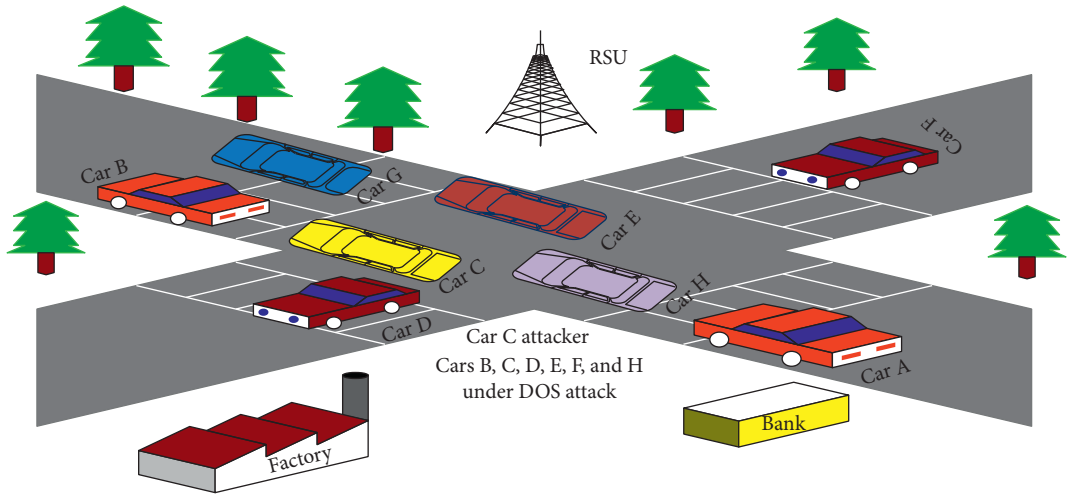


Figure 3: DOS assault.

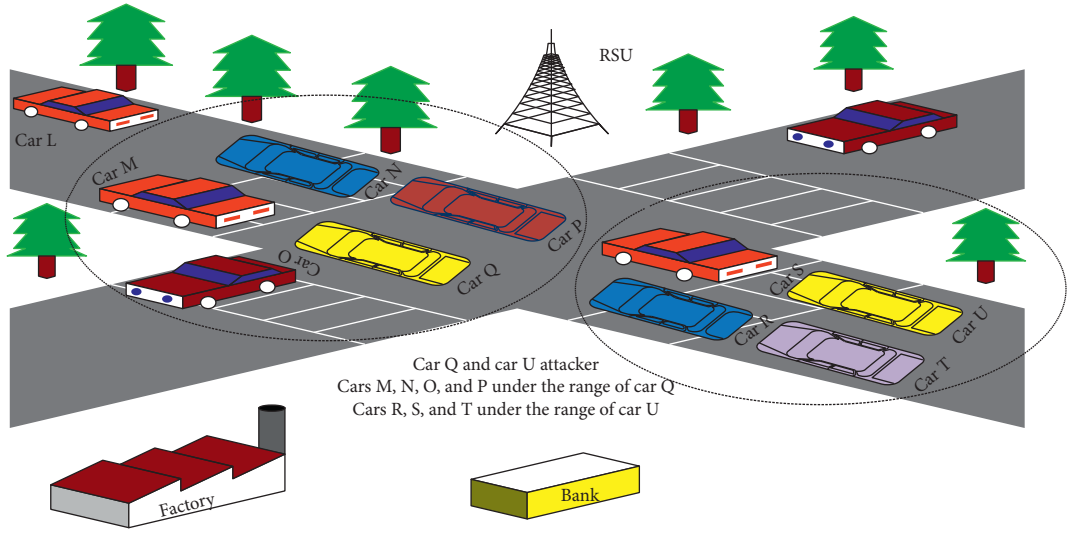


Figure 4: DDOS assault.

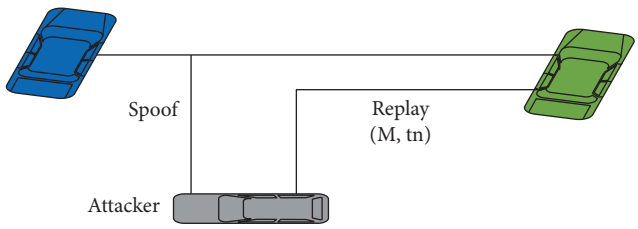


Figure 5: Message replay.

In this, the jammer continuously sends a signal by interfering with the communication of other vehicles in a network. In VANETs, jamming is considered a big threat for its security. Figure 6 shows that the assailant is jamming the network. The victim nodes are always perceived to be busy in a network, since they are unable to send or receive messages in this jammed area. When jamming signal is enabled, the sender sends the data packet, and the receiver does not receive the intended data packet. Therefore, the packet delivery ratio (PDR) is meager. These data packets carry essential information, such as weather conditions, road conditions, accidents, etc. Many incidents may happen if that critical information is not delivered to the nodes in due course of time.

3.3. Other Attacks in VANETs. In this section, we discuss remaining assaults that occurred on VANETs layers during communication.

3.3.1. Sybil Assault. In Sybil assault [39, 40], the assailant generates numerous identities of vehicles and broadcasts the incorrect information on the network. In the case of Sybil assault, data are broadcasted with fictitious identity. This assault is implemented from an OBU upon other OBUs after authentication for acquiring personal benefits. According to this scenario, the assailant creates several identities and sends a message in a network to the authentic user, such as additional traffic on the road, and therefore alters a route. One delusion is generated by the assailant, and the same message is sent to various vehicles. The authentic user will receive the same data packets from various vehicles because the illusion is always created in a network and believes its node will alter the route. This decision goes in favour of the attacker, while the route becomes clear, thus the attacker enjoys the trip [29, 41]. Figure 7 represents a Sybil assault in which an assailant in Car C creates numerous identities and sends those data packets with false identities to other users, which creates an illusion that the road has enormous traffic. After receiving such data packets, Car B and Auto D may decide alternative routes, and, currently, Car C gets a free road.

3.3.2. Node Impersonation Assault. Node impersonation attack is another name for a message tempering attack [29]. In VANETs, every vehicle has a unique identifier and uses it to send the message and verify if something wrong happens in the network. In node impersonation assault, the assailant changes the original data packet and claim that the data packet comes from a genuine user [27, 29]. Figure 8 shows that Vehicle D sends messages about the mishap to place x before acquiring help. However, the assailant junction C will inform the data packet and forward it to the ambulance to happen at place Y.

3.3.3. Black Hole Assault. Black hole assault [42–45] is a category of routing assault in which a malicious node attracts the victim's node on the network. Furthermore, it assures transmitting data through it by presenting the shortest path to the receiver node [29, 46]. The victim node chooses that shortest path and sends the data packet; any malicious node may drop the message or misuse the message for its own [41, 47, 48]. Figure 9 depicts that Car K desires to submit messages to Car P and Car Q, but it has no routing path for those nodes. Therefore, Car K activates the route detection process. Route request is redirected to Car B and Car L. Now, a malicious vehicle, Car L, claims that it has the shortest route to arrive at Car P and Car Q. According to the availability response, Car K sends every data packet to Car L and becomes a black hole assault victim.

3.3.4. Worm Hole Assault. Worm hole assault [49] is another type of routing assault. In a worm hole attack, a malicious node receives the message from the authenticated user at any place in the network, and, with the help of another malicious node, it creates a tunnel between two malicious vehicles [29, 46]. Figure 10 shows a wormhole assault in VANETs.

3.3.5. Gray Hole Assault. Gray hole assault is an extended version of black hole assault, wherein the malicious node also shows itself as part of the network. It sends a request message to victims' nodes and shows as the shortest path route node; in gray hole attacker [50] also received the data packets but did not drop all packets like black hole attack. It only dropped few data packets. In Figure 11, Car H shows that part of the network and presents the shortest path for communication to Car G. It is complicated to identify this type of attack because it is not continuous. It is created for limited time period for a specific purpose [29].

3.3.6. Masquerading Assault. In a masquerading attack, the attacker sends packets on behalf of other vehicles by using the identity of those vehicles [51]. In Figure 12, the C shows itself as a police van, and, through that deception, the node makes the other nodes reduce their speed or stop the node.

3.3.7. Global Positioning System Spoofing Assault. Global positioning system spoofing attack is another name for location faking assault. According to this category of assault, the assailant tries to vary their present location identity and forward fake information from the GPS by using such a method, by not showing the existing location to other nodes and pretending to be in an incorrect location to others. This

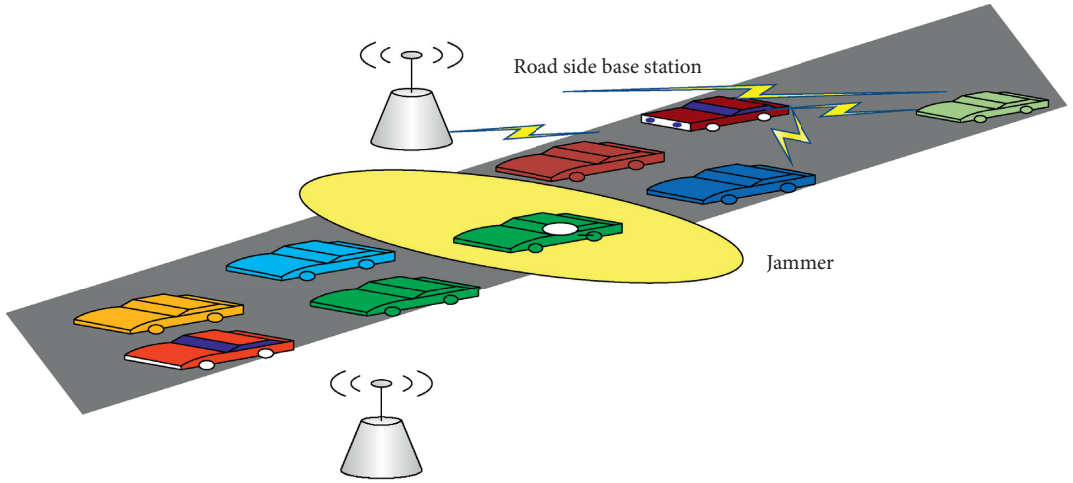


Figure 6: Jamming assault.

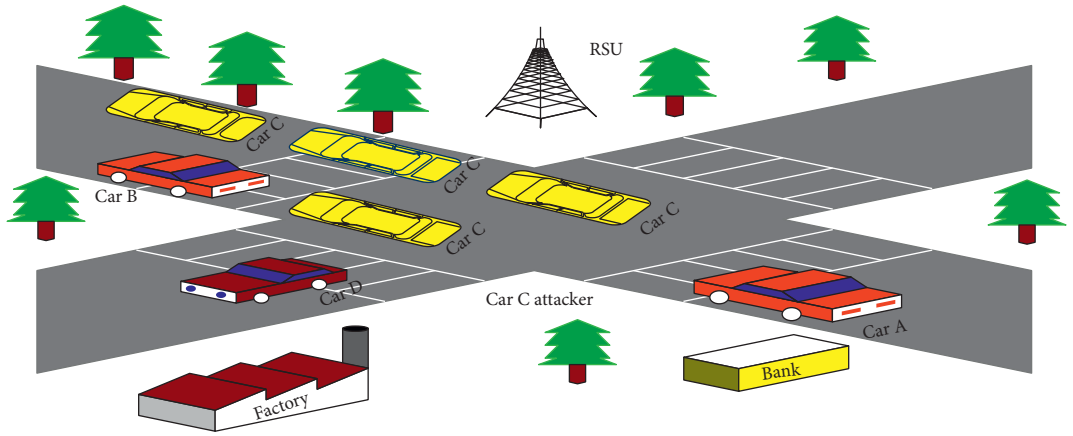


Figure 7: Sybil assault.

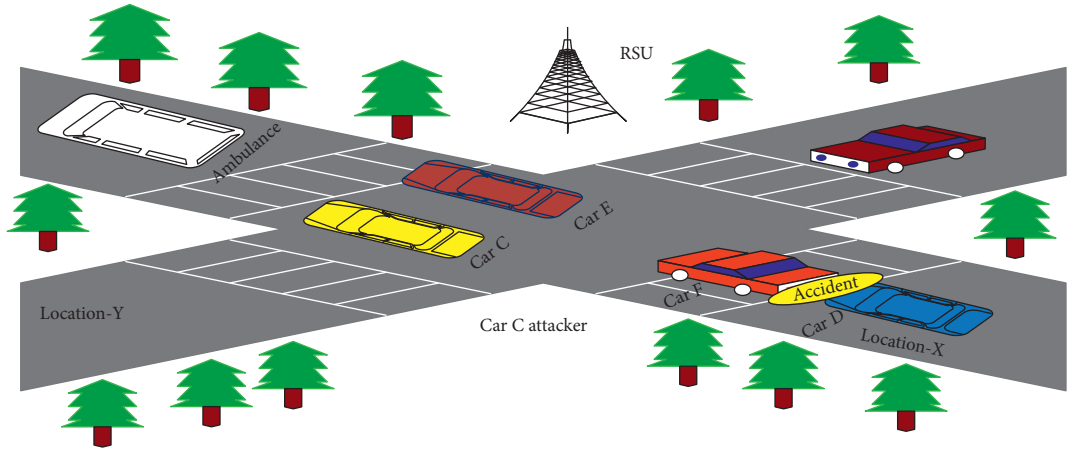


Figure 8: Node impersonation assault.

assault is done by the attacker with the help of set of nodes [29]. In Figure 13, three nodes are moving on the Road-ID 8; however, they do not show their present location and forward the network's incorrect information. RSUs acquiring such details show that there is no node on Road-ID 8.

3.3.8. Brute Force Assault. In the ad hoc network, the sender vehicle sends the message to the receiver vehicle with the help of other nodes if the receiver vehicle is beyond its range. Thus, for the sake of security, the sender nodes or vehicles encrypt the message and submit towards the target via any

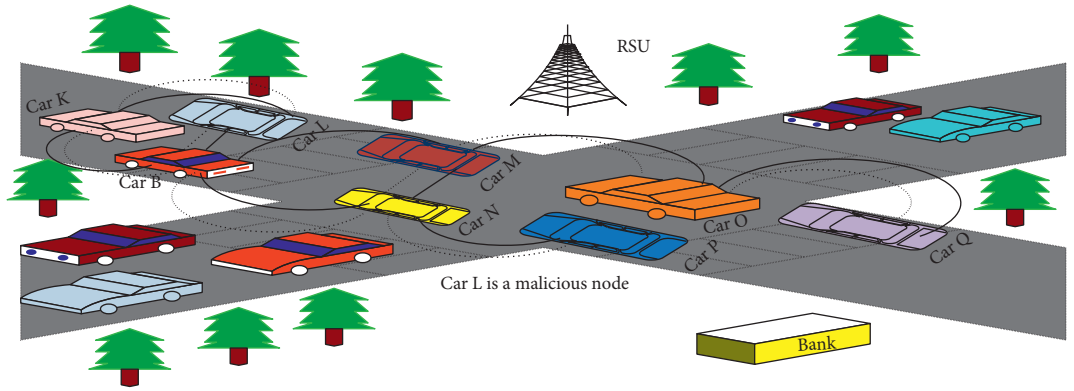


Figure 9: Black hole assault.

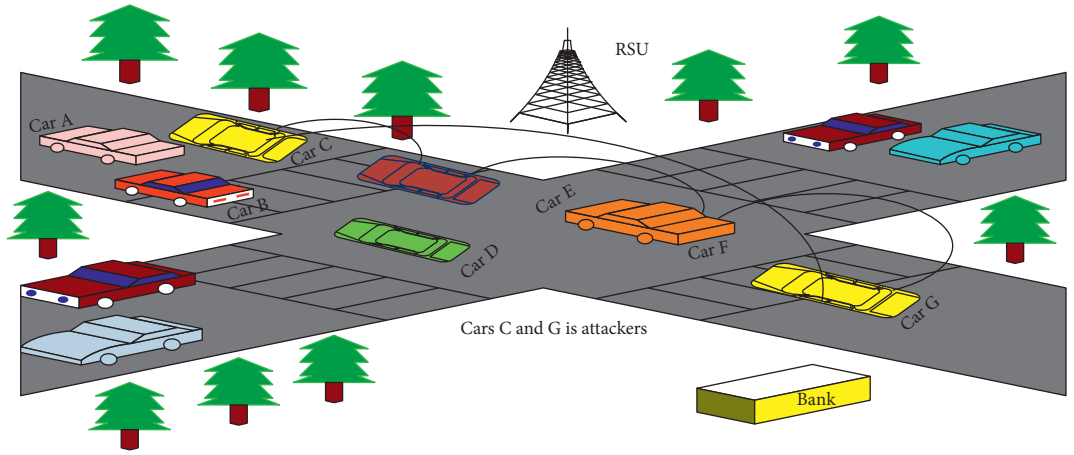


Figure 10: Worm hole assault.

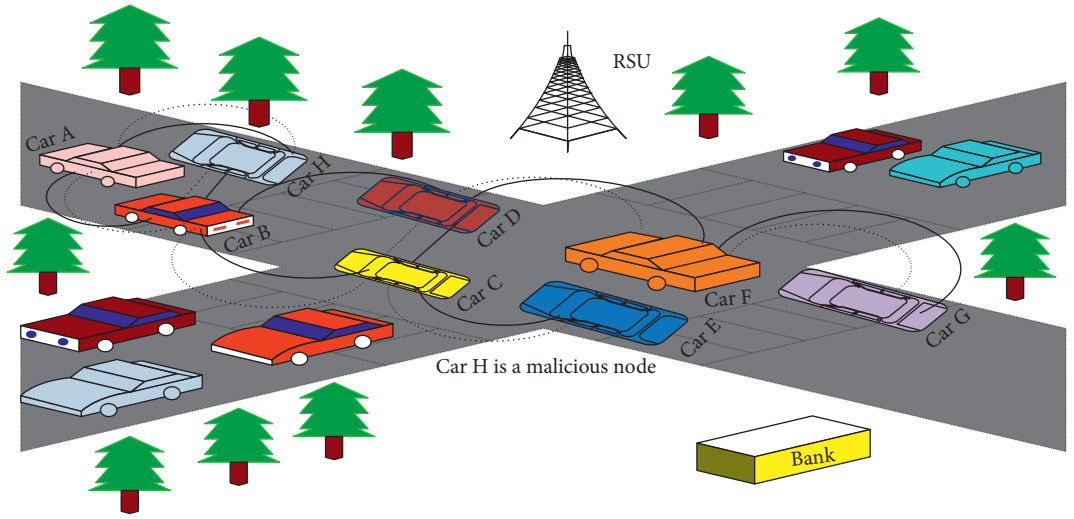


Figure 11: Gray hole assault.

intermediary node. This type of attack is a cryptography assault wherein the intermediary node will serve as an assailant that strives to decrypt the message through various decryption techniques [29, 52]. Figure 14 shows that Car L wants to send information to Car Q, while Car Q is far away. Thus, Car L sends the encrypted data packet to Car Q through Car N that is a malicious node which may attempt brute force assault and decrypt the message through a variety of decryption techniques.

3.4. Threats in Protocol Layer of VANETs. VANETs Routing Protocols (RP) consist of two groups, one is topology-based and the other is position-based routing. Every node is well

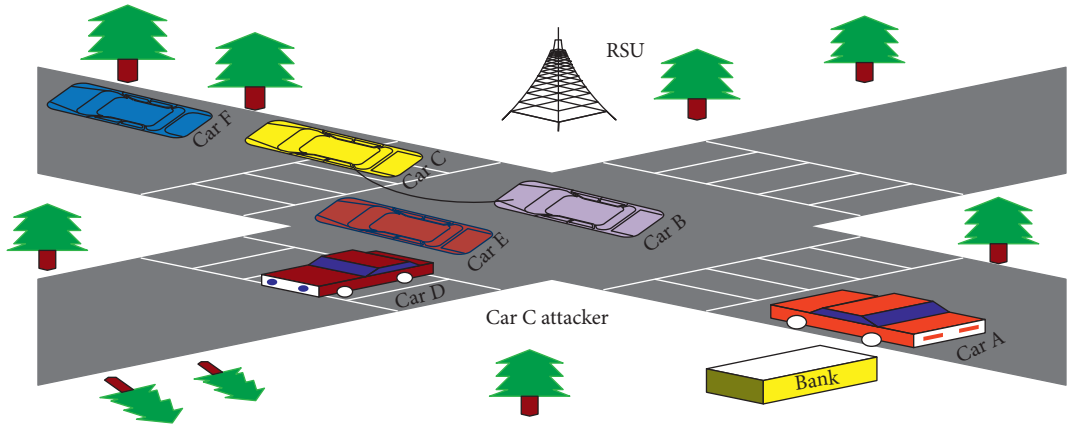


Figure 12: Masquerading assault.

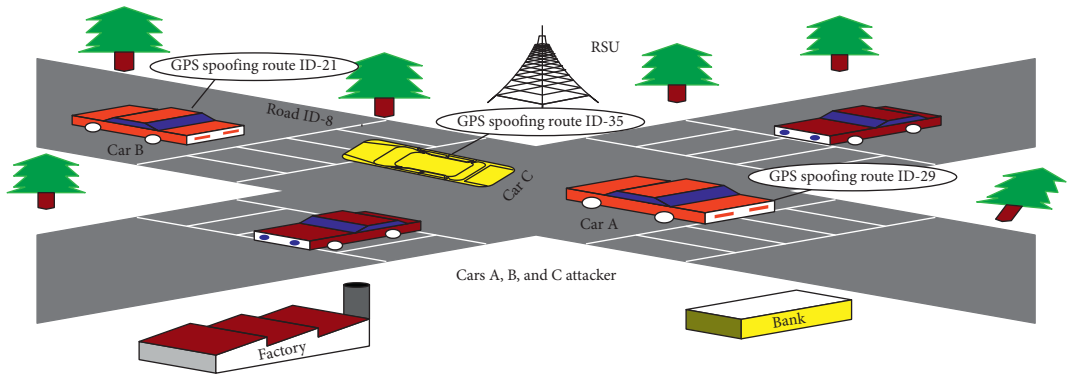


Figure 13: GPS spoofing assault.

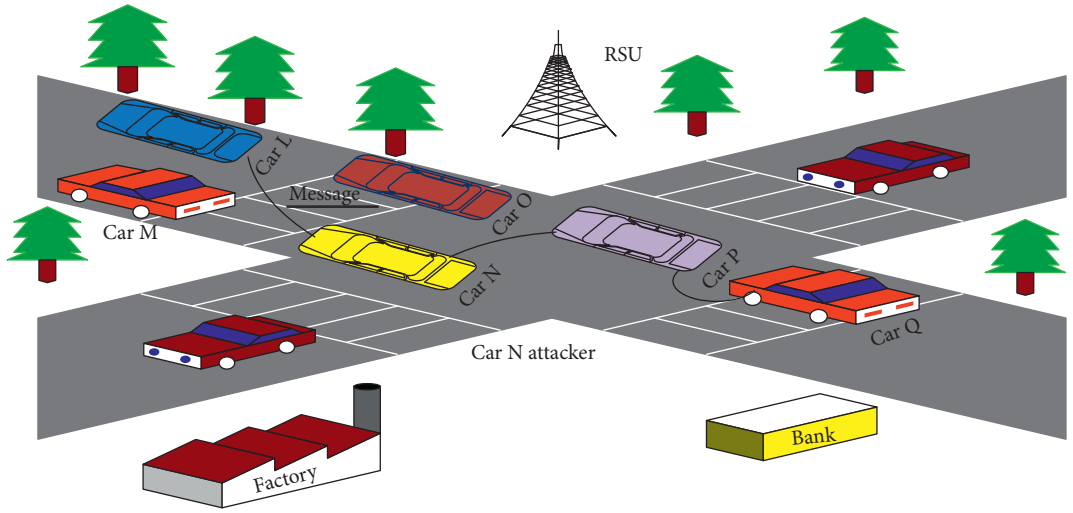


Figure 14: Brute force assault.

aware of the network layout in topology-based RP and sends messages using the accessible nodes and network connection information. One of the other side position-based RP nodes must be aware of the other node's location or position in which packet is being forwarded [53]. Figure 15 shows the two types of VANET routing protocols.

3.4.1. Topology-Based Protocol. The fundamental principle of the table-driven protocol is predetermining the route or path. It must gradually update the routing table every time the routing table is updated and share with neighboring node regularly [54]; therefore, while one node desires to communicate with another node, they already know about

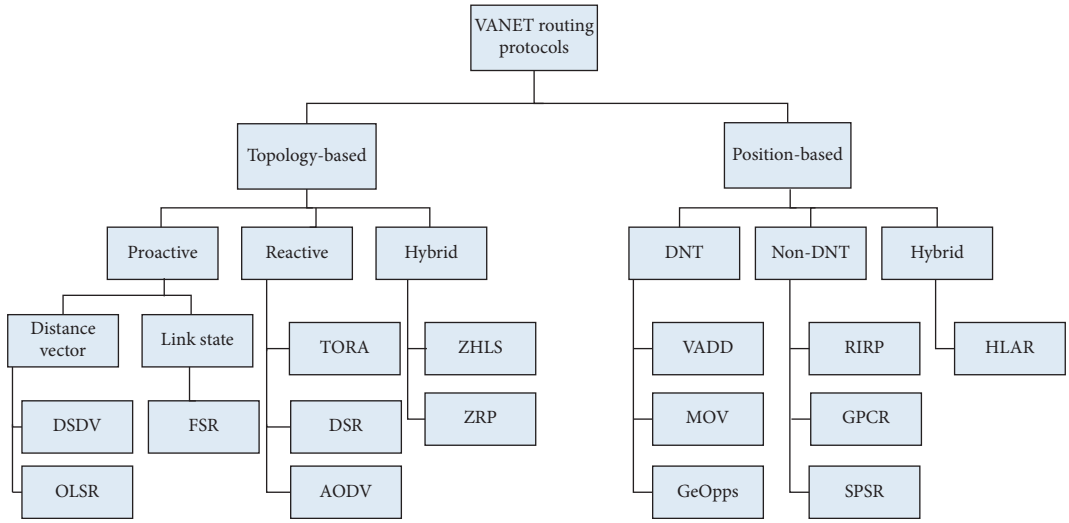


Figure 15: Routing protocols.

the path. One significant advantage of proactive protocols is the availability of path when the node wants to communicate on a network, but bandwidth decline is due to the generation of traffic caused by the swap of control packets [53, 55]. Proactive protocol examples are OLSR, DSDV, and GSR.

(i) Advantages:

Tracing the location of the route is not needed
Low latency when running in real time

(ii) Disadvantages:

Vacant routes consume an important session of the unoccupied bandwidth

3.4.2. Optimized Link State Routing. In MANETs, OLSR [56] is the table-driven routing protocol. OLSR can be regarded as the strength of a link-state algorithm for its benefits in relation to finding the path of any node whenever needed. Initially, using a particular node called multipoint relays (MPRs) [57], OLSR decreases the overhead from flooding of control traffic. In MPRs, select only those communication nodes that are the best path to provide from the source only to the destination, so MPRs help control traffic. Secondly, in OLSR requisite, just partial link states are flooded with an objective to present the shortest path routes [53, 58]. OLSR neighbor list table consists of up-to-date information which can be obtained from the neighboring nodes after exchanging its link-state information with those neighboring vehicle/nodes at regular intervals. As in link-state protocols, the routing messages created on a link are changed dynamically. This minimizes the number of control messages sent over the network which considerably [59] can deal with the blockage in traffic in VANETs, by forwarding and relaying the message to the nodes [58, 60].

3.4.3. Destination Sequenced Distance Vector. Bellman and ford have developed a centralized algorithm for assessing the shortest paths in weighted graphs. It was designed by Bertsekas and Gallager to execute in a distributed vogue called Distributed Bellman-Ford (DBF) algorithm [53, 61]. In DBF, all single nodes keep up the cost to arrive at each familiar destination. Hence, DBF comprises entries in the routing table. The routing table has no entry at the start, and all nodes start issuing a periodic broadcast message to its 1-hope neighborhood. The main drawback of this protocol is that it leads to count-to-infinity and looping problems. The loop may appear if the information regarding the assessment of shortest route becomes outdated. The primary purpose behind the origin of DSDV is to avoid the formation of loops. In DSDV the nodes converse with other network nodes. Each node has a routing table that refers to another network node that stores the necessary information concerning accessible destinations and the number of hopes to reach every node routing table. To maintain reliability in dynamically varying topologies, every vehicle/node exchanges its routing information with other neighbor vehicle/node at regular intervals or instantly while new information is updated in the routing table [54]. Every vehicle/node has its unique sequence number with each path as mentioned below:

(i) The target Internet protocol address
(ii) Number of hops requisite to arrive at the target location
(iii) The sequence number of the information received about that target location as initially marked by the target location

3.4.4. Global State Routing. Global state routing is a table-driven routing protocol; the link state algorithm is the basis of the global state routing protocol. It modifies and extends the connection state algorithm by limiting the message's middle vehicle/node renewal information. Each node in GSR holds a list of neighboring nodes, a topology table, and the next table of hope [59]. The neighbor list table consists of up-to-date information which can be obtained from the

neighboring nodes after exchanging its link-state information with those neighboring vehicle/nodes at regular intervals. As in link-state protocols, the routing messages created on a link are changed dynamically. This minimizes the number of control messages sent over the network considerably [59].

3.5. Reactive Protocols (On-Demand). The fundamental principle of reactive protocols is path allocation when the vehicle wants to communicate with another vehicle. Routing protocols have the key advantage of saving bandwidth in the reactive protocol when the node sends a message to the first path to be discovered. When a path is final from source to an intended destination, it is updated in the routing table and is then used for communication among source node to an intended destination node, and this path remains occupied with another node till the communication is completed [60, 62] (Reactive Protocols Example: AODV and DSR).

(i) Advantages:

To update the routing table, periodic flooding in the network is not required. Flooding is only done when required.
It saves the bandwidth.

(ii) Disadvantages:

For path discovery latency is high.
Too much flooding of the network disrupts the node's communication.

3.5.1. Ad Hoc On-Demand Distance Vector (AOVD). AOVD [28, 47, 61, 63], in MANETs, AODV protocol, is used for on-demand routing purposes with reactive routing. In the AODV protocol, routing table is maintained to store the next node routing information, i.e., for the target location nodes, and each routing table is used for a specific time period. If the path is demanded within a specific time, it becomes expired. Later, if a node wants to communicate, then again it finds a new route. In AODV, when the source node sends data, it checks the routing table and sends if the route is available. Otherwise, it needs to start the pathfinding process again to discover the finest route source to the target location for the purpose of transmitting packets through the broadcasting of route/path request (RREQ) message to its neighbor node. AODV was geared towards reducing the distribution of control traffic and stopping data traffic overhead, improving scalability and efficiency [16, 53, 58, 60, 64]. Figure 16 shows that in AODV the messages RREQ and RREP are used. In this figure, node S wants to communicate with node D, and all nodes are connected to their neighbor nodes and submit an RREQ message while every node sends REEQ message to the neighbor node. After receiving the RREQ message, every node sends back an RREP message. When all RREP messages are received, the source node chooses the best path and starts communication [57].

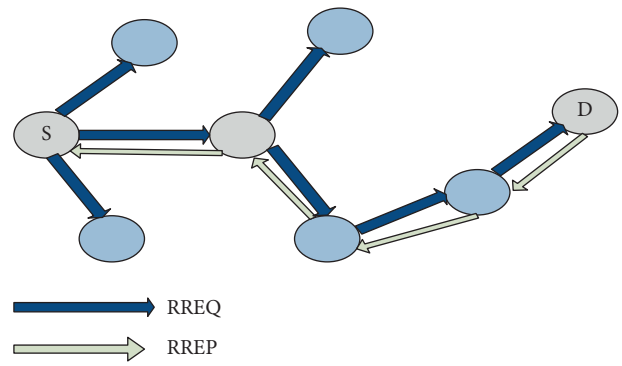


Figure 16: AODV RREQ and RREP message.

3.5.2. Dynamic Source Routing Protocol. DSR [65, 66] is a type of reactive routing protocol. If the vehicle desires to communicate with another vehicle in the network, it will search for a path and send packets to the intended destination. First, the vehicle searches a path after broadcasting a Route Request (RREQ), and this request passes through different nodes till the destination node where data need to be transferred. After they receive the path demand message, the intended destination broadcasts a Route Reply (RREP) packet back to the source vehicle with a unique ID. The dynamic source routing protocol stores the path information. If any unbroken connection or vacant path exists, then information is processed through path repairs. If there is any error on the path, the vehicle will send the Route Error message to the network [66]. DSR protocol is used in VANETs to maintain the network information and submit information about the traffic towards road-side unit [58, 66]. Table 1 shows the features of three routing protocols.

3.5.3. Security Issues for These Protocol Types. The AODV is a part of a reactive routing protocol. AODV's key benefit is that it is uncomplicated, takes less memory, and does not produce additional communication traffic along with the active connection. In AODV, the assailant might publicize a path with a slighter interval metric than the actual interval or publicize routing updates with a big sequence number after annulling all routing updates from supplementary nodes. An additional upgrade edition of AOVD proposed to solve these issues is secure AODV that presents more protected substantiation and truthfulness in AODV through the multihop link [67]. DSR protocol is another type of reactive protocol. The dissimilarity between them utilizes source routing sooner than relying on the routing table at every intermediary node. In DSR, another option is available; i.e., the data packets in this protocol can be forwarded on a hop-by-hop basis. It is feasible to vary the source route as planned in the attacker's route request or route reply packets in dynamic source routing. In DSR, removing a node from a list, changing the order, or adding a new node to a list are potential hazards [67]. In DSDV, significant security issues are scalability and also inappropriate DSDV for extremely dynamic VANETs.

Table 1: Contrast of AODV, DSDV, and OLSR features.

Protocol property	AODV	OLSR	DSDV
Reactive	Agreed	Not	Not
Route maintained in	Route table	Route table	Route table
Quality of service support	Not	Agreed	Not
Multicast routes	Not	Agreed	Not
Distributed	Agreed	Agreed	Agreed
Unidirectional link	Not	Agreed	Not
Support multicast	Agreed	Agreed	Not
Periodic broadcast	Agreed	Agreed	Agreed

3.6. Position-Based Protocol. The geographic location of the destination is determined in location-based routing. The positioning-based RP is generally proposed for the ad hoc network and does not use the network address to send data from the source to the intended target location. In VANETs, the transmission range is lower due to this frequent crash in the routing path. It is also due to gaps and crashes in the network. The problem of fading effect in urban highway environments, like tunnels and giant buildings, causes severe signal loss [68, 69]. Table 2 provides a summary of position-based protocol challenges and countermeasures.

Position-based routing is separated into three major groups detailed as follows:

(i) Nondelay tolerant

(ii) Delay tolerant

(iii) Hybrid

3.6.1. Nondelay Tolerant Network. The position-based first category is based mainly on greedy forwarding. Greedy perimeter stateless routing (GPSR) [70] protocol is used in greedy forwarding. GPSR uses only city scenarios because the dilemma is routing loops, an overlong path structure, and incorrect packet orders enhancing [69]. GPSR is proposed for MANETs; GPSR has a stumpy packet delivery ratio. Another protocol used for connectivity-aware routing is called A-STAR [68] for city buses for maintaining the path-based information. This algorithm might help to find the shortest route by giving connectivity among the vehicular nodes [69].

3.6.2. Delay Tolerant Network. Delay tolerant network [68] is also known as disruption tolerant network [1], delay-tolerant network, and store-carry-and-forward process-based network. Most of the current VANETs protocol had been proposed for immobile destinations. Vehicle-assisted data delivery (VADD) [70] is based on a carry-and-forward mechanism. A protocol connectivity-aware minimum delay geographic routing (CMGR) is similar to VADD. If we compare the CMGR and VADD, CMGR performs better as compared to VADD [69].

3.6.3. Hybrid Protocol. The hybrid protocol is a fusion of a Non-DTN and a disruption tolerant network. Geo-DTN + Nav for geographic transmission is a paradigm of hybrid protocol. In the hybrid protocol, we suppose that the target is standing still, being the reason for delay when one node switches to another. In GeoDTN + Nav [56], the message first switches to the perimeter node before moving to the disruption tolerant network for the enhanced broadcast of the message [69].

3.6.4. Issues for These Protocol Types. The crucial issue of GPRS is packet loss, and high delay could result in the loss of many hopes; as a result, perimeter mode forwarding may be expanded. STAR's reliability is drastically diminished by using a static street map to route packets of approximately possible radio obstacles, such as city buildings. GPCR uses no external static street map, so it is not easy to discover the intersection specifications. VADD is affected by the dynamic nature of the vehicular ad hoc network. It may cause a significant delay in delivery due to the traffic density [70].

3.7. Issues in the Application Layer of VANETs. The primary purpose of the protocol in the application layer is to minimize the end-to-end delay. However, sending emergency messages should arrive at the target vehicle by maintaining the deadline to supply service quality. In other applications, for instance, infotainment services delay is inevitable [71]. Vehicular information transfer protocol [72] is an application layer communication protocol to assist disseminated and ad hoc services infrastructure in VANETs. Two primary attacks on the application layer are malevolent code assault and repudiation assault. In malicious code attacks, malicious vehicles that want to attack networks send malicious codes like a virus, Trojan horse. These types of attacks damage the vehicle application and affect their services. In the repudiation attack [32], for instance, an application runs on a network that is used to control, track, and log user action, hence encouraging malevolent manipulation or spoofing of the recognition of new actions [71].

4. Solutions in VANETs

This section provides a brief review of the works furnished in the domain of VANETs security solutions. Table 3 provides a summary of challenges and countermeasures in VANETs.

4.1. Authenticated Routing for Ad Hoc Networks. The ARAN [73] routing protocol is based on AODV. In ARAN, a third party called certificate authority (CA) is responsible for sending a signed certificate to the nodes, upon receiving a certification request to CA. Asymmetric encryption techniques are used to verify the authenticity of secure path detection, and time tags are used to clear the path [75].

ARAN essentially has five steps:

(i) Certification

(ii) Authentic path finding

(iii) Authentic path setup

(iv) Path maintenance

(v) Key revocation

Table 2: Summary of position-based protocols challenges and countermeasures.

Challenges	Environment (traffic)	Countermeasures
Local optimal and link break	City traffic environment (no use of static external map)	GPSR protocol [70]
Maintain path base information	Static street map	A-STAR protocol [68]
Predictable vehicle mobility	City traffic environment	VADD and CMGR protocols [69, 70]

Table 3: Summary of challenges and countermeasures in VANETs.

Challenges	Techniques/technology	Countermeasures
Replay assault Impersonation assault Eavesdropping assault	Asymmetric encryption techniques are used to verify the authenticity of secure path detection, and time tags are used to clear the path.	Authenticated routing for ad hoc network protocol [73]
DoS assault Routing assault Impersonation assault	It uses the authentication process through one-way hash function.	Secure and efficient ad hoc distance vector protocol [74]
DoS assault Routing assault Replay assault	This protocol uses symmetric cryptographic operations. The one-way hash and MAC functions are used for substantiation and are transmitted via a shared key between nodes.	Ariadne [75]
Routing assault Impersonation assault Bogus information	It uses digital signature and hash function.	SAODV [75]
Routing assault Impersonation assault Bogus information	It uses digital signature and hash function.	A-SAODV [75]
Session hijacking	Cookies are allocated for each session for session management.	One time cookie [75]
Sybil assault	Identifying a malicious node is achieved by discovery of two or supplementary nodes with similar trajectories motion.	Robust method for Sybil assault detection [76]
Impersonation assault	It uses registration ID technique.	Holistic protocol [75]

In the ARAN path, the authentication process is done in every step by adding each middle node's sign and certificate, so this protocol solves the impersonation problem.

4.2. Secure and Efficient Ad Hoc Distance Vector Protocol. Working over DSDV, the secure and efficient ad hoc distance vector protocol (SEAD) [74] uses the authentication process hash function. SEAD uses destination sequence number to ensure path freshness, which assists in avoiding the wrong path. To ensure path authenticity, the SEAD uses hashing on each intermediate node [75].

4.3. Ariadne. Working on DSR, this protocol uses symmetric cryptographic operations. The one-way hash and MAC functions are used for substantiation and are transmitted via a shared key between nodes. The TESLA uses Ariadne-based authentication for data transmission. The TESLA time interval is used in the route discovery and authentication process [75].

4.4. SAODV. This protocol proposed the integration of security measures into the AODV protocol. All routing correspondence is signed digitally to assure legitimacy, and hash functions are used to guard hop count. The route response cannot be sent in this intermediate node method, even though they know the new path. This problem can be solved by double signature; in addition, it raises the system complexity [75].

4.5. A-SAODV. A-SAODV is an extended version of SAODV, which has an experimental adaptive response decision attribute. Depending on the length of the queue and the threshold conditions, each middle node may come to a decision, whether to send a response to the source node or not [75].

4.6. One-Time Cookie. Usually, cookies are allocated for each session for session management. However, this protocol gives OTC the concept to protect the system from session abduction and SID stealing. OTC produces a token for every request, and these tokens are linked to request using HMAC to avoid the token from being reused [75].

4.7. Elliptic Curve Digital Signature Algorithm. ECDSA [77] algorithm utilizes a digital signature. Additionally, ECDSA ensures the genuineness and protection of the digital

signatures through hash and related symmetric key operations. It can be initiated once both the sender and the receiver agree upon the parameters for elliptical curve domain parameters [75].

4.8. Robust Method for Sybil Attack Detection. RobSAD [76] approach's core principle is that drivers cannot have the same movement pattern for two different vehicles, as every human being drives along with their comfort. Identifying a malicious node is achieved by the discovery of two or supplementary nodes with similar trajectories motion [76].

4.9. Holistic Protocol. This protocol describes the method of authentication by registering the vehicle/node by RSU. The vehicles send Hello message to the RSU during the vehicle registration process; RSU then prepares and sends the Registration ID (consisting of the license number and registration number of vehicle) to the node. Additionally, the verification is complete through a RSU certificate. If the vehicle is genuine, only information will be shared; otherwise, it will be blocked [75].

4.10. Challenges in the Physical Layer of VANETs. Due to the high speed, the signals of VANETs entities undergo multipath fading and Doppler frequency shifts. Hence, due to the effects of the multipath fading and frequency shifts, the need of physical layer communication arises. For testing the application, V2V uses radio and infrared (IR) waves to communicate. The V2V communication occurs through excessive frequencies like micro- and millimetre waves. The waves that belong to the infrared and millimetre category use the line of sight communication [71, 78].

The DSRC physical layer includes the 802.11p OFDM, which operates within 5.9 GHz band (5.885–5.9.5) range with a maximum of 10 MHz channel [78]. The underlying data rate is approximately 3 Mbps, and the default data rate is 6 Mbps. The physical layer in VANETs is a thoroughly researched area. From transmission control to using multiple (or individual) antennas and from evaluation to channel-to-channel selection, there are numerous aspects of the physical layer which contribute to network scalability. Owing to the spread of delays and mobility on several roads, the multipath environment makes communication extremely challenging. Delay-spread frequency selective fading and mobility cause time-selective fading. The need of the line of sight leads to a significant delay owing to dispersal, and Doppler spreads [79]. The challenges to the physical layer in VANETs consist of the following.

4.10.1. Dual and Single Radio. The coincidence among single and double radio is still vague. Although dual-radio has different clear benefits, inserting a second radio into the survival of single radios does not boost protection contact efficiency under the default scheme [79].

4.10.2. Model for Propagation. Vehicular ad hoc networks work in three types of environments: countryside, city, and highway. The free-space model used for the highway is not rigorously exact as the signal passes through the adjacent reflections. The city free-space model can be effected by shadowing and multipath fading. In a rural environment, some other factor, like trees and hills, can cause lots of reflection [79].

4.10.3. Selection of the Channel. An analytical and simulation study is required at the physical layer for the channel selection. A game-theoretic approach can be used for selecting the best channel and data rate [79].

4.10.4. Channel Estimation. We require advanced channel estimation techniques in VANETs to acquire a correct channel state information (CSI) [79].

4.10.5. Variety of Techniques. Fading and interfering effects can be minimized using a range of techniques [79].

4.11. Algorithms in the Protocol Layer of VANETs. The reliance on remote correspondence, control, and handling innovation renders IoV dynamically weak against potential ambushes, such as remote interruption, control, and direction [80]. For itself, compelling validation courses of action envisioning unapproved visitors must be directed to adapt to these issues. Thus, this work focuses on the security and protection by structuring up twofold verification conspiring for Internet of Vehicles as demonstrated by its different situations. In any case, the OBU self-makes an unclear personality and provisional encryption key to open a validation session. Second, the trust master's legitimacy of the node's actual and baffling personality can be confirmed (TA). Table 4 provides a summary of algorithms in protocol layers of VANETs challenges and countermeasures.

Zeng et al. [81] proposed a new route for city VANETs formed by connectivity analysis based on geographical position to conquer the general mistakes of VANETs route in the city area. In combination with a digital city map, LCGL manages the geographical position information about nodes and connections. LCGL selects the shortest connected route to forward the data packet to the route and link length.

As per Sun et al. [82], several open communication protocols overlook the nearness of structures or difficulties accessible amid viable use, mainly in urban regions. These deterrents can cause signal fading or even square direct communication. Numerous vehicles are often left on the road side. As a result of their location, these left vehicles can be utilized as transfers to successfully lessen the shadowing impact of deterrents and even tackle communication issues. In this work, the author exhibited left-vehicle right-hand off-routing communication in vehicle ad hoc networks. The author of [82] proposed a practical left vehicle associate hand-off routing calculation made out of four sections: an occasional Hello packet trade instrument, competitor transfer list update, communication connect quality

Table 4: Summary of algorithms in protocol layers of VANETs challenges and countermeasures.

Paper	Challenges	Countermeasures
Zeng et al. [81]	Connectivity analysis in city based on geographical position	Link connectivity analysis on geographic location (LCGL) routing scheme
Sun et al. [82]	Road side vehicle communication issue	Practical left vehicle associate had off routing
Rahman and Tepe [83]	Channel access conflicts confirming improved channel usage in cross layer V2V and V2I communication\enleadertwodots	Multilevel algorithm removing these issues
Kumar and Mann [84]	Multiple malicious node detection in the network and avoiding DOS assault	Packet detection algorithm
Malla and Sahu [85]	Various existing solutions using cryptographic techniques are time and resources consuming	The proposed solution betters the security in VANETs without using cryptographic techniques
Waraich and Batra [86]	DOS assault recognition	Quick response table and recognition of the DOS attack
Jeffane and Ibrahimi [87]	DOS assault on the physical and MAC layers in IEE standard 802.11p	Packet delivery ratio (PDR) metric to detect the DOS attack
RoselinMary et al. [88]	Detecting the DOS assault before the verification time	Attacked packet detection algorithm
Singh and Sharma [89]	DOS attack is the main challenge to network availability	Proposing an enhanced attacked packet detection algorithm
Quyoom et al. [31]	Detecting the DOS assault	Proposed MIPDA
Issac and Mary [90]	Protection against DOS assault to mitigate packet loss	Updated prediction-based authentication method (PBA)
Sohail et al. [91]	Security challenges in VIoT, such as efficient trust assessment, certified user nonfunctioning and secure information diffusion	Proposing a new scheme, trust enhanced on-demand routing (TER)

evaluation, and hopeful hand-off rundown selection. Simulation results uncover evident advantages for lists, such as the nature of communication, achievement rate, and time delay.

Ad hoc vehicular networks have twisted into an increasing innovation that can gratify the interest of advancing associated vehicles and developing prerequisites for the Canny Transportation Framework (ITS). Authentications are utilized to confirm vehicular correspondence though the declarations of vehicles should be disavowed if every vehicle is found to get out of hand hubs. In VANETs, authentication disavowal Certificate Revocation Lists (CRL) must be instantly conveyed to every single vehicular hub to avoid redundant correspondence with the noxious hubs. Be that as it may, because of developing several testaments, the measure of CRL constantly increases, and, subsequently, it ends up hard overseeing and conveying the CRL in vehicular networks. The author presents a compelling and adaptable plan to convey a declaration denial list in the various leveled engineering of VANETs [92].

Rahman and Tepe [83] stated that the DSRC/WAVE system is standardized to broadcast critical security information with IEEE 802.11p as MAC protocol. Studies show that IEEE 802.11p fights the adverse effects of asymmetric radio communications and mobility problems in V2V and V2I communication. The author provides a well-organized and consistent cross-layer algorithm for problems with V2V and V2I communication. The analysis shows that the multilevel algorithm's proposal removes channel access conflicts and confirms improved channel usage. The solution can be the dissemination of up to three jumps without routing protocol. That is chiefly significant for security and emergency critical message of area vehicle network.

Kumar and Mann [84] considered the safety of VANETs. As per Kumar et al., the security of the vehicles or nodes can be enlarged if the network accessibility is increased. If the denial of service of attack happens on the network, the availability of the network decreases. The authors proposed an algorithm that was proficient at sensing the numerous malicious nodes or vehicles that transfer the unrelated packet to squeeze the network and ultimately stop the network from transmitting the safety information messages. The proposed algorithm simulated on NS-2 and the quantitative values of packet delivery ratio, packet loss ratio, and network throughput demonstrates that by detecting the denial of service attack in a good time, the proposed algorithm improves the network security.

Vehicular ad hoc network aims to improve transportation efficiency and safety. VANETs have open nature of wireless medium, so the number of chances of various attacks in this work increases. The authors proposed a solution for DOS attack which uses the redundancy removal mechanism consisting of rate decreasing algorithms and state transition mechanism as its components.

The protocol of Malla and Sahu [85] uses various existing solutions (channel switching, frequency hopping, multiradio transceivers, and communication technology). The proposed solution betters the security in VANETs without using cryptographic techniques.

Due to high mobility in VANETs, secure routing is a big issue [86]. The topology nature of VANETs is dynamic; paths are regularly updated, and sometimes the communication link breaks due to hurdles such as buildings, bridges, and tunnels. It is challenging to determine the reason for packet drop because persistent connection breaks can cause packet drop, resulting in deterioration of

network performance in vehicular ad hoc networks. This also happens due to the existence of security threats. VANETs are subclass of MANETs and exist in the same attack. Researchers have already developed different security mechanisms for safe routing in MANETs, but these solutions are not compatible with VANETs because of specific attributes. A vehicle can communicate with other vehicles (V2V) as well as communicating to infrastructure (V2I). Waraich and Batra [86] proposed a solution to avoid the DOS attack to ensure routing for both forms of communication. They use the quick response table and recognize the DOS attack.

VANETs are a subgroup of MANETs. It is developed to provide communication between vehicles and fixed equipment (RSU) to give each other's range. VANETs are very sensitive to safety issues. Jeffane and Ibrahimi [87] proposed a new mechanism that focuses on the denial of service attack on the physical and MAC layers in IEE standard 802.11p. This solution uses the packet delivery ratio (PDR) metric to detect the DOS attack.

Security for VANETs is vital because their very presence relates to critical circumstances that are life-threatening [88]. VANETs are a subgroup of MANETs. All nodes or vehicles are equipped with an On-Board Unit (OBU), enabling data from one node to another in the network to be sent and received. In vehicular ad hoc network communication interface provided by the on-road infrastructure, to detect the denial of service attack before verification time, Roselin Mary et al. [88] proposed a new algorithm (attacked packet detection algorithm).

Important information shared for vehicle protection is the major issue. The node is self-organized, highly mobile, and of free movement in a vehicular ad hoc network, so any node may communicate with any other node that may (or not) be trustworthy. This is the area of concern inside the VANETs security horizon. The road-side unit is responsible for every node at all times and provides the communication of secure information. The vehicles and the RSU are prone to several security attacks like selfish driver attacks, masquerading attacks, Sybil attacks, and alteration attacks. DOS attack is the main challenge to network availability. Singh and Sharma [89] proposed an enhanced attacked packet detection algorithm, which prohibits network performance deterioration even under this attack. EAPDA checks the nodes, detects malicious nodes, and better gets the throughput with minimized delay, thus improving security.

As per Quyoom et al. [31], the security of VANETs plays a vital role in sustaining essential life. A sensitive, life-related information network must be open at all times for secure communication. Several types of attacks and threads possible in VANET were subject to the network accessibility problem. These attacks include Sybil attacks, misbehaving attacks, incorrect vehicle position information, and selfish driver and jamming attacks. Among these attacks, a significant threat to the information economy is the denial of service attacks. To analyze and detect the DOS attack, the authors proposed a Malicious and Irrelevant Packet Detection Algorithm (MIPDA).

Issac and Mary [90] used the updated prediction-based authentication method (PBA) to protect against VANETs DOS attack to mitigate packet loss caused by vehicle mobility. The primary aim is to reduce the delay in validating emergency vehicles such as ambulances and fire services. The architecture of the PBA is such that the beacons cannot be predicted by the sender vehicles. This process has been shown to be secure as a result.

The IoT plays an essential role in connecting the network with the world and new technologies. However, VANETs being an important segment of IoT have faced various challenges due to the high mobility and dynamic nature of the network. IoT focuses in future to allow internetworking to disseminate information. Previous security solutions to vehicular Internet of Things (VIoT) focus more on privacy protection and security-related challenges using PKI. Sohail et al. [91] proposed a new scheme, trust enhanced on-demand routing (TER). This scheme overcomes the security challenges in VIoT, such as efficient trust assessment, certified user nonfunctioning, and secure information diffusion.

4.12. Solutions in the Application Layer of VANETs. The principal aim for the application layer protocols is to decrease the end-to-end delay caused by sending emergency messages. In other applications, for instance, infotainment services delay is predictable. Vehicular Information Transfer Protocol (VITP) is an application layer communication protocol to support distributed and ad hoc service infrastructure in VANET [71]. Two possible primary assaults in application layer are malevolent code attack and the repudiation attack. In malevolent code assault, malevolent vehicles send malevolent code or programs, for instance, viruses, Trojan horses. These malicious codes damage the vehicle application and affect their services. A repudiation attack, in which attackers control the whole network with the help of the various applications, gets all information quickly and manipulates the message. The application layer is capable of detecting DoS attacks than other layers [71].

Two schemes were proposed in [67]; the first scheme is an application-aware control scheme in which all accessible applications should be periodically registered and updated and forwarded to all other VANETs nodes. The second scheme includes the unified routing scheme that will route a packet of precise applications according to demand and safety requirements.

5. Conclusion

Consisting of mobile information and communication infrastructure, the VANETs play an important role in road safety and travel comfort. However, as technology is growing and VANETs are getting more popular, security vulnerabilities are increasing rapidly, which ultimately restricts the widespread usage of the VANETs. In this article, the security vulnerabilities of VANETs are surveyed. The article also provides layer-specific attack classification in the VANETS protocol stack. Besides, we also provided a discussion on several countermeasures.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Conceptualization was done by J. M. and Y. Y.; investigation was carried out by Y. Y. and M. N. M. B.; original draft was prepared by J. M. and Y. Y.; review and editing were done by Y. Y. and J. N.; supervision was provided by Z. D. and Q. W.; funding acquisition was made by Z. D.\enleadertwodots. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This work was supported by Key Research Item for the Industry of Shaanxi Province under grant no. 2018GY-136.

References

[1] R. Geng, X. Wang, and J. Liu, "A software defined networking-oriented security scheme for vehicle networks," *IEEE Access*, vol. 6, pp. 58195–58203, 2018.

[2] N. S. Samaras, "Using basic manet routing algorithms for data dissemination in vehicular ad hoc networks (VANETs)," in *Proceedings of the 2016 24th Telecommunications Forum (TELFOR)*, pp. 1–4, IEEE, Belgrade, Serbia, November 2016.

[3] J. Wantoro and I. W. Mustika, "M-aodv+: an extension of aodv+ routing protocol for supporting vehicle-to-vehicle communication in vehicular ad hoc networks," in *Proceedings of the 2014 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, pp. 39–44, IEEE, Jakarta, Indonesia, November 2014.

[4] L. Feng, Y. Xiu-Ping, and W. Jie, "Security transmission routing protocol for mimo-vanet," in *Proceedings of the 2014 International Conference on Cloud Computing and Internet of Things*, pp. 152–156, IEEE, Changchun, China, December 2014.

[5] C. Pathak, A. Shrivastava, and A. Jain, "Ad Hoc on demand distance vector routing protocol using dijkastra's algorithm (aodv-d) for high throughput in vanet (vehicular Ad Hoc network)," in *Proceedings of the 2016 11th International Conference on Industrial and Information Systems (ICIIS)*, pp. 355–359, IEEE, Roorkee, India, December 2016.

[6] I. U. Rasool, Y. B. Zikria, and S. W. Kim, "A review of wireless access vehicular environment multichannel operational medium access control protocols: quality-of-service analysis and other related issues," *International Journal of Distributed Sensor Networks*, vol. 13, no. 5, 2017.

[7] M. Arif, G. Wang, M. Z. A. Bhuiyan, T. Wang, and J. Chen, "A survey on security attacks in VANETs: communication, applications and challenges," *Vehicular Communications*, vol. 19, Article ID 100179, 2019.

[8] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETs): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[9] Z. Afzal and M. Kumar, "Security of vehicular ad-hoc networks (vanet): a survey," *Journal of Physics: Conference Series*, vol. 1427, no. 1, Article ID 012015, 2020.

[10] M. S. Sheikh, J. Liang, and W. Wang, "Security and privacy in vehicular ad hoc network and vehicle cloud computing: a survey," *Wireless Communications and Mobile Computing*, vol. 202025 pages, Article ID 5129620, 2020.

[11] A. Awang, K. Husain, N. Kamel, and S. Aissa, "Routing in vehicular ad-hoc networks: a Survey on single- and cross-layer design techniques, and perspectives," *IEEE Access*, vol. 5, pp. 9497–9517, 2017.

[12] S. A. Chaudhry, "Correcting "palk: password-based anonymous lightweight key agreement framework for smart grid"" *International Journal of Electrical Power & Energy Systems*, vol. 125, Article ID 106529, 2021.

[13] X. Li, Y. Han, J. Gao, and J. Niu, "Secure hierarchical authentication protocol in vanet," *IET Information Security*, vol. 14, no. 1, pp. 99–110, 2019.

[14] A. Sari, O. Onursal, M. Akkaya et al., "Review of the security issues in vehicular ad hoc networks (vanet)," *International Journal of Communications, Network and System Sciences*, vol. 8, no. 13, pp. 552–566, 2015.

[15] Z. A. Abdulkader, A. Abdullah, M. Taufik Abdullah, and Z. Ahmad Zukarnain, "Vehicular ad hoc networks and security issues: survey," *Modern Applied Science*, vol. 11, no. 5, Article ID 30, 2017.

[16] R. C. Poonia, D. Bhargava, and B. S. Kumar, "Cdra: cluster-based dynamic routing approach as a development of the aodv in vehicular ad-hoc networks," in *Proceedings of the 2015 International Conference on Signal Processing and Communication Engineering Systems*, pp. 397–401, IEEE, Guntur, India, January 2015.

[17] P. Agarwal, "Technical review on different applications, challenges and security in vanet," *Journal of Multimedia Technology & Recent Advancements*, vol. 4, no. 3, pp. 21–30, 2017.

[18] V. K. Tripathi and S. Venkaeswari, "Secure communication with privacy preservation in vanet-using multilingual translation," in *Proceedings of the 2015 Global Conference on Communication Technologies (GCCT)*, pp. 125–127, IEEE, Thuckalay, India, April 2015.

[19] A. Kumar and M. Bansal, "A review on vanet security attacks and their countermeasure," in *Proceedings of the 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*, pp. 580–585, IEEE, Solan, India, September 2017.

[20] P. Caballero-Gil and X. Wang, "Security issues in vehicular ad hoc networks," *Mobile Ad Hoc networks: Applications*, pp. 67–88, 2011.

[21] A. Irshad, M. Usman, S. A. Chaudhry, H. Naqvi, and M. Shafiq, "A provably secure and efficient authenticated key agreement scheme for energy internet based vehicle-to-grid technology framework," *IEEE Transactions on Industry Applications*, vol. 99, p. 1, 2020.

[22] S. Hussain and S. A. Chaudhry, ""Comments on "biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment"" *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 936–10 940, 2019.

[23] N. K. Chaubey, "Security analysis of vehicular ad hoc networks (VANETs): a comprehensive study," *International Journal of Security and Its Applications*, vol. 10, no. 5, pp. 261–274, 2016.

[24] S. M. Faisal and T. Zaidi, "Timestamp based detection of sybil attack in vanet," *IJ Network Security*, vol. 22, no. 3, pp. 397–408, 2020.

[25] M. B. Mansour, C. Salama, H. K. Mohamed, and S. A. Hammad, "Vanet security and privacy-an overview," *International Journal of Network Security & Its Applications*, vol. 10, no. 2, pp. 13–34, 2018.

[26] R. G. Engoulou, M. Bellaïche, S. Pierre, and A. Quintero, "Vanet security surveys," *Computer Communications*, vol. 44, pp. 1–13, 2014.

[27] I. A. Sumra, I. Ahmad, H. Hasbullah et al., "Classes of attacks in vanet," in *Proceedings of the 2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC)*, pp. 1–5, IEEE, Riyadh, Saudi Arabia, April 2011.

[28] A. Suman and C. Kumar, "A behavioral study of sybil attack on vehicular network," in *Proceedings of the 2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, pp. 56–60, IEEE, Dhanbad, India, March 2016.

[29] A. N. Upadhyaya and J. Shah, "Attacks on vanet security," *International Journal of Computer Engineering and Software Technology*, vol. 9, no. 1, pp. 8–19, 2018.

[30] K. Mahmood, J. Arshad, S. A. Chaudhry, and S. Kumari, "An enhanced anonymous identity-based key agreement protocol for smart grid advanced metering infrastructure," *International Journal of Communication Systems*, vol. 32, no. 16, Article ID e4137, 2019.

[31] A. Quyoom, R. Ali, D. N. Gouttam, and H. Sharma, "A novel mechanism of detection of denial of service attack (dos) in vanet using malicious and irrelevant packet detection algorithm (mipda)," in *Proceedings of the International Conference on Computing, Communication & Automation*, pp. 414–419, IEEE, Noida, India, May 2015.

[32] R. Shringar Raw, M. Kumar, and N. Singh, "Security challenges, issues and their solutions for vanet," *International Journal of Network Security & Its Applications*, vol. 5, no. 5, pp. 95–105, 2013.

[33] H. Hasbullah, I. A. Soomro, and J. Manan, "Denial of service (dos) attack and its possible solutions in vanet," *International Journal of Electronics and Communication Engineering*, vol. 4, no. 5, pp. 813–817, 2010.

[34] D. Rampaul, R. K. Patial, and D. Kumar, "Detection of dos attack in VANETs," *Indian Journal of Science and Technology*, vol. 9, no. 47, pp. 1–6, 2016.

[35] A. Sahi, D. Lai, Y. Li, and M. Diykh, "An efficient DDOS TCP flood attack detection and prevention system in a cloud environment," *IEEE Access*, vol. 5, pp. 6036–6048, 2017.

[36] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman, "Securing smart city surveillance: a lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, vol. 8, pp. 43711–743724, 2020.

[37] S. A. Chaudhry, T. Shon, F. Al-Turjman, and M. H. Alsharif, "Correcting design flaws: an improved and cloud assisted key agreement scheme in cyber physical systems," *Computer Communications*, vol. 153, pp. 527–537, 2020.

[38] D. Kushwaha, P. K. Shukla, and R. Baraskar, "A survey on sybil attack in vehicular Ad Hoc network," *International Journal of Computer Applications*, vol. 98, no. 15, 2014.

[39] M. Rahbari and M. A. J. Jamali, "Efficient detection of sybil attack based on cryptography in vanet," 2011, https://arxiv.org/abs/1112.2257.

[40] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IOV," *Ad Hoc Networks*, vol. 61, pp. 33–50, 2017.

[41] T. Zaidi and S. Faisal, "An overview: various attacks in vanet," in *Proceedings of the 2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pp. 1–6, IEEE, Greater Noida, India, December 2018.

[42] V. Bibhu, K. Roshan, K. B. Singh, and D. K. Singh, "Performance analysis of black hole attack in vanet," *International Journal of Computer Network and Information Security*, vol. 4, no. 11, pp. 47–54, 2012.

[43] I. Dhyani, N. Goel, G. Sharma, and B. Mallick, "A reliable tactic for detecting black hole attack in vehicular ad hoc networks," in *Advances in Computer and Computational Sciences*, pp. 333–343, Springer, Berlin, Germany, 2017.

[44] K. C. Purohit, S. C. Dimri, and S. Jasola, "Mitigation and performance analysis of routing protocols under black-hole attack in vehicular Ad Hoc network (vanet)," *Wireless Personal Communications*, vol. 97, no. 4, pp. 5099–5114, 2017.

[45] H. P. Singh, V. P. Singh, and R. Singh, "Cooperative black-hole/grayhole attack detection and prevention in mobile ad hoc network: a review," *International Journal of Computer Applications*, vol. 64, no. 3, pp. 16–22, 2013.

[46] M. A. H. Al Junaid, A. Syed, M. N. M. Warip, K. N. F. K. Azir, and N. H. Romli, "Classification of security attacks in vanet: a review of requirements and perspectives," in *Proceedings of the MATEC Web of Conferences*, EDP Sciences, Article ID 06038, 2018.

[47] S. Lachdhaf, M. Mazouzi, and M. Abid, "Secured aodv routing protocol for the detection and prevention of black hole attack in vanet," *Advanced Computing: International Journal*, vol. 9, no. 1, 2018.

[48] J. Tobin, C. Thorpe, and L. Murphy, "An approach to mitigate black hole attacks on vehicular wireless networks," in *Proceedings of the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pp. 1–7, IEEE, Sydney, Australia, June 2017.

[49] J. Singh and N. Sharma, "Wormhole attack detection by using intrusion detection system in vanet," *International Journal of Computer Networks and Wireless Communications (IJCNWC), ISSN*, pp. 2250–3501, 2012.

[50] S. Verma, B. Mallick, and P. Verma, "Impact of gray hole attack in vanet," in *Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, pp. 127–130, IEEE, Dehradun, India, September 2015.

[51] N. Phull and P. Singh, "A review on security issues in VANETs," in *Proceedings of the 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1084–1088, IEEE, New Delhi, India, March 2019.

[52] N. Couture and K. B. Kent, "The effectiveness of brute force attacks on rc4," in *Proceedings of the Second Annual Conference on Communication Networks and Services Research, 2004*, pp. 333–336, IEEE, Fredericton, Canada, May 2004.

[53] B. Hamid and E.-N. El Mokhtar, "Performance analysis of the vehicular ad hoc networks (vanet) routing protocols aodv, dsdv and olsr," in *Proceedings of the 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA)*, pp. 1–6, IEEE, Marrakech, Morocco, December 2015.

[54] I. Mouhib, M. Smail, M. D. El Ouadghiri, and H. Naanani, "Network as a service for smart vehicles: a new comparative study of the optimized protocol q-aodv and gpsr protocol,," in *Proceedings of the 2016 International Conference on Engineering & MIS (ICEMIS)*, pp. 1–5, IEEE, Agadir, Morocco, September 2016.

[55] A. Datta, "Modified ant-aodv-vanet routing protocol for vehicular adhoc network," in *Proceedings of the 2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)*, pp. 1–6, IEEE, Kolkata, India, April 2017.

[56] K. N. Qureshi and H. Abdullah, "Topology based routing protocols for vanet and their comparison with manet," *Journal of Theoretical and Applied Information Technology*, vol. 58, no. 3, pp. 707–715, 2013.

[57] L. Rivoirard, M. Wahl, P. Sondi, M. Berbineau, and D. Gruyer, "Performance evaluation of aodv, dsr, grp and olsr for vanet with real-world trajectories," in *Proceedings of the 2017 15th International Conference on ITS Telecommunications (ITST)*, pp. 1–7, IEEE, Warsaw, Poland, May 2017.

[58] A. Chekima, F. Wong, J. A. Dargham et al., "A study on vehicular ad hoc networks," in *Proceedings of the 2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, pp. 422–426, IEEE, Kota Kinabalu, Malaysia, December 2015.

[59] T. P. Venkatesan, P. Rajakumar, and A. Pitchaikkannu, "Overview of proactive routing protocols in manet," in *Proceedings of the 2014 Fourth International Conference on Communication Systems and Network Technologies*, pp. 173–177, IEEE, Washington, DC, USA, April 2014.

[60] A. Nayyar, "Flying adhoc network (fanets): simulation based performance comparison of routing protocols: aodv, dsdv, dsr, olsr, aomdv and hwmp," in *Proceedings of the 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, pp. 1–9, IEEE, Durban, South Africa, August 2018.

[61] S. Tabar, L. Najjar, and M. Gholamalitabar, "Quality of service in the network layer of vehicular ad hoc networks," in *Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, CA, USA, October 2016.

[62] B. Paul and M. Abu Naser Bikas, "Vanet routing protocols: pros and cons," *International Journal of Computer Applications*, vol. 20, no. 3, pp. 28–34, 2011.

[63] M. N. Amara korba Abdelaziz and G. Salim, "Analysis of security attacks in aodv," in *Proceedings of the 2014 International Conference on Multimedia Computing and Systems (ICMCS)*, IEEE, Marrakech, Morocco, April 2014.

[64] N. Garg and P. Rani, "An improved aodv routing protocol for vanet (vehicular Ad Hoc network)," *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 4, no. 16, p. 1024, 2015.

[65] S. Dhankhar and S. Agrawal, "VANETs: a survey on routing protocols and issues," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, no. 6, pp. 13427–13435, 2014.

[66] A. Moravejosharieh, H. Modares, R. Salleh, and E. Mostajeran, "Performance analysis of aodv, aomdv, dsr, dsdv routing protocols in vehicular ad hoc network," *Research Journal of Recent Sciences ISSN*, vol. 2277, p. 2502, 2013.

[67] B. Mokhtar and M. Azab, "Survey on security issues in vehicular ad hoc networks," *Alexandria engineering journal*, vol. 54, no. 4, pp. 1115–1126, 2015.

[68] M. K. Nasir, M. K. Sohel, M. T. Rahman, and A. K. Islam, "A review on position based routing protocol in vehicular adhoc network," *American Journal of Engineering Research*, vol. 2, no. 2, pp. 7–13, 2013.

[69] B. Pete and P. Jaini, "Continuous connectivity aware routing in VANET using hybrid protocol," in *Proceedings of the 2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, IEEE, Coimbatore, India, Febuary 2015..

[70] S. Boussoufa-Lahlah, F. Semchedine, and L. Bouallouche-Medjkoune, "A position-based routing protocol for vehicular ad hoc networks in a city environment," *Procedia Computer Science*, vol. 73, pp. 102–108, 2015.

[71] C. S. Evangeline and V. B. Kumaravelu, "Survey on VANET's layered architecture, security challenges and target network selection schemes," vol. 14, no. 24, pp. 4248–4262, 2006, https://www.researchgate.net/journal/Journal-of-Engineering-and-Applied-Sciences-1819-6608.

[72] M. D. Dikaiakos, S. Iqbal, T. Nadeem, and L. Iftode, "VITP: an information transfer protocol for vehicular computing," in *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pp. 30–39, Cologne, Germany, September 2005.

[73] K. Sanzgiri, D. LaFlamme, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "Authenticated routing for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 598–610, 2005.

[74] J.-W. Wang, H.-C. Chen, and Y.-P. Lin, "A secure destination-sequenced distance-vector routing protocol for ad hoc networks," *Journal of Networks*, vol. 5, no. 8, Article ID 942, 2010.

[75] R. Mishra, A. Singh, and R. Kumar, "VANET security: issues, challenges and solutions," in *Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 1050–1055, IEEE, Chennai, India, March 2016.

[76] C. Kumar Karn and C. Prakash Gupta, "A survey on VANETs security attacks and sybil attack detection," *International Journal of Sensors Wireless Communications and Control*, vol. 6, no. 1, pp. 45–62, 2016.

[77] C. S. Vorugunti and M. Sarvabhatla, "A secure and efficient authentication protocol in VANETs with privacy preservation," in *Proceedings of the Ninth International Conference on Wireless Communication and Sensor Networks*, pp. 189–201, Lecture Notes in Electrical Engineering, Springer, New Delhi, April 2014.

[78] F. D. Da Cunha, A. Boukerche, L. Villas, A. C. Viana, and A. A. Loureiro, "Data communication in VANETs: a survey, challenges and applications," vol. 44, 2014 https://www.researchgate.net/journal/Ad-Hoc-Networks-1570-8705.

[79] U. A. Khan and S. S. Lee, "Multi-layer problems and solutions in VANETs: a review," *Electronics*, vol. 8, no. 2, Article ID 204, 2019.

[80] Y. Liu, Y. Wang, and G. Chang, "Efficient privacy-preserving dual authentication and key agreement scheme for secure v2v communications in an IOV paradigm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2740–2749, 2017.

[81] Q. Zeng, Y. Tang, Z. Yu, and W. Xu, "A geographical routing protocol based on link connectivity analysis for urban VANETs," *Journal of Internet Technology*, vol. 21, no. 1, pp. 41–49, 2020.

[82] G. Sun, L. Song, H. Yu, V. Chang, X. Du, and M. Guizani, "V2v routing in a vanet based on the autoregressive integrated moving average model," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 908–922, 2018.

[83] K. A. Rahman and K. E. Tepe, "Towards a cross-layer based mac for smooth v2v and v2i communications for safety applications in dsrc/wave based systems,," in *Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 969–973, IEEE, Dearborn, MI, USA, June 2014.

[84] S. Kumar and K. S. Mann, "Prevention of dos attacks by detection of multiple malicious nodes in VANETs," in *Proceedings of the 2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, pp. 89–94, IEEE, London, UK,USA, April 2019.

[85] A. M. Malla and R. K. Sahu, "Security attacks with an effective solution for dos attacks in VANET," *International Journal of Computer Applications*, vol. 66, no. 22, pp. 975–8887, 2013.

[86] P. S. Waraich and N. Batra, "Prevention of denial of service attack over vehicle ad hoc networks using quick response table," in *Proceedings of the 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*, pp. 586–591, IEEE, Solan, India, September 2017.

[87] K. Jeffane and K. Ibrahimi, "Detection and identification of attacks in vehicular Ad Hoc network," in *Proceedings of the 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 58–62, IEEE, Fez, Morocco, October 2016.

[88] S. RoselinMary, M. Maheshwari, and M. Thamaraiselvan, "Early detection of dos attacks in VANET using attacked packet detection algorithm (apda)," in *Proceedings of the 2013 International Conference on Information Communication and Embedded Systems (ICICES)*, pp. 237–240, IEEE, Chennai, India, February 2013.

[89] A. Singh and P. Sharma, "A novel mechanism for detecting dos attack in VANET using enhanced attacked packet detection algorithm (eapda)," in *Proceedings of the 2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS)*, pp. 1–5, IEEE, Chandigarh, India, December 2015.

[90] G. A. Issac and A. J. Mary, "Validation scheme for VANET," in *Proceedings of the 2019 2nd International Conference on Signal Processing and Communication (ICSPC)*, pp. 11–15, IEEE, Coimbatore, India, March 2019.

[91] M. Sohail, R. Ali, M. Kashif et al., "Trustwalker: an efficient trust assessment in vehicular internet of things (viot) with security consideration," *Sensors*, vol. 20, no. 14, Article ID 3945, 2020.

[92] A. Wasef and X. Shen, "EMAP: expedite message authentication protocol for vehicular ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 78–89, 2013.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9944460, 17 pages
https://doi.org/10.1155/2021/9944460

Research Article

A Provably Secure Authentication and Key Exchange Protocol in Vehicular Ad Hoc Networks

Tsu-Yang Wu, Zhiyuan Lee, Lei Yang, and Chien-Ming Chen

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

Correspondence should be addressed to Chien-Ming Chen; chienmingchen@ieee.org

Received 17 March 2021; Revised 13 May 2021; Accepted 12 June 2021; Published 29 June 2021

Academic Editor: Azeem Irshad

While cloud computing and Internet of Things (IoT) technologies have gradually matured, mobile intelligent transportation systems have begun to be widely used. In particular, the application of vehicular ad hoc networks (VANETs) is very convenient for real-time collection and analysis of traffic data. VANETs provide a great convenience for drivers and passengers, making it easier to choose routes. Currently, most research on VANETs obtains data through cloud servers. However, there are few studies on cloud servers obtaining vehicle information through the roadside unit (RSU). In the process of reading traffic information, there will be some private and sensitive information, which may be intercepted or tampered with in untrusted public channels. Therefore, it is necessary to propose a protocol to protect vehicle data during the information reading phase. In this paper, we propose a new provably secure authentication protocol to negotiate a session key before transmitting traffic information. This protocol can complete mutual authentication and generate a session key. Finally, security analysis and performance analysis show that our protocol is secure and efficient.

1. Introduction

Due to social and economic development, motor vehicles are rapidly spreading. At the same time, the rapid increase in the number of vehicles on the road has also made the traffic situation more complicated, and there will be many traffic problems, such as traffic accidents and road congestion. Therefore, researchers apply artificial intelligence [1–4], wireless networks, and sensor technology [5, 6] to road vehicle management, so that vehicles can share information and release relevant road information to alleviate traffic problems. This is the vehicular ad hoc network, which consists of vehicle-to-vehicle (V2V) communication and vehicle-to-infrastructure (V2I) communication. In the VANETs, the vehicle is equipped with an on-board unit (OBU), so the vehicle can be regarded as a mobile network node that can communicate. Therefore, the vehicle can obtain the corresponding road information from the cloud server through the RSU and can also send the relevant information recorded by itself to the cloud server. The main goal of VANET technology is to improve traffic efficiency and increase driving experience. With the popularity and development of VANET, it plays a key role in user travel planning and road safety.

Although VANETs have various obvious benefits, their security and privacy issues [7–9] are still the keys to whether they can be widely used. In VANETs, the network environment is open; attackers can capture various messages transmitted in the network and can forge a legitimate vehicle to send wrong information. The transmission of wrong information will mislead the driver to make the wrong decision, bringing corresponding troubles and even dangers. First, before information transmission, mutual authentication must be performed, and a corresponding session key must be generated for subsequent information transmission. Then the integrity of the message must be verified every time a message is received. In addition, anonymity is indispensable in VANET, because if the vehicle transmits its identity on the network in clear text, the attacker captures the information, and the vehicle can be faked or the vehicle can be tracked.

However, several kinds of research in VANETs mainly focus on how to ensure that vehicles obtain corresponding road information. In other aspects, vehicles can receive current traffic conditions through RSU. Based on the information received, the driver can adjust the driving decision. Because the road conditions are changing, the RSU can actively establish a communication request with the vehicle to obtain the road condition information stored by the vehicle sensor (as shown in Figure 1). Based on our best knowledge, we propose a new provably secure mutual authentication scheme for negotiating session keys before transmitting traffic information in this paper. The main contributions of this paper are summarized as follows:

(1) A three-party AKE scheme is proposed, with vehicles, RSU, and cloud servers. RSU actively sends a request, completes mutual authentication with the vehicle through the cloud server, and generates a session key.

(2) Due to environmental constraints, the proposed scheme only performs simple operations, such as elliptic curve (ECC), bitwise XOR, and hash functions.

(3) We conduct a security analysis of the protocol, including formal analysis, informal analysis, and ProVerif simulation.

(4) Finally, the performance of the proposed protocol is evaluated. Compared with the existing methods, we show that our protocol is feasible.

The remainder of this paper is organized as follows. In Section 2, the latest research results of the AKE protocol and related research on security authentication in the VANET environment are reviewed. Section 3 describes our proposed protocol in detail. Then, in Sections 4 and 5, the security analysis and performance analysis of the protocol proposed in Section 3 are carried out. Finally, the article is summarized in Section 6.

2. Related Work

Many researchers have conducted a series of studies on authentication and key exchange protocols in VANETs. However, with the changes of various needs and scenarios, many security issues have emerged in these studies.

First of all, in terms of an authentication protocol, Lamport [10] proposed for the first time password authentication in an insecure channel. Immediately afterward, various two-party authentication schemes were proposed [11, 12]. But, for the VANETs environment, the communication between vehicles can use a two-party authentication scheme, and if the vehicle and the cloud server are authenticated, the two-party authentication will cause transmission delay, because two-party identity authentication is generally used in a single-server environment. In 2001, Li et al. [13] first proposed an authentication scheme in a multiserver environment, but their scheme is inefficient because it takes a lot of time to train neural networks. Later, to complete efficient and secure identity authentication, researchers began to introduce multifactor security. In addition to passwords, security factors such as smart cards and biological information were introduced [14–16]. Recently, Irshad et al. [17] proposed an authentication scheme under a multiserver architecture based on the chaotic mapping. But Wu et al. [18] found that Irshad et al.'s protocol cannot guarantee user anonymity and is vulnerable to attacks by privileged insiders. Therefore, Wu et al. proposed an authentication protocol for distributed cloud environments, claiming that their protocol can resist various known attacks. However, Wu et al. [19] recently proposed an authentication key exchange protocol under a multiserver architecture and found that [18] has multiple security problems, including the inability to provide perfect forward secrecy (PFS) and being vulnerable to privileged internal attacks. Also, in a multiserver environment, in 2017, Truong et al. [20] proposed an ECC-based authentication scheme. Their article discussed that Yeh et al.'s [21] protocol cannot provide mutual authentication and the key agreement phase is incorrect. In 2018, Zhao et al. [22] proposed a secure and efficient authentication protocol based on passwords and smart cards. They claimed that the scheme of Truong et al. could not achieve the security authentication requirements of multiserver authentication and could not resist offline password guessing and impersonation attacks. However, Hassan et al. [23] conducted a security analysis on the scheme proposed by Zhao et al. and found that the scheme is vulnerable to anonymity and traceability issues and is not suitable for a multiserver environment. Then, on this basis, Hassan et al. proposed an improved multiserver authentication scheme.

Currently, there are two research focuses on the VANETs environment; one is efficient authentication, and the other is privacy protection. The former appeared because of the large number of vehicles in the VANETs environment, and data transmission and processing are very challenging. In order to solve this problem, cloud computing began to be applied to the VANETs [24]. In VANETs, cloud computing-based authentication schemes have also begun to be widely proposed [25–29]. These solutions reduce the server-side service response time and improve authentication efficiency. However, due to the number of vehicles involved and management issues, network delays can also be caused. Then cloud computing began to decentralize and fog computing was used to solve the above shortcomings [15, 30–32]. The latter is because, in an open network environment, the private information of vehicle users must be protected. Therefore, the Conditional Privacy Preservation Authentication (CPPA) agreement was proposed [33]. In this protocol, the attacker cannot obtain the true identity of the vehicle user through messages intercepted on the public channel, but a trusted third party can calculate the identity of the vehicle user who sent the message. In 2008, Zhang et al. [34] proposed an identity-based verification scheme and proved that their proposed scheme can practice conditional privacy protection, trusting the authority to retrieve the true identity of the vehicle from any false identity. In 2014, Chuang and Lee [35] proposed the first authentication mechanism using transitive trust relationship. Later, Zhou et al. [36] used elliptic curve cryptography (ECC) to

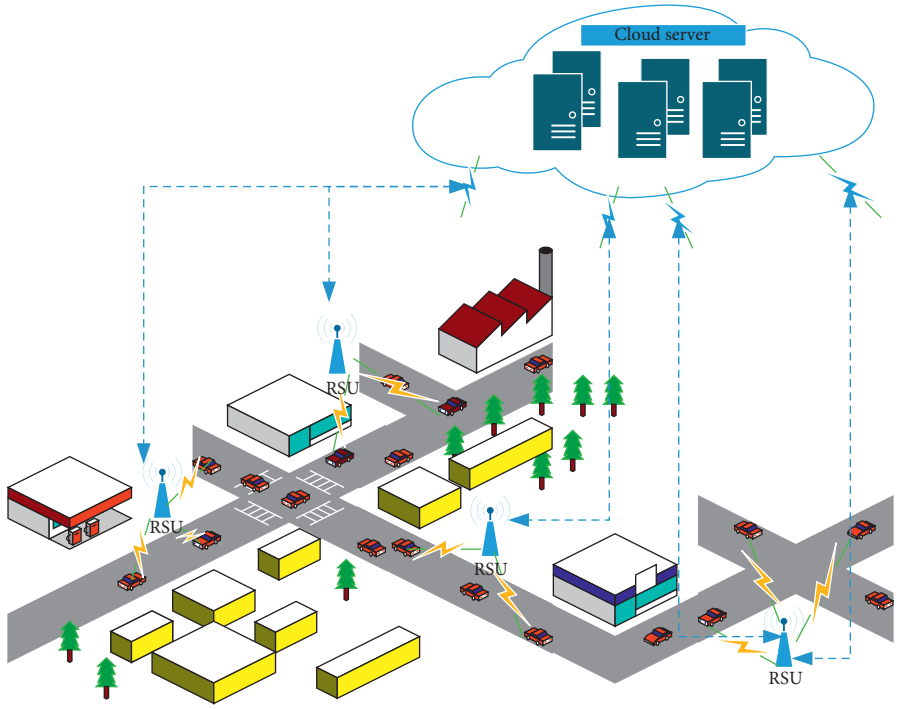


Figure 1: A typical VANETs structure.

propose a new mutual authentication scheme based on the mechanism proposed by Chuang and Lee and mentioned in their paper that the scheme of Chang and Lee cannot resist internal attacks. However, Wu et al. [37] found that Zhou et al.'s scheme could not guarantee anonymity and was vulnerable to identity guessing and impersonation attacks. At the same time, they designed a new privacy protection authentication protocol using ECC technology. Some researchers have proposed the use of fog computing for information processing in the VANET environment. In 2019, Ma et al. [30] proposed a new AKE protocol without bilinear pairing. They believed that the proposed protocol is safe and efficient. However, Eftekhari et al. [38] found that the protocol of Ma et al. had security problems, such as internal attacks, known session-specific temporary information attacks, and stolen smart card attacks, and then they proposed a safer and more efficient protocol. In 2017, Mohit et al. [39] proposed a new vehicle communication protocol and believed that their protocol could resist attacks such as stolen smart card attacks and impersonation attacks. However, Yu et al. [40] found that Mohit et al.'s scheme could not provide security attributes such as anonymity and mutual authentication and would suffer impersonation and traceability attacks. Then Yu et al. proposed a new security authentication protocol and proved that their protocol can resist various known attacks. In 2020, Sadri and Rajabzadeh Asaar [41] proved that Yu et al.'s protocol is vulnerable to tracking attacks, impersonation attacks, sensor capture attacks, and so forth and proposed a secure protocol for application in VANETs.

Some studies have begun to design the AKE protocol for the advantages of low latency and high reliability in the 5G environment [42]; and, for some special occasions, blockchain technology [43] is also used to complete the authentication key exchange. Research similar to VANET currently has similar flying ad hoc networks (FANETs). Moreover, this environment is also vulnerable to serious security threats. Due to these security threats, many security protocols have been proposed in this environment [44–47]. Therefore, when studying VANETs, you can refer to some security solutions in FANETs. However, most of the research is carried out on the premise that the vehicle initiates a communication request. So, it is necessary to propose an authentication scheme in which a cloud server or RSU initiates a communication request to the vehicle user to meet the timely update of road condition information.

3. Proposed Scheme

In this section, we introduce in detail a new provably secure mutual authentication scheme used to negotiate session keys before transmitting traffic information. The communication entities in the proposed protocol include vehicle users, roadside units, and cloud servers. For the convenience of reading, the symbols used in the scheme are listed in Table 1. The proposed protocol has five phases, namely, the initialization phase, the vehicle registration phase, the RSU registration phase, the login phase, and the authentication phase.

3.1. Initialization Phase

(1) The cloud server CS selects two large prime numbers p and q and then constructs an elliptic curve E defined about the domain Z_q for q. The points on E

Table 1: Notations and descriptions.

Notations	Description
V_i	The i_{th} vehicle end user
RSU_j	The j_{th} roadside unit
CS	The cloud server
VID_i	V_i's identity
Pw_i	V_i's password
Bio_i	Biometric features of V_i
$\text{Gen}(\cdot)/\text{Rep}(\cdot)$	Generation/reproduction process of fuzzy extractor
x	The secret key of CSP
PVID_i	The pseudoidentities of V_i
SK	Session key
$\mathcal{A}$	The attacker
$h(\cdot)$	One-way hash function
$x \Vert y$	Concatenation
$x \oplus y$	The exclusive-or operation with x and y

form a cyclic additive elliptic curve group G, and the generator P of G is obtained.

(2) CS selects two random numbers x and α and computes $\beta = \alpha \cdot P$, where x is the long-term key of the CS, α is the private key, and β is the public key.

(3) Finally, CS chooses a one-way hash function $h(\cdot)$.

3.2. Vehicle User Registration Phase. When the vehicle user V_i wants to get the corresponding service, he/she must register through the cloud server CS. The main steps are as follows. Figure 2 describes the process of vehicle user registration in detail.

(1) V_i chooses its own VID_i and then sends it to CS through a secret channel.

(2) On receiving $\{\text{VID}_i\}$, CS selects n_i and computes $K_v = h(\text{VID}_i \Vert h(x \Vert n_i))$ and $\text{PVID}_i = h(\text{VID}_i \Vert K_v)$. Then, CS saves $\{\text{PVID}_i, \text{VID}_i, n_i\}$ to memory and securely transmits $\{\text{PVID}_i, K_v\}$ to V_i.

(3) Finally, V_i computes $(\phi_i, \theta_i) = \text{Gen}(\text{Bio}_i)$, $\text{HP}_i = K_v \oplus h(\text{Pw}_i \Vert \phi_i)$, and $\text{Auth}_i = h(K_v \Vert \text{VID}_i)$ and stores $\{\text{PVID}_i, \text{Auth}_i, \text{HP}_i, \theta_i\}$ into OBU. Among them, Pw_i is the V_i's password, and Bio_i is the V_i's biological information.

3.3. RSU Registration Phase. Through the registration phase, RSU_j can obtain the private key, as shown in Figure 3.

(1) RSU_j selects a random number c_j and computes $d_j = c_j \cdot P$ and then sends the identity RID_j and d_j to CS securely.

(2) CS selects the pseudoidentity PRID_j of RSU_j and the random number k_j. Then CS computes $y_j = k_j \cdot P + d_j$ and $z_j = k_j + (y_j + \text{PRID}_j) \cdot \alpha \bmod p$, stores $\{\text{PRID}_j, \text{RID}_j, y_j, k_j\}$ in its database, and finally sends $\{\text{PRID}_j, y_j, z_j\}$ to RSU_j.

(3) RSU_j computes $x_j = z_j + c_j$ and then verifies whether $x_j \cdot P$ is equal to $z_j \cdot P + c_j \cdot P$. If the verification is passed, the private key distribution is successful. Then $\{\text{PRID}_j, y_j, z_j\}$ is stored in RSU_j memory.

3.4. Login Phase. Since the environment proposed by the scheme is to complete mutual authentication and key exchange during vehicle operation, the vehicle user login will be completed in advance. Figure 4 shows the login information of the vehicle user.

3.5. Authentication Phase. The entire authentication phase is initiated by RSU_j, which wants to communicate with the running vehicle. The detailed information is shown in Figure 5.

(1) First, RSU_j makes a communication request Request and selects a random number r_j to compute $R_j = r_j \cdot P$. RSU_j sends Request and R_j to V_i.

(2) After V_i receives the communication request, it selects a random number a_i and the current timestamp T_1 and computes N_i, M_i, C_1, C_2 (see equations (1)–(4)). Then it sends $\{N_i, C_1, C_2, T_1\}$ to RSU_j.

$$N_i = a_i \cdot P, \tag{1}$$

$$M_i = a_i \cdot \beta, \tag{2}$$

$$C_1 = \text{PVID}_i + h(M_i \Vert R_j \Vert N_i \Vert T_1), \tag{3}$$

$$C_2 = h(\text{PVID}_i \Vert \text{VID}_i \Vert N_i \Vert T_1). \tag{4}$$

(3) RSU_j verifies the validity of the timestamp (by $|T_2 - T_1| < \Delta T$). RSU_j computes M_j, C_3, C_4 (as shown in equations (5)–(7)). Finally, RSU_j sends $\{R_j, C_3, C_4, T_2\}$ to CS.

$$M_j = r_j \cdot \beta, \tag{5}$$

$$C_3 = \text{PRID}_j + h(M_j \Vert R_j \Vert T_2), \tag{6}$$

$$C_4 = h(\text{PRID}_j \Vert \text{RID}_j \Vert M_j \Vert T_2). \tag{7}$$

(4) After CS receives the message, it first verifies whether the timestamp is valid (by $|T_3 - T_2| < \Delta T$). If the verification is passed, it computes $M_i' = x_j \cdot N_i$ and $\text{PVID}_i' = C_1 - h(M_i' \Vert R_j \Vert N_i \Vert T_1)$. Then it takes out VID_i from the memory through PVID_i and computes $C_2' = h(\text{PVID}_i' \Vert \text{VID}_i \Vert N_i \Vert T_1)$. If C_2' and C_2 are equal, then perform the operation; otherwise, terminate the session. Then, CS computes $M_j' = \alpha \cdot R_j$ and $\text{PRID}_j' = C_3 - h(M_j' \Vert R_j \Vert T_2)$ and then retrieves RID_j in the database through PRID_j. After that, CS computes $C_4' = h(\text{PRID}_j' \Vert \text{RID}_j \Vert M_j' \Vert T_2)$ and completes the authentication operation. If authenticated, CS selects a random number b_s and a

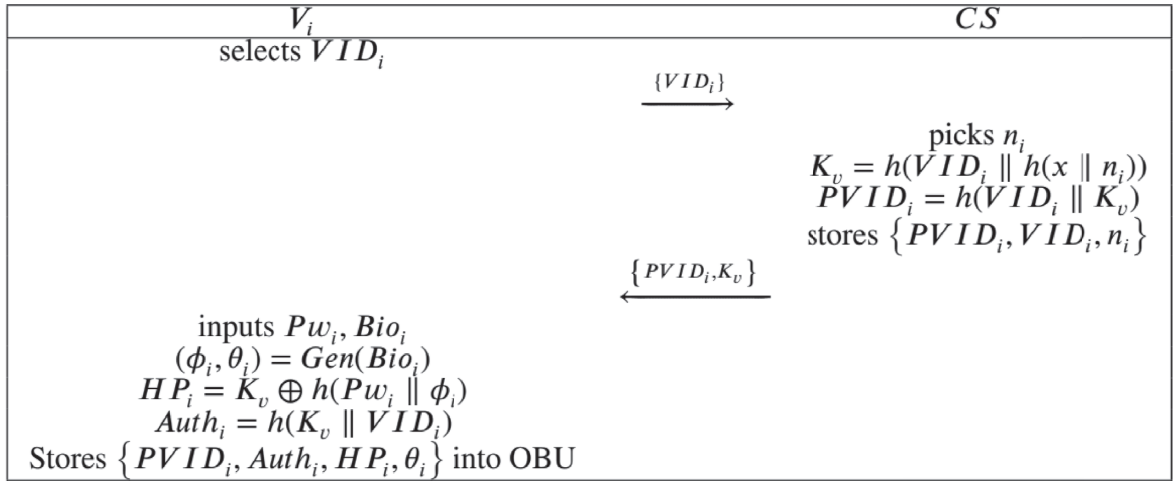


Figure 2: Vehicle user registration phase.

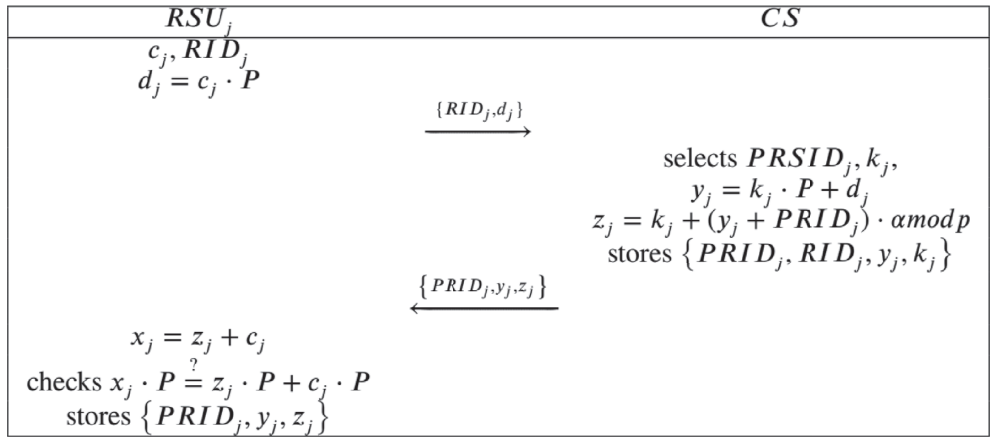


Figure 3: Registration phase.

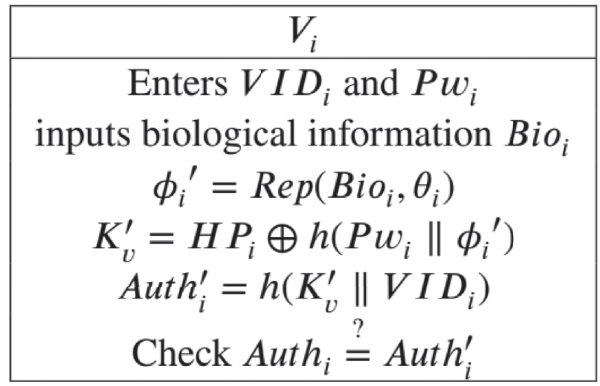


Figure 4: Login phase.

timestamp T_3 and computes $N_s, C_5, C_6, \text{PVID}_{\text{inew}}, \text{PRID}_{\text{jnew}}, C_7$ (see equations (8)–(13)). Finally, CS updates the values of PRID_j and PVID_i in memory and sends $\{C_5, C_6, C_7, N_s, T_3\}$ to RSU_j.

$$N_s = b_s \cdot P, \tag{8}$$

$$C_5 = b_s \cdot \left(y_j + \left(y_j + \text{PRID}_j\right) \cdot \beta\right), \tag{9}$$

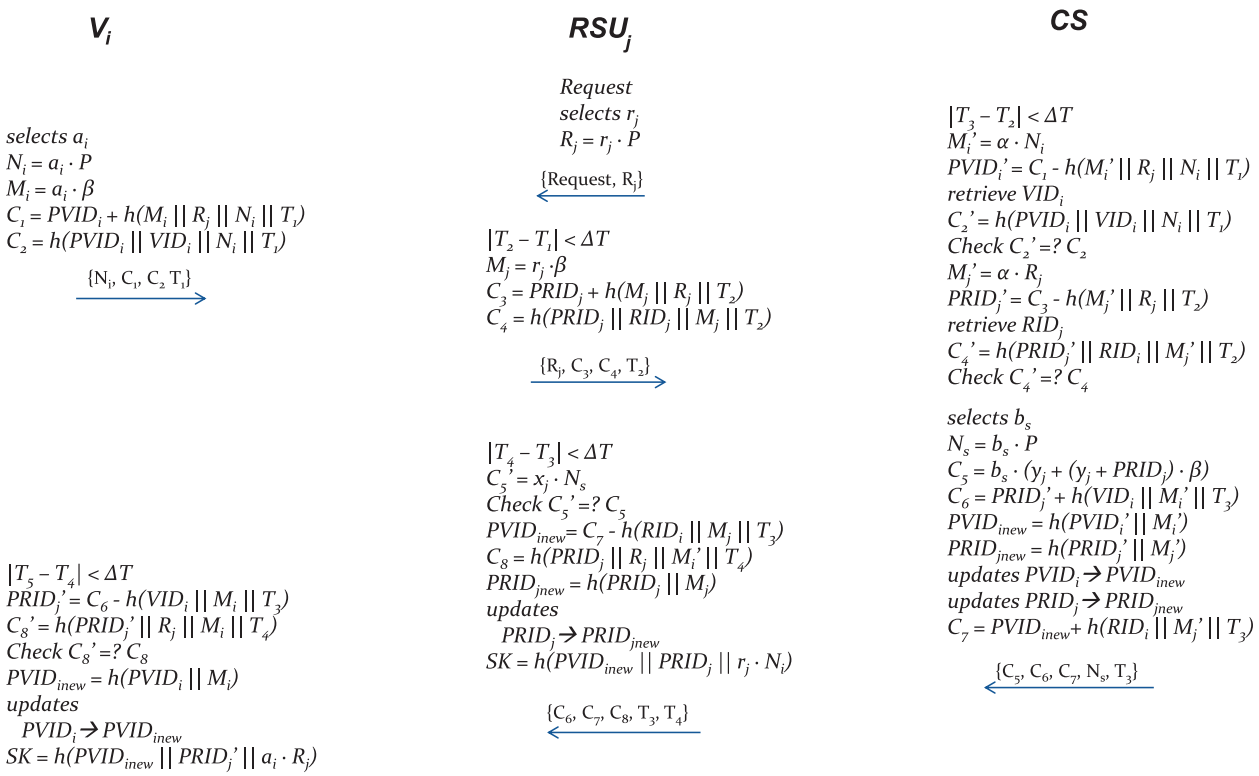


Figure 5: Authentication phase.

$$C_6 = \mathrm{PRID}_j' + h(\mathrm{VID}_i \| M_i' \| T_3), \tag{10}$$

$$\mathrm{PVID}_{\mathrm{inew}} = h(\mathrm{PVID}_i' \| M_i'), \tag{11}$$

$$\mathrm{PRID}_{\mathrm{jnew}} = h(\mathrm{PRID}_j' \| M_j'), \tag{12}$$

$$C_7 = \mathrm{PVID}_{\mathrm{inew}} + h(\mathrm{RID}_i \| M_j' \| T_3). \tag{13}$$

(5) RSU_j also verifies the validity of the timestamp. Then it computes $C_5' = x_j \cdot N_s$ and verifies that C_5' is equal to C_5. If authenticated, RSU_j computes $\mathrm{PVID}_{\mathrm{inew}}$, C_8, $\mathrm{PRID}_{\mathrm{jnew}}$ (see equations (14)–(16)). Then it updates the values of PRID_j in memory. Finally, RSU_j computes the session key $\mathrm{SK} = h(\mathrm{PVID}_{\mathrm{inew}} \| \mathrm{PRID}_j \| r_j \cdot N_i)$ and sends $\{C_6, C_7, C_8, T_3, T_4\}$ to V_i.

$$N_s = b_s \cdot P, \tag{14}$$

$$C_8 = h(\mathrm{PRID}_j \| R_j \| M_i' \| T_4), \tag{15}$$

$$\mathrm{PRID}_{\mathrm{jnew}} = h(\mathrm{PRID}_j \| M_j). \tag{16}$$

(6) After V_i receives the message, it checks the freshness of timestamp. If it is confirmed, V_i computes $\mathrm{PRID}_j' = C_6 - h(\mathrm{VID}_i \| M_i \| T_3)$ and $C_8' = h(\mathrm{PRID}_j' \| R_j \| M_i \| T_4)$ and then verifies $C_8' ? = C_8$. If authenticated, V_i computes a new $\mathrm{PVID}_{\mathrm{inew}} = h(\mathrm{PVID}_i \| M_i)$ and updates this value in the memory. Finally, V_i computes the session key $\mathrm{SK} = h(\mathrm{PVID}_{\mathrm{inew}} \| \mathrm{PRID}_j' \| a_i \cdot R_j)$.

4. Security Analysis

In this section, we conduct a security analysis of the proposed protocol and use the ROR model and ProVerif tool to complete the formal security analysis [48, 49]; and, through informal security analysis, we verified that the proposed protocol has security features and can resist various known attacks.

4.1. Informal Security Analysis. This section is an informal security analysis of the proposed protocol. We verify the security attributes and attacks that the proposed protocol needs to have one by one.

4.1.1. Mutual Authentication. After receiving the authentication request from RSU_j, V_i computes $C_2 = h(\mathrm{PVID}_i \| \mathrm{VID}_i \| N_i \| T_1)$ and sends it to CS through RSU_j. After CS receives the RSU_j message, the computed C_2 contains the parameters $\{\mathrm{PVID}_i, \mathrm{VID}_i\}$. Only legitimate users can generate correct C_2, so that CS can verify the identity of the user and the legitimacy of the information by verifying whether C_2' is equal to C_2; that is, CS authenticates V_i. Similarly, the server computes C_4, RSU_j computes C_5, and V_i computes C_8, respectively, indicating that CS has authenticated RSU_j, RSU_j has authenticated CS, and V_i has authenticated CS. In summary, V_i and RSU_j can perform mutual authentication in the protocol.

4.1.2. Man-in-the-Middle Attacks. By intercepting the information in the public channel, $\mathscr{A}$ may launch man-in-the-

middle attacks. But after CS receives the message, it needs to verify $C_2? = h(\text{PVID}_i' \| \text{VID}_i \| N_i \| T_1)$ and $C_4? = h(\text{PRID}_j' \| \text{RID}_j \| M_j' \| T_2)$ to authenticate the sender. Suppose that when $\mathscr{A}$ tries to tamper with the information sent to RSU_j, he needs to generate a new authentication information C_5, but he cannot obtain the parameters x_j, b_s, and so forth. This means that $\mathscr{A}$ cannot complete the verification after tampering with the information. Similarly, when $\mathscr{A}$ tampered with the information sent to V_i and CS, he could not complete the relevant authentication. This shows that the protocol can resist man-in-the-middle attacks.

4.1.3. Replay Attacks. In the protocol, when a new round of authentication is performed, new random numbers r_j, a_i, and b_s will be generated; and every time the authentication is completed, the values stored in the memory such as PVID_i and PRID_j will be updated. The random number and the updated PVID_i are used when generating the session key. Therefore, when $\mathscr{A}$ resends the previous message, new random numbers and related parameters updated in the memory have been generated, and he cannot pass the verification and cannot compute the session key. Therefore, the proposed protocol can resist replay attacks.

4.1.4. Known Session-Specific Temporary Information Attacks. Under the CK attack model [50], $\mathscr{A}$ can obtain the random number a_i or r_j generated during the authentication phase. Assuming that $\mathscr{A}$ obtains the random number a_i generated by V_i; then N_i, M_i, and PVID_i can be calculated. However, since $\mathscr{A}$ cannot obtain VID_i and PRID_j, he still cannot compute the session key SK; and when $\mathscr{A}$ tries to use a random number to perform a man-in-the-middle attack or an impersonation attack, he cannot complete the verification by recalculating C_2. Therefore, the proposed protocol can resist known session-specific temporary information attacks.

4.1.5. Perfect Forward Secrecy. This security feature requires that the leakage of the long-term key does not reveal the previously generated session key. $\text{SK} = h(\text{PVID}_{\text{inew}} \| \text{PRID}_j \| a_i \cdot R_j) = h(\text{PVID}_{\text{inew}} \| \text{PRID}_j \| r_j \cdot N_i)$ in the scheme. That is, the long-term key x of CS is not used in the calculation of the session key. Since the private key α of CS does not change after each authentication, it is assumed that $\mathscr{A}$ can get α. Then $\mathscr{A}$ can compute $M_i = \alpha \cdot N_i$ and $M_j = \alpha \cdot R_j$; that is, $\text{PVID}_i = C_1 - h(M_i \| R_j \| N_i \| T_1)$ and $\text{PRID}_j = C_3 - h(M_j \| R_j \| T_2)$, and the updated $\text{PVID}_{\text{inew}} = h(\text{PVID}_i \| M_i)$ and $\text{PRID}_{\text{jnew}} = h(\text{PRID}_j \| M_j)$. However, $\mathscr{A}$ cannot obtain the random number a_i or r_j needed to compute SK, so there is no way to compute SK; that is, the proposed protocol can provide perfect forward secrecy.

4.1.6. Internal Attacks. Assuming that $\mathscr{A}$ is a CS internal staff, he can easily obtain the information transmitted during the registration phase, including $\{\text{VID}_i\}$, $\{\text{PVID}_i, K_v\}$, $\{\text{RID}_j, d_j\}$, and $\{\text{PRID}_j, y_j, z_j\}$. However, $\mathscr{A}$ cannot compute a_i and r_j from this information. Therefore, the proposed protocol can resist internal attacks.

4.1.7. User Anonymity and Untraceability. During the authentication process, VID_i is used to compute C_2 and $\mathscr{A}$ cannot obtain PVID_i to guess VID_i. So, the scheme can guarantee anonymity. At the same time, due to the use of random numbers and the update of the pseudoidentity after each authentication, it is also ensured that $\mathscr{A}$ cannot confirm the user's identity by tracing a specific piece of information. Therefore, the protocol satisfies anonymity and untraceability.

4.1.8. Three-Factor Secrecy. The proposed protocol uses passwords, biological information, and storage devices (OBU) for security encryption, so it is a three-factor authentication protocol. For this type of protocol, it is assumed that the extreme case is that $\mathscr{A}$ can obtain two of the three factors and can launch an attack on the protocol.

Assume that $\mathscr{A}$ obtains VID_i, Pw_i, and Bio_i. It is necessary to compute Auth_i when logging in, where $K_v = HP_i \oplus h(Pw_i \| \phi_i)$, but HP_i is stored in OBU. In other words, $\mathscr{A}$ cannot complete the login operation. The proposed protocol is safe in this situation. Assume that $\mathscr{A}$ obtains VID_i, Pw_i, and OBU. Since ϕ_i cannot be computed through Bio_i, $\mathscr{A}$ cannot compute K_v and Auth and cannot complete login verification. That is, the protocol is safe in this situation. Similarly, when $\mathscr{A}$ knows Bio_i and OBU, there is no way to compute Auth_i because there is no password and identity. Therefore, the protocol is safe in the three situations, and the proposed protocol satisfies the three-factor security characteristics.

4.1.9. No Key Control. In this protocol, the session key SK can only be generated through negotiation between V_i and RSU_j; that is, a single entity cannot generate SK by itself. When computing SK, V_i needs to know PRID_j and R_j generated by RSU_j. In the same way, RSU_j needs to negotiate to obtain $\text{PVID}_{\text{inew}}$ and N_i during the calculation to compute SK. Therefore, the proposed protocol is satisfied with no key control property.

4.2. Formal Security Analysis Based on Random Oracle Model. In this section, a random oracle model (ROR model) is used to formally prove the security of our proposed protocol. This analysis model was proposed by Canetti et al. [51]. By launching different rounds of Games, the ROR model can compute the probability of $\mathscr{A}$ successfully guessing the SK in various situations and thus judge the security of the protocol. Assume that I_V^x, I_{RSU}^y, and I_{CS}^z, respectively, represent the xth communication of V_i, the yth communication of RSU_j, and the zth communication of CS. $\mathscr{A}$ can initiate the following query, where $O = \{I_V^x, I_{\text{RSU}}^y, I_{\text{CS}}^z\}$.

(i) Execute(O): through this query, $\mathscr{A}$ can eavesdrop on the message transmitted on the public channel.

(ii) Hash(string): $\mathscr{A}$ executes the query and can get the hash value of the input parameter string.

(iii) Send(O, M): $\mathscr{A}$ executes the query, sends a message M to O, and can receive the corresponding response.

(iv) Reveal(O): $\mathscr{A}$ executes this query to obtain the return result of current session key SK generated by M.

(v) Corrupt(O): by executing the query, $\mathscr{A}$ can obtain some secret values, such as long-term private keys and temporary information.

(vi) Test(O): $\mathscr{A}$ executes the query and judges the correctness of the session key by flipping coin C. If the result is $C = 1$, $\mathscr{A}$ will receive the correct session key returned; if the result is $C = 0$, $\mathscr{A}$ will receive a random string.

Definition 1. (elliptic curve discrete logarithm problem (ECDLP)). Our proposed protocol uses elliptic curve cryptography (ECC). Here, we describe the computational difficulties and assumptions of ECC. Suppose that C is an elliptic curve generation group. At the same time, given points P and $a \cdot P$, where P belongs to C and a belongs to F_p, it is computationally infeasible to obtain a. In polynomial time, the probability that $\mathscr{A}$ solves this problem is defined as follows: $\mathrm{Adv}_{\mathscr{A}}^{\mathrm{ECDLP}}(\xi) = Pr[A(P, aP) = a: a \in F_p, P \in \xi]$. For a sufficiently small η, we have $\mathrm{Adv}_{A}^{\mathrm{ECDLP}}(\xi) < \eta$.

Theorem 1. *If $\mathscr{A}$ attempts to initiate some queries in polynomial time, then the advantage that he can break through the proposed protocol P is as follows: $Adv_{\mathscr{A}}^{P}(\xi) = (q_{hash}^2/2^l)Adv_{\mathscr{A}}^{ECDLP}(\xi) + 2max\{C' \cdot q_{send}^s, (q_{send}/2^l)\} + 2q_{send}$ $Adv_{\mathscr{A}}^{ECDLP}$ $(\xi) + ((q_{send} + q_{exe})^2/p) + (q_{hash}^2/2^l) + (q_{send}/2^{(l-1)})$, where q_{hash} represents the number of times to execute Hash(string) queries, q_{send} represents the number of times to execute Send(O, M) queries, q_{exe} represents the number of times to Execute(O) queries, l represents the number of bits of the operation, and C' and s are constants in Zipf's law [52].*

Proof. We use the game sequence $GM_0, GM_1, GM_2, GM_3, GM_4, GM_5, GM_6$ to verify the above theorem. $\mathrm{Succ}_{\mathscr{A}}^{GM_n}(\xi)$ represents the probability of $\mathscr{A}$'s success in game GM_n. Finally, using the Test query to determine $\mathrm{Succ}_{\mathscr{A}}^{GM_5}(\xi)$, the specific description is as follows:

(i) **Game** GM_0: GM_0 represents a real attack, and $\mathscr{A}$ did not initiate any query at this time. Therefore, in GM_0, the probability of $\mathscr{A}$ cracking P is $\mathrm{Adv}_{\mathscr{A}}^{\mathscr{P}}(\xi) = |2\mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{\mathrm{GM}_0}(\xi)] - 1|$.

(ii) **Game** GM_1: GM_1 adds Execute query on the basis of GM_0, and there is no difference in the others. So, $\mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{\mathrm{GM}_1}(\xi)] = \mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{\mathrm{GM}_0}(\xi)]$.

(iii) **Game** GM_2: GM_2 adds Send query on the basis of GM_1. According to Zipf's law, we get $|\mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{GM_2}(\xi)] - \mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{GM_1}(\xi)]| \le q_{\mathrm{send}}/2^l$.

(iv) **Game** GM_3: GM_3 adds Hash query on the basis of GM_2. According to the birthday paradox, we can get the maximum probability of hash collision as $q_{\mathrm{hash}}^2/2^{l+1}$; the maximum probability of collision in the transmitted text is $(q_{\mathrm{send}} + q_{\mathrm{exe}})^2/2p$; and so $|\mathrm{Pr}[\mathrm{Succ}_{A}^{GM_3}(\xi)] - \mathrm{Pr}[\mathrm{Succ}_{A}^{GM_2}(\xi)]| \le (q_{\mathrm{send}} + q_{\mathrm{exe}})^2/2p + q_{\mathrm{hash}}^2/2^{l+1}$.

(v) **Game** GM_4: In this game, we consider the security of the session key. Here, we divide the discussion into two situations. The first is to obtain a long-term private key to verify perfect forward secrecy; the second is to provide temporary information leakage to verify whether the known session-specific temporary information attacks can be resisted.

Perfect forward secrecy: $\mathscr{A}$ uses Corrupt(I_{RSU}^y) to try to get the private key x_j of RSU_j or uses Corrupt(I_V^x) or Corrupt(I_{Cs}^z) to try to get a certain secret value in the registration phase

Known session-specific temporary information attacks: $\mathscr{A}$ uses Corrupt(I_V^x) or Corrupt(I_{RSU}^y) or Corrupt(I_{Cs}^z) to try to obtain temporary information of one party

In both cases, ECDLP needs to be solved to compute the session key SK. For $\mathrm{SK} = h(\mathrm{PVID}_{\mathrm{inew}} \| \mathrm{PRID}_j \| a_i \cdot R_j)$, in the first case, even if M_i and PVID_i are calculated by x_j, the random number r_j is unknown. While getting a_i through Corrupt(I_V^x), $\mathscr{A}$ cannot get $\mathrm{VID}_i, \mathrm{PVID}_i$. In the second case, even if $a_i \cdot R_j$ is calculated through a_i, the long-term private key is unknown. Similarly, for the second formula $\mathrm{SK} = h(\mathrm{PVID}_{\mathrm{inew}} \| \mathrm{PRID}_j \| r_j \cdot N_i)$ also holds, $|\mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{GM_4}(\xi)] - \mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{GM_3}(\xi)]| \le q_{\mathrm{send}}\,\mathrm{Adv}_{\mathscr{A}}^{\mathrm{ECDLP}}(\xi)$.

(i) **Game** GM_5: GM_5 uses Corrupt(I_V^x) to query; $\mathscr{A}$ can get the information $(\mathrm{PVID}_i, \mathrm{Auth}_i, \mathrm{HP}_i, \theta_i)$ in OBU. The user uses the password and biological information to register. $\mathscr{A}$ wants to guess +, but the possibility of guessing the biological characteristics is $(1/2^l)$, which can be almost ignored. Using Zipf's law, we can get $|\mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{GM_5}(\xi)] - \mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{GM_4}(\xi)]| \le \max\{C' q_{\mathrm{send}}^s, (q_{\mathrm{send}}/2^l)\}$.

(ii) **Game** GM_6: the purpose of this game is to verify forgery attacks. In GM_6, if $\mathscr{A}$ issues $h(\mathrm{PVID}_{\mathrm{inew}} \| \mathrm{PRID}_j \| r_j \cdot N_i)$ or $h(\mathrm{PVID}_{\mathrm{inew}} \| \mathrm{PRID}_j \| a_i \cdot R_j)$ query, the game is terminated. At this point, the probability of $\mathscr{A}$ guessing SK is $|\mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{GM_6}(\xi)] - \mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{GM_5}(\xi)]| \le (q_{\mathrm{hash}}^2/2^{l+1})$ $\mathrm{Adv}_{\mathscr{A}}^{\mathrm{ECDLP}}(\xi)$. Because the probability of success and unsuccess of GM_6 is half, $\mathrm{Pr}[\mathrm{Succ}_{\mathscr{A}}^{GM_6}(\xi)] = 1/2$.

In summary, we can get the following conclusions:

$$\begin{aligned}\frac{1}{2\mathrm{Adv}_{\mathscr{A}}^{\mathscr{P}}(\xi)} &= \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_0}(\xi)\right] - \frac{1}{2} = \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_0}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_6}(\xi)\right] \\ &= \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_1}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_6}(\xi)\right] \\ &\le \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_6}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_5}(\xi)\right] + \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_5}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_4}(\xi)\right] \\ &\quad + \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_4}(\xi)\right] - Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_3}(\xi)\right] + Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_3}(\xi)\right] \\ &\quad - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_2}(\xi)\right] + Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_2}(\xi)\right] - Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_1}(\xi)\right] \\ &= \frac{q_{\mathrm{hash}}^2}{2^{l+1}\mathrm{Adv}_{\mathscr{A}}^{\mathrm{ECDLP}}(\xi)} + \max\left\{C' q_{\mathrm{send}}^s, \frac{q_{\mathrm{send}}}{2^l}\right\} + q_{\mathrm{send}}\mathrm{Adv}_{\mathscr{A}}^{\mathrm{ECDLP}}(\xi) + \frac{\left(q_{\mathrm{send}} + q_{\mathrm{exe}}\right)^2}{2p} \\ &\quad + \frac{q_{\mathrm{hash}}^2}{2^{l+1}} + \frac{q_{\mathrm{send}}}{2^l}.\end{aligned} \tag{17}$$

Thus, we can obtain

$$\begin{aligned}\mathrm{Adv}_{\mathscr{A}}^{\mathscr{P}}(\xi) &= \frac{q_{\mathrm{hash}}^2}{2^l}\mathrm{Adv}_{\mathscr{A}}^{ECDLP}(\xi) \\ &\quad + 2\max\left\{C' \cdot q_{\mathrm{send}}^s, \frac{q_{\mathrm{send}}}{2^l}\right\} + 2q_{\mathrm{send}}\mathrm{Adv}_{\mathscr{A}}^{ECDLP}(\xi) \\ &\quad + \frac{\left(q_{\mathrm{send}} + q_{\mathrm{exe}}\right)^2}{p} + \frac{q_{\mathrm{send}}}{2^{l-1}}.\end{aligned} \tag{18}$$

□

4.3. ProVerif Security Analysis. ProVerif [53] is a formal cryptographic protocol security verification tool proposed by Bruno Blanchet in 2001 and developed using the Prolog language. The tool is based on the DY model and can handle basic cryptographic operations such as symmetric encryption and decryption, public-key encryption and decryption, hash operations, and XOR operations. The security attributes that can be verified are confidentiality, authentication, consistency, and equivalence between processes. Through the use of code to achieve the registration and authentication phases of vehicle users, RSU, and cloud server, a protocol simulation experiment is created in this section. The following is the whole process:

(1) The definition of the channel is ch and sch. The former is a common channel used in the login and authentication phases, and the latter is a secure channel used in the registration phase. SK_v and SK_r are the session keys generated by OBU_i and S_j. The subsequent definitions are string concatenation operations, XOR operations, hash functions, and fuzzy extractor functions. Next is to use some queries to verify the security requirements. The entire definition is shown in Figure 6.

(2) The process of V_i is shown in Figure 7.

(3) The process of RSU_j is shown in Figure 8.

(4) The process of CS is shown in Figure 9.

(5) In Figure 10, we show the results of the verification. We use VehicleStarted(), VehicleAuthed(), ServerAcVehicle(), ServerAcRSU(), RSUAcServer(), and VehicleAcRSU() to declare the beginning and the end of the agreement and whether the mutual authentication between the vehicle user, RSU, and CS is correct. The verification result shows that the session key we established has withstood the attack, and the mutual authentication is successful and correct. The protocol proposed in this chapter has passed the security verification of ProVerif.

5. Security and Performance Comparisons

This section will analyze the performance of the proposed protocol and verify the performance of the protocol by comparing its security, computing consumption, and communication consumption among similar protocols.

5.1. Security Comparisons. In this section, we compare the security of the proposed protocol with Ma et al.'s scheme [30], Jia et al.'s scheme [31], Eftekhari et al.'s scheme [38], and Liu et al.'s scheme [54]. The details are shown in Table 2. According to the informal security analysis above, it can be seen that the current common network attacks mainly include $A1$: mutual authentication; $A2$: man-in-the-middle attacks, $A3$: replay attacks, $A4$: known session-specific temporary information attacks, $A5$: perfect forward secrecy, $A6$: internal attacks, $A7$: user anonymity, $A8$: three-factor secrecy, $A9$: no key control, and $A10$: impersonation attacks. Yes means that it can resist this attack or has this security feature.

5.2. Performance Comparisons. In the performance analysis of the AKE protocol, the computation cost is an important part to be considered. In the VANETs

```
(*   channel*)
free ch :channel. (*  public channel *)
free sch: channel [private]. (*  secure channel, used for registering *)

(*  shared keys *)
free SKv : bitstring  [private].
free SKr : bitstring  [private].
free VIDi : bitstring  [private].

(* constants  *)
free  x:bitstring [private].
free  P:bitstring.
free  B:bitstring.
free  yj:bitstring.

(*  functions  &  reductions & equations  *)
fun h(bitstring) :bitstring. (*  hash function *)
fun mult(bitstring,bitstring) :bitstring. (* scalar multiplication operation *)
fun add(bitstring,bitstring):bitstring. (* Addition operation *)
fun sub(bitstring,bitstring):bitstring. (* Subtraction operation *)
fun mod(bitstring,bitstring):bitstring. (* modulus operation *)

fun con(bitstring,bitstring):bitstring. (* concatenation operation *)
reduc forall m:bitstring, n:bitstring; getmess(con(m,n))=m.

fun xor(bitstring,bitstring):bitstring. (*  XOR operation *)
equation forall m:bitstring, n:bitstring; xor(xor(m,n),n)=m.

fun Gen(bitstring):bitstring. (* Generator operation *)
fun Rep(bitstring,bitstring):bitstring.

(*  queries  *)
query attacker(SKv).
query attacker(SKr).
query attacker(VIDi).
query inj-event(VehicleAuthed()) ==> inj-event(VehicleStarted()).
query inj-event(ServerAcRSU()) ==> inj-event(ServerAcVehicle()).
query inj-event(RSUAcServer()) ==> inj-event(ServerAcRSU()).
query inj-event(VehicleAcRSU()) ==> inj-event(RSUAcServer()).

(* event *)
event VehicleStarted().
event VehicleAuthed().
event ServerAcVehicle().
event ServerAcRSU().
event RSUAcServer().
event VehicleAcRSU().
```

Figure 6: Predefinition code.

environment, due to the mobility of vehicles, the required computational time needs to be less, which reduces the time required for key establishment and makes the proposed protocol more practical. The experimental environment we used here is shown in Table 3 to test the time-consuming performance of different encryption and decryption algorithms. The algorithm was run 30 times on the device to find the average value. The results are shown in Table 4. We found that the time of the fuzzy extraction function is similar to that of the hash function during the experiment, so we use the fuzzy extraction function as a hash function.

Compared with other phases, in order to ensure the security of the session key, the authentication phase will be executed multiple times, so the calculation cost in this section only considers the calculation performed in the authentication phase. The comparison is shown in Table 5. Substitute the execution data in Table 4 to get the computation cost histogram in Figure 11.

Next, we analyze the communication consumption of the proposed protocol and compare it with related protocols. We use the number of bits specified in [11]. For example, the point of the ECC is 320 bits, the hash function is set to 256 bits, the length of the identity information is 64 bits, and the length of the random number and timestamp is 32 bits.

The protocol we propose has four transmission rounds in the authentication phase, and the transmitted information is $\{R_j, N_i, C_1, C_2, T_1, R_j, C_3, C_4, T_2, C_5, C_6C_7, N_s, T_3, C_8, T_4\}$. It contains 4 ECC points, 8 hash function outputs, and 4 timestamps' information. That is, a total of 3456 bits of information are transmitted.

The protocol of Liu et al. transmits 4 rounds, and the transmitted information is $\{\mathrm{AID}_i, A_i, TS_i, r_iP, p_{\mathrm{pub}}\}$, $\{C, \mathrm{AID}_i, \mathrm{AID}_j, \mathrm{MAC}, \mathrm{TS}_r, P_r\}$, and $\{\mathrm{AID}_m, \mathrm{TD}_m, E_{x_m}\ (q^{r_m r_t}), \mathrm{TS}_t, \sigma, M_m\}$, including 3 hash outputs, 4 identification information, 4 timestamps' information, 3 ECC points, and 3 symmetric encryptions' information (calculated according

```
(*  ------------------Vehicle's process-----------------  *)
let ProcessVehicle=
    new VIDi : bitstring; (* the Vehicle's ID *)
    out(sch,(VIDi));
    in(sch,(xPVIDi:bitstring,xKv:bitstring));
    new PWi : bitstring; (* the Vehicle's password *)
    new Bioi : bitstring; (* the Vehicle's biometric *)
    let (a: bitstring, b: bitstring)=Gen(Bioi) in
    let HPi=xor(xKv,h(con(PWi,a))) in
    let Authi=h(con(xKv,VIDi)) in
    !
    (
    event VehicleStarted();
    let a'=Rep(Bioi,b) in
    let Kv'=xor(HPi,h(con(PWi,a'))) in
    let Authi'=h(con(Kv',VIDi)) in
    if Authi'=Authi   then
    in(ch,(xRequest:bitstring,xRj:bitstring));
    new ai:bitstring;
    new xPRIDj:bitstring;
    new T1:bitstring;
    let Ni=mult(ai,P) in
    let Mi=mult(ai,B) in
    let C1=add(xPVIDi,h(con(con(con(Mi,xRj),Ni),T1))) in
    let C2=h(con(con(con(xPVIDi,VIDi),Ni),T1)) in
    out(ch,(Ni,C1,C2,T1)); (*--------authentication--------- *)
    event  VehicleAuthed();
    in(ch,(xC6:bitstring,xC8:bitstring,xT3:bitstring,xT4:bitstring));
    let PRIDj'=sub(xC6,h(con(con(VIDi,Mi),xT3))) in
    let C8'=h(con(con(PRIDj',xRj),Mi)) in
    if C8'=xC8    then   event VehicleAcRSU();
    let PVIDinew=h(con(xPVIDi,Mi)) in
    let SKv=h(con(con(PVIDinew,PRIDj'),mult(ai,xRj))) in
    0
    ).
```

Figure 7: The process of V_i.

```
(* ------RSU's process------- *)
let ProcessRSU=
    new cj:bitstring;
    new RIDj:bitstring;
    let dj=mult(cj,P) in
    out(sch,(RIDj,dj));
    in(sch,(yPRIDj:bitstring,yyj:bitstring,yzj:bitstring));
    let xj=add(yzj,cj) in
    let xj1=mult(xj,P) in
    let xj2=add(mult(yzj,P),mult(cj,P)) in
    if xj1=xj2    then
    !
    (
    new Request:bitstring;
    new rj:bitstring;
    new T2:bitstring;
    new T4:bitstring;
    let Rj=mult(rj,P) in
    out(sch,(Request,Rj));
    in(ch,(yNi:bitstring,yC1:bitstring,yC2:bitstring,yT1:bitstring));
    let Mj=mult(rj,B) in
    let C3=add(yPRIDj,h(con(con(Mj,Rj),T2))) in
    let C4=h(con(con(con(yPRIDj,RIDj),Mj),T2)) in
    out (ch,(Rj,C3,C4,T2));
    in(ch,(yC5:bitstring,yC6:bitstring,yC7:bitstring,yNs:bitstring,yT3:bitstring));
    let C5'=mult(xj,yNs) in
    if C5'=yC5    then event RSUAcServer();
    new yRIDi:bitstring;
    let PVIDinew=sub(yC7,h(con(yRIDi,con(Mj,yT3)))) in
    let C8=h(con(con(yPRIDj,Rj),T4)) in
    let PRIDjnew=h(con(yPRIDj,Mj)) in
    let SKr=h(con(con(PVIDinew,yPRIDj),mult(rj,yNi))) in
    out (ch,(yC6,C8,yT3,T4));
    0
    ).
```

Figure 8: The process of RSU_j.

```
(* --------Server's process--------- *)
let VehicleReg=
    in(sch,(zVIDi:bitstring));
    new ni:bitstring;
    let Kv=h(con(zVIDi,h(con(x,ni)))) in
    let PVIDi=h(con(zVIDi,Kv)) in
    out(sch,(PVIDi,Kv));
    0.

let RSUReg=
    in(sch,(zRIDj:bitstring,zdj:bitstring));
    new kj:bitstring;
    new PRIDj:bitstring;
    let yj=add(mult(kj,P),zdj) in
    new A:bitstring;
    new p:bitstring;
    let zj=mod(add(kj,mult(add(yj,PRIDj),A)),p) in
    out(sch,(PRIDj,yj,zj));
    0.

let ServerAuth=
    in(ch,(zRj:bitstring,zC3:bitstring,zC4:bitstring,zT2:bitstring));
    in(ch,(zNi:bitstring,zC1:bitstring,zC2:bitstring,zT1:bitstring));
    new A:bitstring;
    let Mi'=mult(A,zNi) in
    let PVIDi'=sub(zC1,h(con(con(con(Mi',zRj),zNi),zT1))) in
    new zVIDi:bitstring;
    let C2'=h(con(con(con(PVIDi',VIDi),zNi),zT1)) in
    if C2'=zC2    then event ServerAcVehicle();
    let Mj'=mult(A,zRj) in
    let PRIDj'=sub(zC3,h(con(con(Mj',zRj),zT2))) in
    new RIDj:bitstring;
    let C4'=h(con(con(con(PRIDj',RIDj),Mj'),zT2)) in
    if C4'=zC4    then event ServerAcRSU();
    new bs:bitstring;
    new T3:bitstring;
    let Ns=mult(bs,P) in
    let C5=mult(bs,add(yj,mult(add(yj,PRIDj'),B))) in
    let C6=add(PRIDj',h(con(zVIDi,con(Mi',T3)))) in
    let PVIDinew=h(con(PVIDi',Mi')) in
    let PRIDjnew=h(con(PRIDj',Mj')) in
    let C7=add(PVIDinew,h(con(zVIDi,con(Mj',T3)))) in
    out(ch,(C5,C6,C7,Ns,T3));
    0.

let ProcessServer= VehicleReg  |   RSUReg  |    ServerAuth.
```

Figure 9: The process of CS.

to 128 bits). Therefore, a total of 2496 bits of information are transmitted.

The protocol of Jia et al. transmits 4 rounds, and the transmitted information is $\{A, \mathrm{PID}_i, N_i, T_u\}$, $\{A, B, \mathrm{PID}_i, \mathrm{PID}_j, N_i, L_j, T_u, T_f\}$, $\{C, \mathrm{Auth}_i, \mathrm{Auth}_j, T_c\}$, and $\{B, C, \mathrm{Auth}_i, T_c\}$. It includes 6 ECC points, 9 hash function outputs, and 5 timestamps' information. Therefore, a total of 4384 bits of information are transmitted.

The protocol of Ma et al. transmits 4 rounds, and the transmitted information is $\{\mathrm{AID}_{U_i}, T_{U_i}, R_1, \alpha, \mathrm{AID}_{FN_j}, T_{FN_i}, R_2, R_2', \beta, R_3, R_3', R_3'', T_{CS}, \gamma, \gamma', R_2\}$. It contains 7 ECC points, 3 hash function outputs, and 4 timestamps' information. A total of 3904 bits of information are transmitted.

The protocol of Eftekhari et al. transmits 4 rounds, and the transmitted information is $\{\mathrm{Rid}_{\mathrm{DR}}, X_{\mathrm{VE}}, y_{\mathrm{VE}}, h_{\mathrm{VE}}^{\mathrm{CS}}, T\}$, $\{\mathrm{RId}_{\mathrm{FS}}, \mathrm{RId}_{\mathrm{DR}}, X_{\mathrm{FS}}, y_{\mathrm{VE}}, h_{\mathrm{FS}}^{\mathrm{CS}}, T\}$, $\{m\mathrm{RId}_{\mathrm{CS}}^{\mathrm{DRnew}}, m\mathrm{RId}_{\mathrm{CS}}^{\mathrm{FSnew}}, X_{\mathrm{CS}}, h_{\mathrm{CS}}^{\mathrm{FS}}, h_{\mathrm{CS}}^{\mathrm{FS}}\}$, and $\{m\mathrm{RId}_{\mathrm{CS}}^{\mathrm{DRnew}}, X_{\mathrm{FS}}, X_{\mathrm{CS}}, T\}$. It contains 6 ECC points, 14 hash function outputs, and 2 timestamps' information. A total of 5568 bits of information are transmitted. The comparison of communication consumption is shown in Table 6. In order to see the comparison effect more clearly, we have generated Figure 12.

Combined with Tables 2, 5, and 6, we discussed the results of the performance analysis. The protocol of Eftekhari et al. has no obvious security vulnerabilities, and the computation cost is similar to that of the protocol we proposed; the main computation cost gap is on the server side. Because the server has strong computing power, it has little effect on the overall computation cost; and, from Table 6 we can see that the communication cost of Eftekhari et al.'s protocol is much higher than that of the proposed protocol. Also, the proposed protocol is similar to Jia et al.'s protocol in terms of computation cost, but Jia et al.'s protocol has security vulnerabilities. The communication cost of all the schemes participating in the comparison is slightly higher than that of the protocol of Liu et al. It can be seen from Figure 11 that the computation cost of the protocol of Liu et al. is the highest, and the security

```
-- Query not attacker(SKv[])
nounif mess(sch[],zVIDi_431)/-5000
Completing...
Starting query not attacker(SKv[])
RESULT not attacker(SKv[]) is true.
-- Query not attacker(SKr[])
nounif mess(sch[],zVIDi_1137)/-5000
Completing...
Starting query not attacker(SKr[])
RESULT not attacker(SKr[]) is true.
-- Query not attacker(VIDi[])
nounif mess(sch[],zVIDi_1825)/-5000
Completing...
Starting query not attacker(VIDi[])
RESULT not attacker(VIDi[]) is true.
-- Query inj-event(VehicleAuthed) ==> inj-event(VehicleStarted)
nounif mess(sch[],zVIDi_2513)/-5000
Completing...
Starting query inj-event(VehicleAuthed) ==> inj-event(VehicleStarted)
RESULT inj-event(VehicleAuthed) ==> inj-event(VehicleStarted) is true.
-- Query inj-event(ServerAcRSU) ==> inj-event(ServerAcVehicle)
nounif mess(sch[],zVIDi_3202)/-5000
Completing...
Starting query inj-event(ServerAcRSU) ==> inj-event(ServerAcVehicle)
RESULT inj-event(ServerAcRSU) ==> inj-event(ServerAcVehicle) is true.
-- Query inj-event(RSUAcServer) ==> inj-event(ServerAcRSU)
nounif mess(sch[],zVIDi_3892)/-5000
Completing...
Starting query inj-event(RSUAcServer) ==> inj-event(ServerAcRSU)
RESULT inj-event(RSUAcServer) ==> inj-event(ServerAcRSU) is true.
-- Query inj-event(VehicleAcRSU) ==> inj-event(RSUAcServer)
nounif mess(sch[],zVIDi_4582)/-5000
Completing...
Starting query inj-event(VehicleAcRSU) ==> inj-event(RSUAcServer)
RESULT inj-event(VehicleAcRSU) ==> inj-event(RSUAcServer) is true.
```

Figure 10: Verification result.

Table 2: Security comparison.

Attack methods	Liu et al.	Ma et al.	Jia et al.	Eftekhari et al.	Our scheme
*A*1	Yes	Yes	No	Yes	Yes
*A*2	Yes	Yes	—	—	Yes
*A*3	—	Yes	Yes	Yes	Yes
*A*4	No	No	No	Yes	Yes
*A*5	No	Yes	Yes	Yes	Yes
*A*6	No	No	Yes	Yes	Yes
*A*7	No	No	Yes	Yes	Yes
*A*8	—	No	Yes	—	Yes
*A*9	Yes	Yes	Yes	Yes	Yes
*A*10	Yes	Yes	Yes	Yes	Yes

Table 3: Experimental environment.

Denomination	Description
Hardware equipment	Laptop
Processor	AMD Ryzen 5 4600H
Running memory	16 GB
System	Windows 10
Software	IntelliJ IDEA 2019.3
Cryptography library	JPBC-2.0.0

performance is very poor. The computation cost and communication cost of Ma et al.'s protocol are relatively average, but both are slightly higher than those of our proposed protocol, and their protocol is vulnerable to known session-specific temporary information attacks and internal attacks and cannot guarantee user anonymity. In general, it is more reasonable for the proposed protocol to combine security, computation cost, and communication cost analysis.

Table 4: The computational cost of complex operations.

Operation	Definition	Execution time (ms)
T_{pm}	Elliptic curve scalar point multiplication	8.8
T_{pa}	Elliptic curve scalar point addition	0.057
T_h	Hash function	0.0058
T_{bp}	Bilinear pairing	11.43
T_{h2p}	String to point hash operation	26.1
T_{se}	Symmetric encryption	18.37

Table 5: Computation cost comparison.

Scheme	V_i	RSU_j	CS	Total
Liu et al.	$T_{pm} + 2T_{h2p} + T_{bp} + T_{se}$	$T_{pm} + 2T_{h2p} + T_{bp} + 2T_{se}$	$T_{pm} + 2T_{h2p} + T_{se}$	$3T_{pm} + 6T_{h2p} + 2T_{bp} + 4T_{se}$
Jia et al.	$2T_{pm} + 5T_h + T_{bp}$	$2T_{pm} + 4T_h + T_{bp}$	$3T_{pm} + 11T_h + T_{bp}$	$7T_{pm} + 21T_h + 3T_{bp}$
Ma et al.	$3T_{pm} + 3T_h$	$4T_{pm} + 3T_h$	$6T_{pm} + 9T_h$	$13T_m + 15T_s$
Eftekhari et al.	$3T_{pm} + 13T_h + T_{pa}$	$2T_{pm} + 15T_h + T_{pa}$	$3T_{pm} + 17T_h + 2T_{pa}$	$8T_{pm} + 45T_h + 4T_{pa}$
Ours	$2T_{pm} + 5T_h$	$3T_{pm} + 5T_h$	$4T_{pm} + 8T_h$	$9T_{pm} + 18T_h$

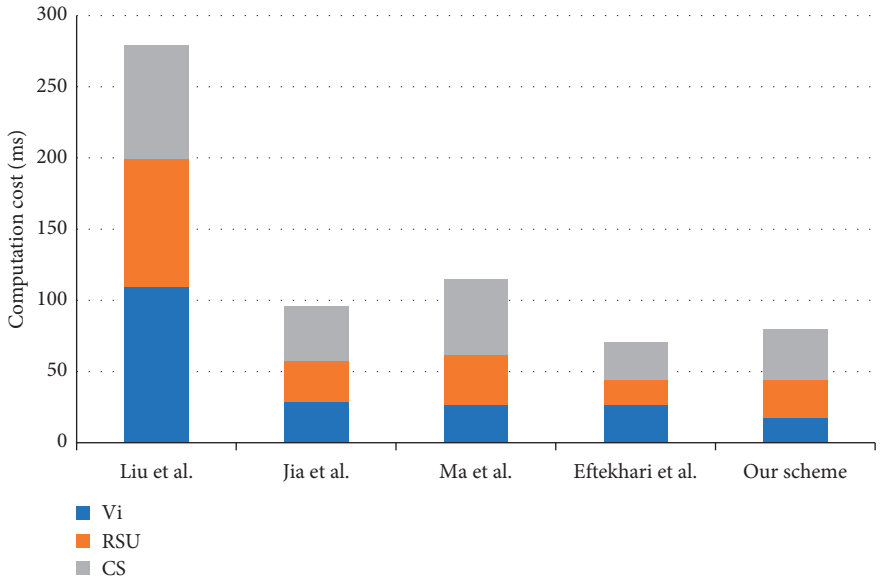


Figure 11: Computation cost histogram.

Table 6: Comparison of communication and massage rounds.

Scheme	Communication cost (bits)	Massage rounds
Liu et al.	2496	4
Jia et al.	4384	4
Ma et al.	3904	4
Eftekhari et al.	5568	4
Our scheme	3456	4

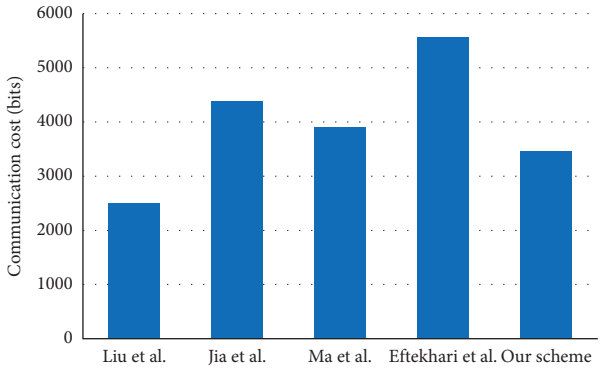


Figure 12: Comparison of communication cost.

6. Conclusion

Based on ECC, this paper designs a new provably safe AKE scheme before transmitting road condition information. We first reviewed the research status of AKE protocol in the VANET environment and found that it is necessary to propose a scheme to protect vehicle data in the information reading phase. We conducted an informal security analysis of the proposed protocol from mutual authentication, anonymity, perfect forward secrecy, man-in-the-middle attacks, internal attacks, and so forth and passed strict formal security analyses, such as the ROR model and ProVerif security verification tools, indicating that the proposed protocol is secure. Through the comparison of security and performance, the proposed protocol is secure, more effective, and more reasonable than the existing protocol. The application of authentication and key exchange in the VANETs environment is the general trend of the development of the VANETs. With the continuous development of the VANETs, subsequent application scenarios are also diverse, such as social Internet of Vehicles, which involve more user privacy information, and this topic will have great research value and research space in the future. Therefore, the communication security of the VANETs environment must also be a key research topic for scholars.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] Z. Meng, J.-S. Pan, and K.-K. Tseng, "PaDE: an enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80–99, 2019.

[2] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, "An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems," *Information Sciences*, vol. 561, pp. 304–325, 2021.

[3] X. Xue, X. Wu, C. Jiang, G. Mao, and H. Zhu, "Integrating sensor ontologies with global and local alignment extractions," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6625184, 10 pages, 2021.

[4] X. Xue, C. Yang, C. Jiang, P. W. Tsai, G. Mao, and H. Zhu, "Optimizing ontology alignment through linkage learning on entity correspondences," *Complexity*, vol. 2021, Article ID 5574732, 12 pages, 2021.

[5] H. Xiong, Y. Wu, C. Jin, and S. Kumari, "Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIOT," *IEEE Internet of Things Journal*, vol. 7, no. 12, Article ID 11713, 2020.

[6] H. Xiong, Y. Zhao, Y. Hou et al., "Heterogeneous signcryption with equality test for IIoT environment," *IEEE Internet of Things Journal*, 2020.

[7] J. S. Pan, X. X. Sun, S. C. Chu, A. Abraham, and B. Yan, "Digital watermarking with improved SMS applied for QR code," *Engineering Applications of Artificial Intelligence*, vol. 97, Article ID 104049, 2021.

[8] J. M.-T. Wu, G. Srivastava, A. Jolfaei, P. Fournier-Viger, and J. C.-W. Lin, "Hiding sensitive information in eHealth datasets," *Future Generation Computer Systems*, vol. 117, pp. 169–180, 2021.

[9] J. M. T. Wu, G. Srivastava, U. Yun, S. Tayeb, and J. C. W. Lin, "An evolutionary computation-based privacy-preserving data mining model under a multithreshold constraint," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 3, Article ID e4209, 2021.

[10] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[11] T. Y. Wu, Y. Q. Lee, C. M. Chen, Y. Tian, and N. A. Al-Nabhan, "An enhanced pairing-based authentication scheme for smart grid communications," *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[12] J. C. Hsu, Y. S. Jheng, S. M. Mizanur Rahman, and R. Tso, "Password-based authenticated key exchange from lattices for client server model," *Journal of Computer Security and Data Forensics*, vol. 1, no. 1, pp. 1–17, 2021.

[13] L. H. Li-Hua Li, L. C. Luon-Chang Lin, and M. S. Min-Shiang Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.

[14] C. C. Chang and J. S. Lee, "An efficient and secure multi-server password authentication scheme using smart cards," in *Proceedings of the 2004 International Conference on Cyberworlds*, pp. 417–422, Tokyo, Japan, November 2004.

[15] Y. Wang, Y. Liu, H. Ma, Q. Ma, and Q. Ding, "The research of identity authentication based on multiple biometrics fusion in complex interactive environment," *Journal of Network Intelligence*, vol. 4, no. 4, pp. 124–139, 2019.

[16] T. Y. Wu, Z. Lee, L. Yang, J. N. Luo, and R. Tso, "Provably secure authentication key exchange scheme using fog nodes in vehicular ad hoc networks," *The Journal of Supercomputing*, vol. 77, pp. 6992–7020, 2021.

[17] A. Irshad, H. F. Ahmad, B. A. Alzahrani, M. Sher, and S. A. Chaudhry, "An efficient and anonymous Chaotic Map based authenticated key agreement for multi-server architecture," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 12, pp. 5572–5595, 2016.

[18] F. Wu, X. Li, L. Xu, A. K. Sangaiah, and J. J. P. C. Rodrigues, "Authentication protocol for distributed cloud computing: an explanation of the security situations for Internet-of-Things-

enabled devices," *IEEE Consumer Electronics Magazine*, vol. 7, no. 6, pp. 38–44, 2018.

[19] T.-Y. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, and C.-M. Chen, "An authenticated key exchange protocol for multi-server architecture in 5G networks," *IEEE Access*, vol. 8, Article ID 28096, 2020.

[20] T.-T. Truong, M.-T. Tran, A.-D. Duong, and I. Echizen, "Provable identity based user authentication scheme on ECC in multi-server environment," *Wireless Personal Communications*, vol. 95, no. 3, pp. 2785–2801, 2017.

[21] K.-H. Yeh, "A provably secure multi-server based authentication scheme," *Wireless Personal Communications*, vol. 79, no. 3, pp. 1621–1634, 2014.

[22] Y. Zhao, S. Li, and L. Jiang, "Secure and efficient user authentication scheme based on password and smart card for multi-server environment," *Security and Communication Networks*, vol. 2018, Article ID 9178941, 13 pages, 2018.

[23] M. Hassan, A. Sultan, A. A. Awan, S. Tahir, and I. Ihsan, "An enhanced and secure multiserver-based user authentication protocol," in *Proceedings of the International Conference on Cyber Warfare and Security (ICCWS)*, pp. 1–6, Islamabad, Pakistan, October 2020.

[24] M. Eltoweissy, S. Olariu, and M. Younis, "Towards autonomous vehicular clouds," *Ad Hoc Networks*, vol. 49, pp. 1–16, 2010.

[25] S. Bitam, A. Mellouk, and S. Zeadally, "VANET-cloud: a generic cloud computing model for vehicular Ad Hoc networks," *IEEE Wireless Communications*, vol. 22, no. 1, pp. 96–102, 2015.

[26] Q. Jiang, J. Ni, J. Ma, L. Yang, and X. Shen, "Integrated authentication and key agreement framework for vehicular cloud computing," *IEEE Network*, vol. 32, no. 3, pp. 28–35, 2018.

[27] S. K. Bhoi, S. K. Panda, S. R. Ray et al., "TSP-HVC: a novel task scheduling policy for heterogeneous vehicular cloud environment," *International Journal of Information Technology*, vol. 11, no. 4, pp. 853–858, 2019.

[28] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for Internet of vehicles," *IEEE Access*, vol. 7, Article ID 12047, 2019.

[29] J. Zhang, H. Zhong, J. Cui, and Y. Xu, "SMAKA: Secure many-to-many authentication and key agreement scheme for vehicular networks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1810–1824, 2020.

[30] M. Ma, D. He, H. Wang, N. Kumar, and K.-K. R. Choo, "An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8065–8075, 2019.

[31] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Networks*, vol. 25, no. 8, pp. 4737–4750, 2019.

[32] C. M. Chen, Y. Huang, K. H. Wang, and S. Kumari, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Information Systems*, 2020.

[33] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[34] C. Zhang, R. Lu, X. Lin, and P. H. Ho, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proceedings of The 27th Conference on Computer Communications*, pp. 246–250, Phoenix, AZ, USA, April 2008.

[35] M. C. Ming-Chin Chuang and J. F. Jeng-Farn Lee, "TEAM: trust-extended authentication mechanism for vehicular ad hoc networks," *IEEE systems journal*, vol. 8, no. 3, pp. 749–758, 2014.

[36] Y. Zhou, X. Zhao, Y. Jiang, F. Shang, S. Deng, and X. Wang, "An enhanced privacy-preserving authentication scheme for vehicle sensor networks," *Sensors*, vol. 17, no. 12, p. 2854, 2017.

[37] L. Wu, Q. Sun, X. Wang et al., "An efficient privacy-preserving mutual authentication scheme for secure V2V communication in vehicular ad hoc network," *IEEE Access*, vol. 7, Article ID 55050, 2019.

[38] S. A. Eftekhari, M. Nikooghadam, and M. Rafighi, "Security-enhanced three-party pairwise secret key agreement protocol for fog-based vehicular ad-hoc communications," *Vehicular Communications*, vol. 28, Article ID 100306, 2020.

[39] P. Mohit, R. Amin, and G. P. Biswas, "Design of authentication protocol for wireless sensor network-based smart vehicular system," *Vehicular Communications*, vol. 9, pp. 64–71, 2017.

[40] S. Yu, J. Lee, K. Lee, K. Park, and Y. Park, "Secure authentication protocol for wireless sensor networks in vehicular communications," *Sensors*, vol. 18, no. 10, p. 3191, 2018.

[41] M. J. Sadri and M. Rajabzadeh Asaar, "A lightweight anonymous two-factor authentication protocol for wireless sensor networks in Internet of Vehicles," *International Journal of Communication Systems*, vol. 33, no. 14, Article ID e4511, 2020.

[42] P. Wang, C.-M. Chen, S. Kumari et al., "HDMA: hybrid D2D message authentication scheme for 5G-enabled VANETs," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[43] H. Zhu, X. Wang, C. M. Chen, and S. Kumari, "Two novel semi-quantum-reflection protocols applied in connected vehicle systems with blockchain," *Computers & Electrical Engineering*, vol. 86, Article ID 106714, 2020.

[44] M. A. Khan, I. Ullah, N. Kumar et al., "An efficient and secure certificate-based access control and key agreement scheme for flying ad-hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4839–4851, 2021.

[45] M. A. Khan, I. Ullah, S. Nisar et al., "An efficient and provably secure certificateless key-encapsulated signcryption scheme for flying ad-hoc network," *IEEE Access*, vol. 8, Article ID 36807, 2020.

[46] M. A. Khan, I. M. Qureshi, I. Ullah, S. Khan, and F. Khanzada, "An efficient and provably secure certificateless blind signature scheme for flying ad-hoc network based on multi-access edge computing," *Electronics*, vol. 9, no. 1, p. 30, 2020.

[47] M. A. Khan, I. Ullah, S. Nisar et al., "Multiaccess edge computing empowered flying ad hoc networks with secure deployment using identity-based generalized signcryption," *Mobile Information Systems*, vol. 2020, Article ID 8861947, 15 pages, 2020.

[48] T. Y. Wu, L. Yang, Z. Lee, S. C. Chu, S. Kumari, and S. Kumar, "A provably secure three-factor Authentication protocol for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5537018, 15 pages, 2021.

[49] T. Y. Wu, T. Wang, Y. Q. Lee, W. Zheng, S. Kumari, and S. Kumar, "Improved authenticated key agreement scheme for fog-driven IoT healthcare system," *Security and Communication Networks*, vol. 2021, Article ID 6658041, 16 pages, 2021.

[50] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," *Advances in Cryptology*, Springer, Berlin, Germany, pp. 337–351, 2002.

[51] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," *Journal of the ACM*, vol. 51, no. 4, pp. 557–594, 2004.

[52] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[53] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules," in *Proceedings 14th IEEE Computer Security Foundations Workshop*, pp. 82–96, Cape Breton, NS, Canada, June 2001.

[54] Y. Liu, Y. Wang, and G. Chang, "Efficient privacy-preserving dual authentication and key agreement scheme for secure V2V communications in an IoV paradigm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2740–2749, 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5554318, 9 pages
https://doi.org/10.1155/2021/5554318

Research Article

Designing an Efficient and Secure Message Exchange Protocol for Internet of Vehicles

Shehzad Ashraf Chaudhry

Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey

Correspondence should be addressed to Shehzad Ashraf Chaudhry; sashraf@gelisim.edu.tr

Received 28 February 2021; Accepted 8 May 2021; Published 19 May 2021

Academic Editor: Prosanta Gope

In the advancements in computation and communication technologies and increasing number of vehicles, the concept of Internet of Vehicles (IoV) has emerged as an integral part of daily life, and it can be used to acquire vehicle related information including road congestion, road description, vehicle location, and speed. Such information is very vital and can benefit in a variety of ways, including route selection. However, without proper security measures, the information transmission among entities of IoV can be exposed and used for wicked intentions. Recently, many authentication schemes were proposed, but most of those authentication schemes are prone to insecurities or suffer from heavy communication and computation costs. Therefore, a secure message authentication protocol is proposed in this study for information exchange among entities of IoV (SMEP-IoV). Based on secure symmetric lightweight hash functions and encryption operations, the proposed SMEP-IoV meets IoV security and performance requirements. For formal security analysis of the proposed SMEP-IoV, BAN logic is used. The performance comparisons show that the SMEP-IoV is lightweight and completes the authentication process in just 0.198 ms.

1. Introduction

The Internet of Vehicles (IoV) is a self-organized network of vehicles on the road and the road side units (RSUs). The IoV provides intervehicles (V2V) and vehicles to RSUs (V2R) communication infrastructure [1], which can benefit in many ways including the information relating to road congestion/traffic issues, parking information, alternative routes, and warnings of potential accidents. Using the information, the drivers can quickly make decisions relating to vehicles and/or road/s. It can further help the unmanned vehicles regarding the accuracy and safety through the use of more sophisticated information and artificial intelligence techniques. The information exchanged or the communication among entities of IoV is always through public wireless channel, which makes it prone to several attacks. An attacker can easily listen and extract the meaningful information from the exchanged messages. Such information can be very crucial for the accuracy and safety of the vehicles in an IoV. The attacker can replay an old message or can inject a message with total fake information, and it can cause some severe consequences on the vehicles and the riders including the accidents. Moreover, the listened information can be used by an attacker to trace/track a vehicle/rider, and such information can be used for criminal/terrorism purposes. The information can also be faked for marketing purposes to gain attraction of the riders, while they are attracted to a specific route through false information of traffic as well as to compete for the parking lots [2].

Therefore, the security and privacy of the entire IoV including the communicating entities have more importance than all other factors. The goal can be achieved through authentication of the entities including vehicles before initiation of the communication among the entities of IoV. In this study, we proposed a lightweight symmetric key-based authentication scheme to secure message exchange among the entities of the IoV. We organize rest of the study as follows: Table 1 provides the notation guide. In Subsection

Table 1: Notations guide.

Symbols	Representations
VS, V_i	Vehicle server, vehicle
RSU_j	Road side unit
ID_{vi}, ID_{rj}	ID's of V_i and RSU_j
K_{VS}	Master secret key of VS
K_{rj}	Shared key among RSU_j and V_i
t_x, r_s	Timestamp and random number of entity x
PID_{vi}	Pseudoidentity of V_i
$H(a), \|$	Hash of a and concatenation

1.1, the system model is described. The motivations and contributions of the study are explained in Subsection 1.2, while the Subsection 1.3 discusses the adopted adversarial model. The Section 2 summarizes the existing related literature; whereas, our proposed secure message exchange protocol for IoV (SMEP-IoV) scheme is explained in Section 3. Using BAN logic, the Section 4 formally proves the security of the SMEP-IoV. In Section 5, a discussion on functional security and attack resilience of the proposed SMEP-IoV is given. Security and performance comparisons of the proposed SMEP-IoV with related schemes are given in Section 6. The study is concluded finally in Section 7.

1.1. System Model. Figure 1 shows a typical IoV scenario. It consists of vehicles, each having installed a processing unit called on-board unit, which is responsible for communication and processing of exchanged data among the vehicle and other entities of an IoV. Along with vehicles, there are road side units (RSUs), which are the infrastructure deployed on the road. Typically, communication is performed among vehicles and nearby RSU. Moreover, inter-vehicle communication is also an important component of the IoV. The whole network is administered by a trusted authority called vehicle server (VS). All the vehicles and related entities (RSUs) join the IoV by registering with VS. After getting registered with VS, the two entities can communicate with each other, for which both have to authenticate each other, and the authentication ensures that both communicating entities are legitimate.

1.2. Motivation and Contributions. Recently, many authentication schemes are proposed to secure message exchange among the entities of an IoV. However, many authentication schemes for IoV lack the required security features and resistance to known attacks. In this connection, Yu et al. proved some of the weaknesses of the scheme of Vasudev et al. Yu et al. further claimed to propose a secure authentication scheme with all required security features. The arguments in preceding section of this study refute their claim and the proof relating to several insecurities of Yu et al.'s scheme calls for an authentication scheme with all required security features.

The contributions of this study are many folds:

(i) Initially, we unveiled that the insecurities of the IoV authentication scheme proposed by Yu et al. We then proposed a robust authentication scheme using symmetric key-based encryption and hash functions.

(ii) The security of the proposed scheme is proved using formal RoR.

(iii) The comparative study with respect to efficiency and security among proposed and several existing studies is also provided in this study.

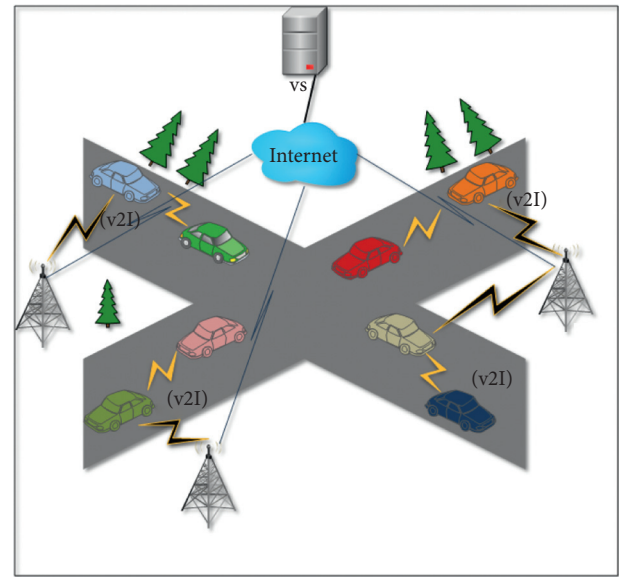


Figure 1: Typical IoV scenario.

1.3. Attack Model. We have taken into consideration the eCK adversary model [3], with strong adversary as compared with DY [4] and CK [5] models. The eCK is an extension of the CK model with a more strong adversary having capabilities to launch a key compromise impersonation attack in addition to controlling the communication channel, launching the power analysis to extract secrets stored in the smart card and access to all public parameters [6,7].

2. Related Literature

In their survey, Contreras-Castillo et al. [8] pointed out some security requirements and suggested to address authentication, integrity, confidentiality, and related security requirement before the IoV gain popularity. Some future directions were also discussed in [8]. In addition to the mentioned security requirements in [8], Mokhtar and Azab [9] stressed vehicle privacy, untraceability, access control, and resistance against tempering/forgery and jamming attacks.

In recent times, some authentication schemes were proposed [10–13]. Two different schemes were proposed by Lin et al. [14] and Yin et al. [15] using hashchains. Both schemes provided efficient and rapid authentication but lacked vehicle/user anonymity. The absence of anonymity could lead towards the leakage of sensitive vehicle/user

information, IoV. In 2015, the scheme of Li et al. [16] was proved to have weaknesses against disclosure of session key attack by Dua et al. [17]. Afterwards in 2016, Wang et al. also proposed a smartcard-based two-factor authentication scheme for IoV [18], which was proved as having weaknesses against many attacks including vehicle/user forgery and smart card stolen attacks, and the scheme was also lacking anonymity by Amin et al. [19]. A pairing-based scheme was also proposed by Liu et al. [20]. However, due to usage of expensive pairing operations, a considerable delay can happen, which is unsuitable for fast moving vehicles. Another lightweight scheme was proposed by Ying et al. [21]. However, Chen et al. [2] found critical weaknesses in the scheme of Ying et al. Due to usage of modular exponentiation, the scheme of Chen et al. entails inefficiencies against storage, communication cost, and computation time. Quite recently, in 2020, Vasudev et al. [22] presented another efficient authentication scheme. In 2020, Yu et al. [23] pointed out that the scheme of Vasudev et al. lacks mutual authentication and has weaknesses against some attacks including session key disclosure and vehicle/user forgery attacks. Yu et al. also proposed an improved scheme. However, the scheme of Yu et al. is prone to many attacks including disclosure of master secret key K_{VS} of the vehicle server. Due to leakage of K_{VS}, the scheme of Yu et al. cannot be deployed in any environment because if an attacker is able to get K_{VS}, it can generate secret parameters for any of the existing device to impersonate on behalf of that device; moreover, the attacker can register and deploy fake vehicles in the system. Any registered device can compute K_{VS} using the Q_i stored in the smartcard and its own password and identity related parameters, i.e., RPW_i and RID_i. For this, a vehicle/user V_i computes enters password (PW_{Ai}), identity (ID_{Ai}), and computes $\mathrm{RID}_i = h(\mathrm{ID}_{Ai} \| \mathrm{PW}_{Ai})$, $\mathrm{RN}_i = E_i \oplus h(\mathrm{PW}_{Ai} \| \mathrm{RID}_i)$ and $\mathrm{RPW}_i = h(\mathrm{PW}_{Ai} \| \mathrm{RN}_i)$. The V_i now computes $K_{\mathrm{VS}} = Q_i \oplus h(\mathrm{RID}_i \| \mathrm{RPW}_i)$. Here, K_{VS} is the master secret key of the vehicle server. Now, using K_{VS}, any dishonest vehicle of the system can launch any attack on any devices. For example,

(i) The dishonest vehicle with extracted K_{VS} can disclose any session key shared among two vehicles. Let V_x initiates a login request by sending $\{M_{i1}, M_{i2}, M_{\mathrm{AE}}, T_1\}$. By just listening to the request, the dishonest vehicle using M_{i1}, M_{i2}, and T_1 can compute $R_1 = M_{i1} \oplus h(K_{\mathrm{VS}} \| T_1)$ and $M_{\mathrm{request1}} = M_{i2} \oplus h(R_1 \| K_{\mathrm{VS}})$, on the fly. Similarly, when the responding vehicle V_y sends reply message $\{M_3, M_{\mathrm{EA}}, T_2\}$, the dishonest vehicle using M_3 and T_2 can compute $(M_{\mathrm{request2}} \| R_2)$ and the session key $\mathrm{SK} = h(R_1 \| R_2 \| K_{\mathrm{VS}})$ by just executing instep two hash functions on the public parameters.

(ii) Likewise, the dishonest device can launch man in middle, impersonation, and all related attacks using K_{VS}. For example, when V_x sends request message $\{M_{i1}, M_{i2}, M_{\mathrm{AE}}, T_1\}$ to V_y, the dishonest vehicle can extract R_1 and then can generate another valid response message $\{M_3, M_{\mathrm{AE}}, T_2\}$ by using K_{SV} and current timestamp. Ultimately, the possession of K_{SV} enables a dishonest vehicle to generate a valid request and a response message, and it can act like a man in middle.

3. Proposed SMEP-IoV

The proposed secure message exchange protocol for IoV (SMEP-IoV) consists of four phases. Table 1 provides the notation guide to understand the technical details of the proposed SMEP-IoV, briefed in following subsections:

3.1. SMEP-IoV: Initialization. The vehicle server (VS) selects its secret key K_{vs}, a one way hash function $H: \{0,1\}^* \longrightarrow \{0,1\}^l$ and a symmetric encryption/decryption function $X = E_k(Y)$.

3.2. SMEP-IoV: RSU Registration. During this phase, VS registers all road side units by assigning a unique identity ID_{rj} and a shared secret key $K_{rj} = h(\mathrm{ID}_{rj} \| K_{\mathrm{VS}})$. The VS stores ID_{rj} in its database.

3.3. SMEP-IoV: Vehicle Registration. During this phase, VS registers all vehicles by assigning a unique identity ID_{vi}. Moreover, VS computes $A_{vi} = h(K_{\mathrm{VS}} \| \mathrm{ID}_{vi})$, $\mathrm{PID}_{vi} = E_{K_{\mathrm{VS}}}(\mathrm{ID}_{vi} \| r_i 0)$, and $B_{vi} = h(\mathrm{PID}_{vi} \| K_{\mathrm{VS}})$. The VS stores ID_{vi}, PID_{vi}, A_{vi}, and B_{vi} in the memory of the vehicle V_i. Furthermore, the VS stores ID_{vi} in its own memory. Please note, except ID_{vi}, the VS does not store any other parameter relating to a vehicle say V_i. Specifically, PID_{vi}, A_{vi}, and B_{vi} are not stored in the memory of VS.

3.4. SMEP-IoV: Message Authentication. For message authentication, the vehicle V_i initiates the following steps with RSU_j and vehicle server VS, the in-sequence steps as shown in Figure 2:

3.4.1. PMA 1.

$$V_i \longrightarrow \mathrm{RSU}_j\colon M_{vi} = \{\mathrm{PID}_{vi}, M_{i1}, M_{i2}, t_i\}, \tag{1}$$

where V_i initiates the message authentication process by generating fresh timestamp t_i and a random number r_i. V_i further computes $M_{i1} = h(A_{vi} \| \mathrm{ID}_{vi} \| t_i \| r_i)$ and $M_{i2} = r_i \oplus B_{vi}$. V_i finalizes these steps by sending $M_{vi} = \{\mathrm{PID}_{vi}, M_{i1}, M_{i2}, t_i\}$ to RSU_j.

3.4.2. PMA 2.

$$\mathrm{RSU}_j \longrightarrow \mathrm{VS}\colon M_{rj1} = \{\mathrm{ID}_{rj}, M_{vi}, M_{j1}, t_j\} \tag{2}$$

On receiving $M_{vi} = \{\mathrm{PID}_{vi}, M_{i1}, M_{i2}, t_i\}$, the RSU_j checks the freshness of t_i by comparing it with current timestamp; if the delay is not within a predefined tolerable range ΔT, the RSU_j terminates the process; otherwise, RSU_j generates new timestamp t_j and a random number r_j. Moreover, RSU_j computes $M_{j1} = E_{K_{rj}}(r_j, t_j)$ and sends $M_{rj1} = \{\mathrm{ID}_{rj}, M_{vi}, M_{j1}, t_j\}$ to VS.

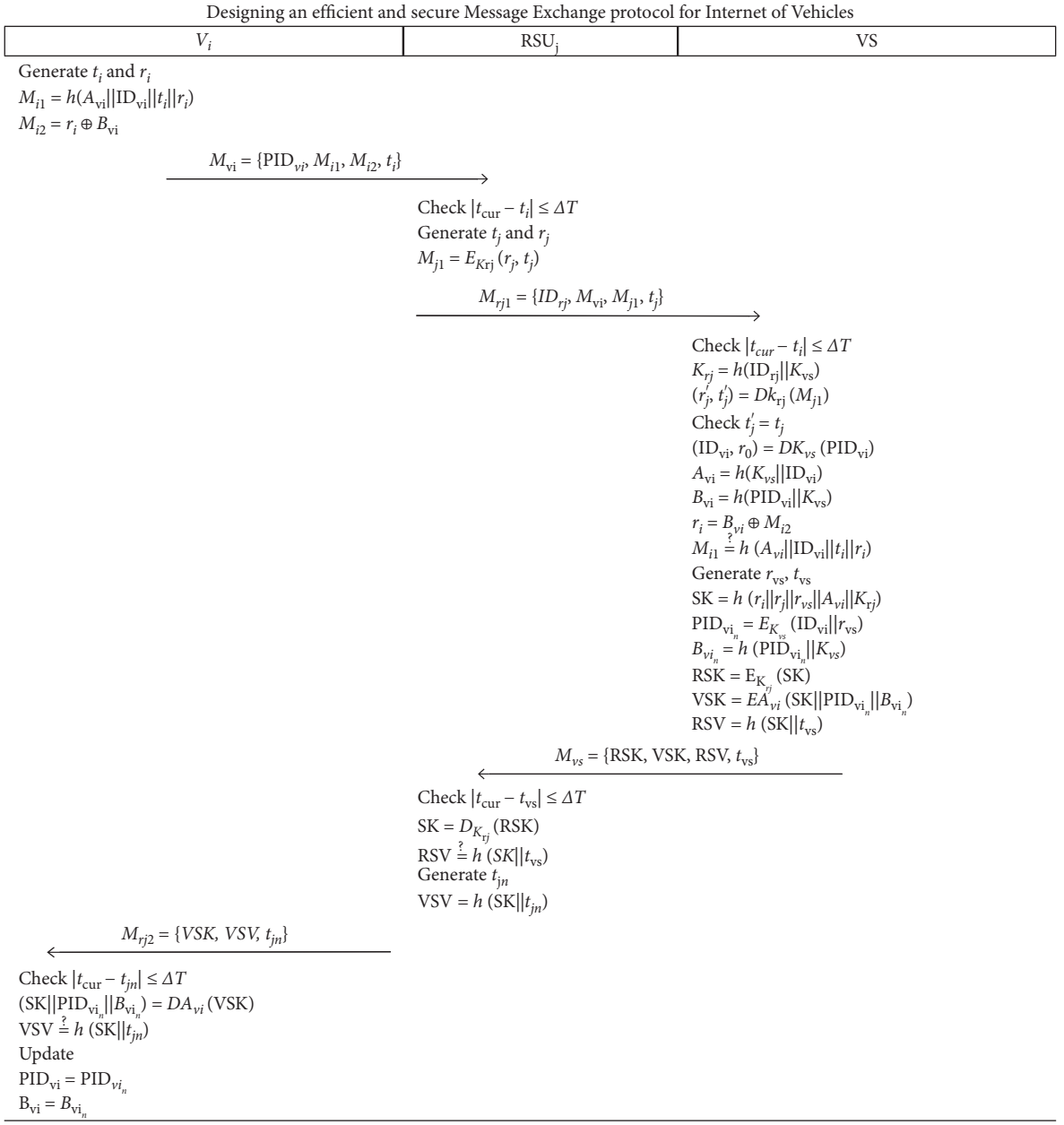


Figure 2: Proposed SMEP-IoV.

3.4.3. PMA 3.

$$VS \longrightarrow \text{RSU}_j\colon M_{vs} = \{\text{RSK}, \text{VSK}, \text{RSV}, t_{vs}\} \quad (3)$$

After receiving $M_{rj1} = \{\text{ID}_{rj}, M_{vi}, M_{j1}, t_j\}$, the VS checks the freshness of t_j by comparing it with current timestamp; if the delay is not within a predefined tolerable range ΔT, the VS terminates the process; otherwise, VS computes $K_{rj} = h(\text{ID}_{rj}\|K_{\text{VS}})$ and decrypts M_{j1} using K_{rj} to obtain the pair (r_j', t_j'). The VS also verifies the sameness of received t_j with decrypted t_j', and in case both are same, the VS using its secret key K_{vs} decrypts PID_{vi} and obtains (ID_{vi}, r_0). Now, the VS computes $A_{vi} = h(K_{\text{vs}}\|\text{ID}_{vi})$ and $B_{vi} = h(\text{PID}_{vi}\|K_{\text{vs}})$ and gets $r_i = B_{vi} \oplus M_{i2}$. After that, the VS checks $M_{i1} \stackrel{?}{=} h(A_{vi}\|\text{ID}_{vi}\|t_i\|r_i)$, and if verification is successful, the VS generates timestamp t_{vs} and a random number r_{vs}. The VS computes session key $\text{SK} = h(r_i\|r_j\|r_{vs}\|A_{vi}\|K_{rj})$, $\text{PID}_{vi_n} = E_{K_{\text{vs}}}(\text{ID}_{vi}\|r_{\text{vs}})$, $B_{vi_n} = h(\text{PID}_{vi_n}\|K_{\text{vs}})$, and $\text{RSK} = E_{K_{rj}}(\text{SK})$. In addition, the VS computes $\text{VSK} = E_{A_{vi}}(\text{SK}\|\text{PID}_{vi_n}\|B_{vi_n})$ and $\text{RSV} = h(\text{SK}\|t_{\text{vs}})$. Now, the VS sends $M_{\text{vs}} = \{\text{RSK}, \text{VSK}, \text{RSV}, t_{\text{vs}}\}$ to RSU_j.

3.4.4. PMA 4.

$$\text{RSU}_j \longrightarrow V_i\colon M_{rj2} = \{\text{VSK}, \text{VSV}, t_{j_n}\}. \quad (4)$$

After receiving $M_{\text{vs}} = \{\text{RSK}, \text{VSK}, \text{RSV}, t_{\text{vs}}\}$, the RSU_j checks the freshness of t_{vs} by comparing it with current timestamp; if the delay is not within a predefined tolerable range ΔT, the RSU_j terminates the process; otherwise, RSU_j

computes session key $SK = D_{K_{rj}}(RSK)$. Now, RSU_j checks the session key verifier $RSV \stackrel{?}{=} h(SK \| t_{vs})$, and if RSV is verified successfully, the RSU_j accepts the session key. Now, RSU_j generates new timestamp t_{j_n} and session key verifier $VSV = h(SK \| t_{j_n})$ for V_i. After that, the RSU_j sends $M_{rj2} = \{VSK, VSV, t_{j_n}\}$ to V_i.

3.4.5. PMA 5. After receiving $M_{rj2} = \{VSK, VSV, t_{j_n}\}$, the V_i checks the freshness of t_{j_n} by comparing it with current timestamp; if the delay is not within a predefined tolerable range ΔT, the V_i terminates the process; otherwise, V_i decrypts VSK using A_{vi} and obtains $(SK \| PID_{vi_n} \| B_{vi_n})$. Now, V_i checks the session key verifier $VSV \stackrel{?}{=} h(SK \| t_{j_n})$, and if VSV is verified successfully, the V_i accepts the session key and updates $PID_{vi} = PID_{vi_n}$ and $B_{vi} = B_{vi_n}$.

4. Formal Security Analysis through BAN

The Burrows–Abadi–Needham (BAN) logic analysis is performed to test the protocol from various security aspects with a focus on mutual key agreement, key sharing, and protection from exposure to session key. We used the following symbolic tokens to perform this analysis.

(i) $L| \equiv \overline{w}$: L believes $\overline{w}$
(ii) $L \triangleleft \overline{w}$: L sees $\overline{w}$
(iii) $L| \sim \overline{w}$: L once said $\overline{w}$, some time ago
(iv) $L| \Rightarrow \overline{w}$: L has got the entire jurisdiction over $\overline{w}$
(v) (#bar$w$): the message $\overline{w}$ is fresh
(vi) $(L)\overline{w}$: L is used in formulae with $\overline{w}$
(vii) $(\overline{w}, \overline{w}')_k$: $\overline{w}$ or $\overline{w}'$ is symmetrically encrypted with key K
(viii) $\{\overline{w}, \overline{w}'\}_k$: $\overline{w}$ or $\overline{w}'$ is hashed with key K
(ix) $\{L, \overline{w}\}_k$: K is used in formula with $\overline{w}$ and L
(x) $LK \leftrightarrow L'$: L communicates with the key K

The following BAN logic rules are used to verify the security features:

Rule 1. Message meaning.

$$\frac{L| \equiv LK \leftrightarrow L', L \triangleleft < \overline{w} >_{\overline{w}}}{L| \equiv L| \sim \overline{w}}. \tag{5}$$

Rule 2. Nonce verification.

$$\frac{L| \equiv \#(\overline{w}), L| \equiv L'| \sim \overline{w}}{L| \equiv L'| \equiv \overline{w}}. \tag{6}$$

Rule 3. Jurisdiction.

$$\frac{L| \equiv L' \Rightarrow \overline{w}, L| \equiv L'| \equiv \overline{w}}{L| \equiv \overline{w}}. \tag{7}$$

Rule 4. Freshness conjunction.

$$\frac{L| \equiv \#(\overline{w})}{L| \equiv \#(\overline{w}, \overline{w}')}. \tag{8}$$

Rule 5. Belief rule.

$$\frac{L| \equiv (\overline{w}), L| \equiv (\overline{w}')}{L| \equiv (\overline{w}, \overline{w}')}. \tag{9}$$

Rule 6. Session key.

$$\frac{L| \equiv \#(\overline{w}), L| \equiv L' \equiv \overline{w}}{L| \equiv LK \leftrightarrow L'}. \tag{10}$$

Corresponding with the above rules and assumptions, we accomplish the following goals in the BAN logic analysis. The symbols used here, i.e., (g, RSU_j, V_i, V_s), represent the goal, road side unit, vehicle, and vehicle server.

(i) G1: $RSU_j| \equiv (RSU_j \overset{SK}{\leftrightarrow} V_s)$
(ii) G2: $RSU_i| \equiv V_s| \equiv (RSU_i \overset{SK}{\leftrightarrow} V_s)$
(iii) G3: $V_i| \equiv (RSU_j \overset{SK}{\leftrightarrow} V_i)$
(iv) G4: $V_i| \equiv RSU_j| \equiv (RSU_j \overset{SK}{\leftrightarrow} V_i)$
(v) G5: $V_s| \equiv (V_i \overset{SK}{\leftrightarrow} V_s)$
(vi) G6: $V_s| \equiv (V_i \overset{SK}{\leftrightarrow} V_i)$

Initially, the communication contents must be adapted into idealized form as shown in the following:

(i) M1: $V_i \longrightarrow RSU_j$: $PID_{vi}, M_{i1}, M_{i2}, t_i$: $\{PID_{vi}, ID_{vi}, t_i, (r_i)_{Bvi}, t_i\}$
(ii) M2: $RSU_j \longrightarrow V_s$: $ID_{rj}, PID_{vi}, \langle ID_{vi}, t_i, r_{iAvi} \rangle, \langle ri_{Bvi} \rangle, t_i, M_{j1}, t_j$: $\{ID_{rj}, PID_{vi}, \langle ID_{vi}, t_i, r_{iAvi} \rangle, (r_i)_{Bvi}, t_i, \{r_j, t_j\}_{Krj}, t_j\}$
(iii) M3: $V_s \longrightarrow RSU_j$: RSK, VSK, RSV, t_{vs}: $\{(SK)_{Krj}, \{SK, PID_{vin}, B_{vin}\}_{Avi}, (tvs)_{SK}, t_{vs}\}$
(iv) M4: $RSU_s \longrightarrow V_i$: VSK, VSV, T_{jn}: $\{\{SK, PID_{vin}\}_{Avi}, (t_{jn})_{SK}, t_{jn}\}$

Furthermore, we take the following assumptions to support the security proof.

(i) $B1$: $V_i| \equiv \#(t_i)$
(ii) $B2$: $RSU_j| \equiv \#t_j, t_{jn}$
(iii) $B3$: $V_s| \equiv \#t_{vs}$
(iv) $B4$: $V_i| \equiv (V_i A_{vi} \leftrightarrow V_s)$
(v) $B5$: $V_i| \equiv (V_i \overset{SK \| t_{jn}}{\leftrightarrow} RSU_j)$
(vi) $B6$: $RSU_J| \equiv (RSU_j \overset{SK \| t_{jn}}{\leftrightarrow} V_i)$
(vii) $B7$: $RSU_j| \equiv RSU_j \overset{K_{rj}}{\leftrightarrow} RSU_j$
(viii) $B8$: $V_s| \equiv (V_s A_{vi} \leftrightarrow V_i)$
(ix) $B9$: $V_s| \equiv V_s \overset{K_{rj}}{\leftrightarrow} RSU_j$
(x) $B10$: $V_i| \equiv RSU_j| \equiv V_i \overset{SK}{\leftrightarrow} RSU_j$
(xi) $B11$: $RSU_j| \equiv V_i| \equiv V_i \overset{SK}{\leftrightarrow} RSU_j$

(xii) $B12$: $V_s| \equiv V_i| \equiv V_i \overset{SK}{\leftrightarrow} V_s$

(xiii) $B13$: $\text{RSU}_j| \equiv V_s| \equiv V_s \overset{SK}{\leftrightarrow} RSU_j$

(xiv) $B14$: $V_s| \equiv \text{RSU}_j| \equiv V_i \overset{SK}{\leftrightarrow} \text{RSU}_i$

(xv) $B15$: $V_i| \equiv V_s| \equiv V_i \overset{SK}{\leftrightarrow} \text{RSU}_j$

Next, employing the above assumptions, we further analyze the idealized forms.

Taking the idealized version of M1 and M2:

(i) M1: $V_i \longrightarrow \text{RSU}_j$: $\text{PID}_{vi}, M_{i1}, M_{i2}, t_i$: $\{\text{PID}_{vi}, \text{ID}_{vi}, t_i, (r_i)_{Bvi}, t_i\}$

(ii) M2: $\text{RSU}_j \longrightarrow V_s$: $\text{ID}_{rj}, \text{PID}_{vi}, \langle \text{ID}_{vi}, t_i, r_{iAvi} \rangle, \langle ri_{Bvi} \rangle, t_i,$

(iii) M_{j1}, t_j: $\{\text{ID}_{rj}, \text{PID}_{vi}, \langle \text{ID}_{vi}, t_i, r_{iAvi} \rangle, (r_i)_{Bvi}, t_i, \{r_j, t_j\}_{Krj}, t_j\}$

By applying seeing rule, we get

(i) $X1$: $\text{RSU}_j \triangleleft \{\text{PID}_{vi}, M_{i1}, M_{i2}, t_i$: $\{\text{PID}_{vi}, \text{ID}_{vi}, t_i, (r_i)_{Bvi}, t_i\}\}$

(ii) $X2$: $V_s \triangleleft \{\text{ID}_{rj}, \text{PID}_{vi}, \langle \text{ID}_{vi}, t_i, r_{i_{Avi}} \rangle, \langle ri_{Bvi} \rangle, t_i, M_{j1}, t_j$: $\{\text{ID}_{rj}, \text{PID}_{vi}, \langle \text{ID}_{vi}, t_i, r_{i_{Avi}} \rangle, (r_i)_{Bvi}, t_i, \{r_j, t_j\}_{Krj}, t_j\}\}$

According to $D1, D2, P8, B9$, and $R1$, we get

(i) $X3$: $V_s| \equiv V_i \sim \{\text{PID}_{vi}, \text{ID}_{vi}, t_i, (r_i)_{Bvi}, t_i\}$

(ii) $X4$: $V_s| \equiv \text{RSU}_j \sim \{\text{ID}_{rj}, \text{PID}_{vi}, \langle \text{ID}_{vi}, t_i, r_{i_{Avi}} \rangle, (r_i)_{Bvi}, t_i, \{r_j, t_j\}_{Krj}, t_j\}$

Referring to $X3, B1, R2$, and $R4$, we get

(i) $X5$: $V_s| \equiv V_i \equiv \{\text{PID}_{vi}, \text{ID}_{vi}, t_i, (r_i)_{Bvi}, t_i\}$

(ii) $X6$: $V_s| \equiv \text{RSU}_j \equiv \{\text{ID}_{rj}, \text{PID}_{vi}, \langle \text{ID}_{vi}, t_i, r_{i_{Avi}} \rangle, (r_i)_{Bvi}, t_i, \{r_j, t_j\}_{Krj}, t_j\}$

Referring to $X5, B12$, and $R3$,

(i) $X7$: $V_s| \equiv \{\text{PID}_{vi}, \text{ID}_{vi}, t_i, (r_i)_{Bvi}, t_i\}$

In accordance with $X6, B14$, and $R3$, we have

(i) $X8$: $V_s| \equiv \{\text{ID}_{rj}, \text{PID}_{vi}, \langle \text{ID}_{vi}, t_i, r_{i_{Avi}} \rangle, (r_i)_{Bvi}, t_i, \{r_j, t_j\}_{Krj}, t_j\}$

Referring to $X5, X7$, and $R6$, we have

(i) $X9$: $V_s| \equiv V_i \overset{SK}{\leftrightarrow} V_s$ (goal 5)

Using $X5, X7, B8$, and $R2$, we get

(i) $X10$: $V_s| \equiv V_i| \equiv V_i \overset{SK}{\leftrightarrow} V_s$ (goal 6)

Taking the idealized version of $M3$,

(i) $M3$: $V_s \longrightarrow \text{RSU}_j$: $\text{RSK}, \text{VSK}, \text{RSV}, t_{vs}$: $\{(\text{SK})_{Krj}, \{\text{SK}, \text{PID}_{vin}, B_{vin}\}_{Avi}, (\text{tvs})_{\text{SK}}, t_{vs}\}$

On the application of seeing rule for $M3$, we get

(i) $X11$: $V_s| \equiv V_i \sim \text{RSK}, \text{VSK}, \text{RSV}, t_{vs}$: $\{(\text{SK})_{Krj}, \{\text{SK}, \text{PID}_{vin}, B_{vin}\}_{Avi}, (\text{tvs})_{\text{SK}}, t_{vs}\}$

Using $X11, B7$, and $R1$, we have

(i) $X12$: $\text{RSU}_j| \equiv V_s \sim \{(\text{SK})_{Krj}, \{\text{SK}, \text{PID}_{vin}, B_{vin}\}_{Avi}, (tvs)_{\text{SK}}, t_{\text{vs}}\}$

(ii) $\text{RSU}_j| \equiv (\text{RSU}_j \overset{SK}{\leftrightarrow} V_s)$ (goal 1)

According to $X12, B3, B13, R2$, and $R4$, we have

(i) $X13$: $\text{RSU}_j| \equiv V_s | \equiv \{(\text{SK})_{Krj}, \{\text{SK}, \text{PID}_{vin}, \backslash\backslash B_{vin}\}_{Avi}, (tvs)_{\text{SK}}, t_{\text{vs}}\}$

(ii) $\text{RSU}_i| \equiv V_s| \equiv (\text{RSU}_i \overset{SK}{\leftrightarrow} V_s)$ (goal 2)

Next, considering $M4$ idealized form,

(i) M4: $\text{RSU}_s \longrightarrow V_i$: $\text{VSK}, \text{VSV}, T_{jn}$: $\{\{\text{SK}, \text{PID}_{vin}\}_{Avi}, (t_{jn})_{\text{SK}}, t_{jn}\}$

On the application of seeing rule for $M4$, we have

(i) $X14$: $V_i \triangleleft V_i$: $\text{VSK}, \text{VSV}, T_{jn}$: $\{\{\text{SK}, \text{PID}_{vin}\}_{Avi}, (t_{jn})_{\text{SK}}, t_{jn}\}$

While $X14, B4, B5$, and $R1$ imply

(i) $X15$: $V_i| \equiv RSU_j \sim \{\{\text{SK}, \text{PID}_{vin}\}_{Avi}, (t_{jn})_{\text{SK}}, t_{jn}\}$

Referring to $X15, B2, B3, R2$, and $R4$, we have

(i) $X16$: $V_i| \equiv \text{RSU}_j| \equiv \{\{\text{SK}, \text{PID}_{vin}\}_{Avi}, (t_{jn})_{\text{SK}}, t_{jn}\}$

From $X16, B4, B10, B15$, and rule 3, we get

(i) $X17$: $V_i| \equiv \{\{\text{SK}, \text{PID}_{vin}\}_{Avi}, (t_{jn})_{\text{SK}}, t_{jn}\}$

Referring to $X17$, we apply $R6$ as

(i) $X18$: $V_i| \equiv (\text{RSU}_j \overset{SK}{\leftrightarrow} V_i)$ (goal 3)

According to $X18, B2$, we apply the $R6$ as

(i) $X19$: $V_i| \equiv \text{RSU}_j| \equiv (\text{RSU}_j \overset{SK}{\leftrightarrow} V_i)$ (goal 3)

The discussed cases for proving the protocol in BAN logic make obvious that the contributed scheme entirely supports mutual authentication and protects the established session key among the three participating members.

5. Informal Security Analysis

An informal security discussion on the security features of the proposed scheme is provided in following subsection:

5.1. Mutual Authentication. The SMEP-IoV ensures mutual authenticity for all participating entities of the system. In particular, the RSU_j authenticates both entities, VS and V_i, by means of equality check comparing RSV against the computed $h(\text{SK} \| t_{\text{vs}})$ parameter. Since, RSU_j is aware of the fact, the generated session key SK can only be constructed by a legitimate VS entity having access to master secret key K_{vs}. Using K_{vs}, VS can access r_i, r_j, A_{vi}, and K_{rj} factors to compute a valid SK. Likewise, V_i authenticates RSU_j on the basis of VSV equality check, after comparing it with the computed $h(\text{SK} \| t_{jn})$. Similarly, Vs authenticates V_i by computing $h(A_{vi} \| \text{ID}_{vi} \| t_i \| r_i)$ against M_{i1}. Realizing the fact that A_{vi} is only held with a valid V_i entity, it can validate the

vehicle V_i. If these equality checks fail, the mutual authentication cannot be assured in the protocol.

5.2. Stolen Verifier Attack. In the proposed scheme, the vehicle server VS stores only public identities ($\{ID_{vi}: i = 1, 2 \ldots n\}$) of all the registered vehicles in its memory. VS does not store any other vehicle-related secret parameter in its own memory, and the verifier is with the vehicle. Therefore, the possibility of stolen verifier attack on proposed SMEP-IoV is negligible.

5.3. Vehicle Anonymity. The SMEP-IoV employs a pseudoidentity PID_{vi} for each vehicle, which is renewed and replaced after the termination of each session. In this manner, the vehicle or user remains anonymous during the execution of the protocol. Moreover, there is no desynchronization possible in case an adversary holds or blocks the message on its way.

5.4. VS Impersonation Attack. No adversary A can impersonate as Vs in the SMEP-IoV scheme. This is because, if an adversary attempts the same towards V_i, the latter may discern the possibility of attack by comparing VSV against the computed factor $h(SK\|t_{jn})$. Similarly, if A attempts to impersonate as VS against RSU_j, the RSU_j may successfully thwart this attack on the basis of comparison of RSV and calculated $h(SK\|t_{vs})$. Hence, the SMEP-IoV is immune to VS impersonation attack.

5.5. RSU Impersonation Attack. The SMEP-IoV is immune to RSU impersonation attack. Both entities V_i and VS may easily prevent any attempt of impersonation as RSU on the part of adversary. This is due to the fact that VS shares a secret with RSU_j. The use of fresh timestamps along with the shared secrets helps the VS entity in authenticating a legitimate RSU. Similarly, V_i authenticates RSU_j on account of the derived session key SK from the VSK message as submitted by a valid VS, which is further used in the later comparison of VSV. In this manner, both of the entities validate a legal RSU_j on account of provided logical comparison of equality checks.

5.6. Man-in-the-Middle Attack (MiDM). To launch a successful MiDM attack on SMEP-IoV, the adversary needs access to either the V_i registration parameters such as A_{vi} and B_{vi} or access to secret key K_{rj} or the master secret key K_{vs}. On the other hand, as we see earlier, it is less likely for an adversary to initiate an impersonation attack on the protocol.

5.7. Session Key Security. As we see earlier, no adversary could engage in the mutual authentication process until it gains access to secure credentials of the system either held by the registration authority or registered entities. Since, the SMEP-IoV provides mutual authentication to all participants, the established session key is only known to the legitimate members involved in the protocol execution.

5.8. Denial of Service. Our scheme is resistant to denial of service attacks, since it engages fresh timestamps for the generation of M_{vi} and M_{rj1}. Due to these timestamps, the receiving entity may check the freshness of the incoming message and discard the message immediately if the latency is beyond a certain preset threshold.

5.9. Replay Attack. In case an adversary attempts to initiate a replay attack towards any entity V_i, RSU_j, or VS, the SMEP-IoV may foil this attempt immediately after checking the freshness of timestamps t_i, t_j/t_{jn}, and t_{vs}, respectively. Hence our scheme is immune to this threat.

6. Performance and Security Comparisons

The performance and security comparisons of the proposed scheme with related existing scheme [22–24] are explained in the following subsections.

6.1. Performance Comparisons. For measuring the computation time and cost, Pi3 B+ is used with Cortex A53 (ARMv8) 64 bit SoC and with processing speed 1.4 GHz along with 1 GB LPDDR2 SDRAM RAM. The simulation results of basic operations executed over Pi3 are given in Table 2. For completion of authentication and a key agreement (AKA) among a vehicle V_i and RSU RSU_j through the intermediate agent VS-Vehicle Server, V_i executes $2C_{hs}$ and $3C_{ed}$ operations. Likewise, RSU_j performs $2C_{hs} + 2C_{ed}$ operations while VS accomplishes $7C_{hs}$ and $7C_{ed}$ operations. Hence, total computational operations performed to complete a cycle of AKA are $11C_{hs} + 12C_{ed}$. Using the experiment with computational times represented in Table 2, the performance comparisons are briefed in Table 3. The proposed scheme completes single AKA cycle in ≈0.198 ms. In contrast to the proposed scheme, the scheme of Yu et al. [23] completes single AKA cycle in ≈0.132 ms, the scheme of Vasudev et al. [22] and Mohit et al. [24] complete the one cycle of AKA in ≈0.082 ms and ≈0.108 ms, respectively.

For communication cost comparisons, subsequent consideration is taken as per the sizes of different parameters. Timestamps and identity are taken as 32 and 64 bits, respectively; whereas, the sizes of the outputs of the symmetric key and asymmetric key operations are taken as 128 and 1024 bits. The value of hash output is fixed at 160 bits. Moreover, the size of random numbers is also assumed as 160 bit of length. The communication cost of SMEP-IoV and related schemes of Yu et al. [23], Vasudev et al. [22], and Mohit et al. [24] is computed as the bits exchanged among the IoV entities. The V_i sends $M_{vi} = \{PID_{vi}, M_{i1}, M_{i2}, t_i\}$ to RSU_j, where the size of M_{vi} is $\{128 + 160 + 160 + 32\} = 480$. Subsequently, the RSU_j sends $M_{rj1} = \{ID_{rj}, M_{vi}, M_{j1}, t_j\}$, where the size of M_{rj} is $\{64 + 480 + 128 + 32\} = 704$. The VS replies RSU_j with $M_{vs} = \{RSK, VSK, RSV, t_{vs}\}$, where the

Table 2: Operational Cost of the primitives.

Operation	Notation	Time (ms)
Enc-decryption	C_{ed}	≈0.011
Hash function	C_{hs}	≈0.006

Table 3: Performance comparisons.

↓ protocols	C_a.	RT	C_b
Proposed	$11C_{hs} + 12C_{ed}$	≈0.198	2848
Yu et al. [23]	$22C_{hs}$	≈0.132	864
Vasudev et al. [22]	$10C_{hs} + 2C_{ed}$	≈0.082	800
Mohit et al. [24]	$18C_{hs}$	≈0.108	1760

Note: C_a, computation cost; RT, running time in ms; C_b, communication cost in bits.

Table 4: Security features.

Schemes	Our	[23]	[22]	[24]
Mutual authentication	✓	✗	✗	✓
Stolen verifier	✓	✓	✓	✓
Vehicle anonymity	✓	✗	✓	✓
VS impersonation	✓	✗	✗	✓
RSU impersonation	✓	✗	✗	✓
Vehicle impersonation	✓	✗	✗	✓
Man in middle attack	✓	✗	✗	✗
Session key security	✓	✗	✗	✓
Denial of service	✓	✓	✓	✓
Replay attack	✓	✓	✓	✓

Note: ✓, provides or resists; ✗,does not provide or does not resist.

size of M_{vs} is $\{256 + 512 + 160 + 32\} = 960$. The final message $M_{rj2} = \{VSK, VSV, t_{j_n}\}$ was sent from RSU_j to V_i, where the size of M_{rj2} is $\{512 + 160 + 32\} = 704$. Therefore, total communication cost is 2848 bits. The communication cost of Yu et al.'s scheme is 864, while the communication costs of Vasudev et al. and Mohit et al. are 800 and 1760, respectively.

6.2. Security Features. The security comparisons of the SMEP-IoV and related existing schemes [22–24] are provided in this subsection. Table 4 solicits the summary of the security comparisons. Due to disclosure of master secret key K_{VS}, the Yu et al.' scheme [23] is vulnerable to many attacks including impersonation of vehicle, RSU, and vehicle server, along with session key disclosure and vehicle/user anonymity violations attack. The scheme of Vasudev et al. [22] lacks mutual authentication and has insecurities against vehicle, RSU, and vehicle server impersonation attacks. Moreover, Vasudev et al.'s scheme is insecure against man-in-the-middle attack. The scheme of Mohit et al. [24] is also weak against man in middle attack. In contrast, proposed SMEP-IoV provides all security features and is robust against the known attacks.

7. Conclusion

Initially, this study reviewed some of the recent authentication schemes for securing IoVs. Then, we developed a symmetric key-based authentication scheme, through which a vehicle can share a secret key with corresponding RSU through the mediation of the vehicle server. The proposed secure message exchange protocol for IoV (SMEP-IoV) uses only lightweight symmetric encryption and hash functions. The comparisons of the SMEP-IoV show that proposed scheme compromises slight performance overhead and provides adequate security, which other competing schemes do not provide. Hence, due to performance and security provisions, SMEP-IoV best suits the security requirements of the fast moving vehicles in the IoV scenario.

Data Availability

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

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

The author would like to thank the the academic editor Dr. Prosanta Gope for valuable suggestions to improve the quality, correctness, presentation, and readability of the manuscript.

References

[1] M. Kamal, G. Srivastava, and M. Tariq, "Blockchain-based lightweight and secured v2v communication in the internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–8, 2020.

[2] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[3] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Proceedings of the International conference on provable security*, pp. 1–16, Springer, Wollongong, NSW, Australia, November 2007.

[4] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[5] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," *Lecture Notes in Computer Science Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Bruges, Belgium, pp. 453–474, May 2001.

[6] S. A. Chaudhry, M. S. Farash, N. Kumar, and M. H. Alsharif, "Pflua-diot: a pairing free lightweight and unlinkable user access control scheme for distributed iot environments," *IEEE Systems Journal*, pp. 1–8, 2020.

[7] S. Hussain, S. A. Chaudhry, O. A. Alomari, M. H. Alsharif, M. K. Khan, and N. Kumar, "Amassing the security: an ecc-based authentication scheme for internet of drones," *IEEE Systems Journal*, pp. 1–8, 2021.

[8] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles: architecture, protocols, and security," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3701–3709, 2018.

[9] B. Mokhtar and M. Azab, "Survey on security issues in vehicular ad hoc networks," *Alexandria Engineering Journal*, vol. 54, no. 4, pp. 1115–1126, 2015.

[10] S. A. Chaudhry, K. Yahya, F. Al-Turjman, and M. H. Yang, "A secure and reliable device access control scheme for iot based sensor cloud systems," *IEEE Access*, vol. 8, pp. 139244–139254, 2020.

[11] S. A. Chaudhry, "Correcting "PALK: password-based anonymous lightweight key agreement framework for smart grid," *International Journal of Electrical Power and Energy Systems*, vol. 125, Article ID 106529, 2021.

[12] S. A. Chaudhry, I. L. Kim, S. Rho, M. S. Farash, and T. Shon, "An improved anonymous authentication scheme for distributed mobile cloud computing services," *Cluster Computing*, vol. 22, no. 1, pp. 1595–1609, 2019.

[13] A. Irshad, M. Usman, S. A. Chaudhry, H. Naqvi, and M. Shafiq, "A provably secure and efficient authenticated key agreement scheme for energy internet-based vehicle-to-grid technology framework," *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 4425–4435, 2020.

[14] X. Lin, X. Sun, X. Wang, C. Zhang, P.-H. Ho, and X. Shen, "Tsvc: timed efficient and secure vehicular communications with privacy preserving," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 4987–4998, 2008.

[15] B. Ying, D. Makrakis, and H. T. Mouftah, "Privacy preserving broadcast message authentication protocol for vanets," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1352–1364, 2013.

[16] J. Li, H. Lu, and M. Guizani, "Acpn: a novel authentication framework with conditional privacy-preservation and non-repudiation for vanets," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 938–948, 2015.

[17] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4359–4373, 2018.

[18] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu, "2flip: a two-factor lightweight privacy-preserving authentication scheme for vanet," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 896–911, 2016.

[19] R. Amin, P. Lohani, M. Ekka, S. Chourasia, and S. Vollala, "An enhanced anonymity resilience security protocol for vehicular ad-hoc network with scyther simulation," *Computers and Electrical Engineering*, vol. 82, pp. 1–18, 2020.

[20] Y. Liu, Y. Wang, and G. Chang, "Efficient privacy-preserving dual authentication and key agreement scheme for secure v2v communications in an iov paradigm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2740–2749, 2017.

[21] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10626–10636, 2017.

[22] H. Vasudev, D. Das, and A. V. Vasilakos, "Secure message propagation protocols for iovs communication components," *Computers and Electrical Engineering*, vol. 82, pp. 1–15, 2020.

[23] S. Yu, J. Lee, K. Park, A. K. Das, and Y. Park, "Iov-smap: secure and efficient message authentication protocol for iov in smart city environment," *IEEE Access*, vol. 8, pp. 167875–167886, 2020.

[24] P. Mohit, R. Amin, and G. P. Biswas, "Design of authentication protocol for wireless sensor network-based smart vehicular system," *Vehicular Communications*, vol. 9, pp. 64–71, 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5514944, 15 pages
https://doi.org/10.1155/2021/5514944

Research Article

Chaotic Reversible Watermarking Method Based on IWT with Tamper Detection for Transferring Electronic Health Record

Mahboubeh Nazari and Arash Maneshi

Department of Computer Engineering, Imam Reza International University, Mashhad, Iran

Correspondence should be addressed to Arash Maneshi; maneshiarash@imamreza.ac.ir

Received 18 February 2021; Accepted 26 April 2021; Published 18 May 2021

Academic Editor: Azeem Irshad

Health IoT deals with sensitive medical information of patients, therefore security concerns need to be addressed. Confidentiality of Electronic Health Record (EHR) and privacy are two important security requirements for IoT based healthcare systems. Recently, watermarking algorithms as an efficient response to these requirements is in the spotlight. Further, as smart city-based applications have to react to real-time situations, efficient computation is a demand for them. In this paper, a secure, lightweight, reversible, and high capacity watermarking algorithm with tamper detection capability is proposed for IoT based healthcare systems. The scheme has applied Integer Wavelet Transform (IWT) and chaotic map for efficiency and increasing security. EHR is encrypted and then embedded into the Least Significant Bits (LSB) of wavelet coefficients of medical images. The proposed method has been extensively tested for various color and grayscale commonly used medical and general images. Investigations on experimental results and criterions such as Peak Signal to Noise Ratio (PSNR) and Bit Error Ratio (BER) above 45.41 dB and 0.04, respectively, for payload of 432,538 bits indicate that the proposed method, besides providing security, being reversible, tamper detection capability, and high embedding capacity, has high imperceptibility and adequate resistance against different types of attacks.

1. Introduction

Nowadays, exchanging sensitive data through insecure Internet channels is inevitable. Moreover, the Internet of Things (IoT) is a revolutionary technology that prepares a reliable infrastructure for actuators and sensors to collaborate and exchange data with each other. By using IoT opportunities, the healthcare system will be revolutionized. IoT based healthcare systems with the capability of collecting patient's real-time health data have attracted many researchers' attention [1].

Patient healthcare data which was collected by equipment and sensors of IoT based healthcare system is used to create Electronic Health Record (EHR) for each patient. EHR, including medical images, Electronic Patient Record (EPR), etc., plays an important role in the diagnosis process of a patient. Thus, any manipulation and tampering in such reports may cause fatal diagnosis to the patient. EHR are exchanged among hospitals, doctors, and insurance companies; therefore preserving the confidentiality of EHR and privacy of patients are the most important security requirements. On the other side, an IoT driven healthcare system must be capable of handling the real-time situation. Therefore, data transferring and computational overhead in such a system have to be efficient. In this scope, researchers are looking for alternative solutions that are proper for resource-constrained IoT devices.

The information hiding techniques are an adequate solution to address these challenges. Information hiding is classified mainly into steganography and watermarking. In both of them, secret data is embedded in the cover data such as text, voice, and pictures with different goals. In the steganography, data hiding in digital media should be done in such a way that the existence of secret data in the host media is unnoticeable. In these algorithms, the cover could be unrelated to the secret data, and only transferring the

hidden secret data and imperceptibility are important. But in watermarking algorithms, the secret data is dependent on the cover to preserve its security through manipulations. In some watermarking applications, information on cover media is hidden inside of cover media for copyright protection or content authentication purposes.

There are two spatial and transform domains for applying steganography and watermarking algorithms. In the spatial domain, the secret message is embedded directly into the cover image by manipulating pixel values. Spatial domain-based algorithms have less computational overhead and less resistance to different types of attacks. In the transform domain, at first, the cover image is transformed by using various transform operations such as Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), and Discrete Wavelet Transform (DWT), and then a secret message is embedded into coefficients [2]. Transform domain-based algorithms have more computational overhead and more resistance against different types of attacks. For designing a practical scheme for smart city applications, a trade-off between computational complexity and security must be considered.

The main requirements for all watermarking and steganography algorithms are imperceptibility, robustness, and payload capacity. Imperceptibility refers to the amount of deterioration that has been made to the cover media and robustness indicates how much a watermark can resist against different attacks. Corresponding to the application, a different level of robustness is required. Payload capacity also refers to the amount of data embedded inside of a cover media. For some watermarking applications, reversibility is a requirement which means that, after the extraction of the secret messages, host media have to be recoverable [3–5]. In irreversible techniques host media do not need to be restored after extraction of secret message. Considering applications, reversibility is not a requirement for all steganography techniques which could bring more payload capacity [6–8]. Reversible information hiding finds applications in e-healthcare, military communication, and smart cities, etc., where host media need to be recovered after the extraction phase [9].

As already mentioned, low computational complexity while keeping a high amount of payload is highly desirable feature for real-time situations like IoT based electronic healthcare systems and smart city applications. Many high capacity information hiding schemes have been suggested for image [9–14]. Most of them have used the interpolation concept to generate a resized version of the original image before embedding. These techniques estimate pixels values of the resized image by using several algebraic equations which increase computational complexity.

In this paper, we have tried to address these challenges. To provide confidentiality, EHR is encrypted by using a chaotic sequence that is unpredictable for an adversary. To tamper detection, an Integrity Check Code (ICC) is calculated. Moreover, computational complexity is reduced while keeping high capacity. The encrypted EHR and ICC are embedded in two Least Significant Bits (LSB) of cover image's Integer Wavelet Transform (IWT), being a fast and lossless transform. Therefore, the confidentiality of EHR and patient privacy are preserved during transmission.

The important contributions of the proposed method are as follows:

(1) Designing an efficient reversible watermarking algorithm with high capacity and tamper detection, providing EHR confidentiality, and preserving patient privacy for IoT based healthcare system for both grayscale and color images.

(2) Providing a high level of security by using the chaotic sequence as an efficient tool intelligently in host coefficients selection, encryption process, and structure of the symmetric key.

(3) Keeping quality of the cover image high and reducing computational complexity significantly in comparison to the state-of-the-art Pixel Repetition Method (PRM), Neighbor Mean Interpolation (NMI), and Interpolation Neighboring Pixels (INP) techniques.

The rest of the paper is organized as follows. In Section 2, a literature review is presented. Preliminaries are discussed in Section 3. Section 4 also describes the proposed method in detail, and Section 5 provides detailed experimental results and discussion. The conclusion of the paper is in Section 6.

2. Literature Review

In recent years, because of prominent advancements and expansion of IoT and prepared infrastructure for deploying IoT in real life, researchers have tried to take advantage of IoT potentials in real-life scenarios. One of the most attractive topics in this area is using IoT potentials to present smarter healthcare services. Healthcare systems always look for reducing costs and providing better quality in healthcare services. Therefore, many approaches have been introduced in this area. An extensive survey for applications of IoT for healthcare systems, IoT based technologies for healthcare, IoT security including security requirements and attacks classification, and taxonomy have been reported in [1, 15, 16]. Constraints of IoT sensors are also discussed in [17].

Various schemes have been proposed for home healthcare monitoring and telemonitoring. These schemes are based on Wireless Sensor Networks (WSN), Near Field Communication (NFC), and Radio Frequency Identification (RFID). Using IoT components based on these technologies, techniques for fall detection and seizure detection were introduced in [18–21]. In [20], a system was proposed which detects the risk of bedsores by using sensors. Furthermore, applications of IoT based wearable devices for monitoring Parkinson's gait disturbance and cardiac and neurological disorders are reported in [21]. These systems make it possible for caregivers to prevent dangerous situations by taking immediate action and providing better treatment.

Although the above schemes have proposed architectures for mHealth/eHealth, a seamless method for securing patient's EHR has not been reported [9]. Since EHR

contains patient's vital health data, designing a method that provides confidentiality for EHR, preserves patient's privacy, and applies to resource-constrained IoT devices for real-time scenarios in the smart city is highly desirable. Considering requirements and challenges, information hiding techniques seem to be an adequate option to design and develop such a system. Unlike conventional cryptography, information hiding techniques hide information inside of a cover medium intelligently, instead of encrypting information by using costly algorithms. A survey of applications of information hiding techniques in medical and healthcare systems has been presented in [22]. In order to increase capacity for hiding more amount of data, various information hiding techniques based on interpolation have been introduced. Neighbor Mean Interpolation has been introduced by [23]. The average PSNR of this scheme is 35 dB for a payload of 1.622 BPP. Chin Feng Lee and Yu-Lin Huang [24] proposed Interpolation by Neighboring Pixels (INP) to improve the performance of the data hiding scheme proposed by [23]. The payload of this method is up to 2.28 BPP. In [9], a reversible information hiding method based on interpolation has been proposed. This method at first resizes the original image with a size of $M \times N$ into $M/2 \times N/2$. Then using an algorithm called Pixel Repetition Method (PRM) creates a cover image from the resized image by repeating pixels from the original image. Then EHR is embedded into the cover image. However, the interpolation-based method can increase embedding capacity but the calculating number of algebraic equations and operations such as upscaling and downscaling for creating the cover image from the original image increases computational complexity and decreases the original image quality.

A watermarking scheme for the security of medical and nonmedical images based on 2-level Singular Value Decomposition (2-D SVD) has been proposed by [25]. A digital signature is embedded into the cover image for tamper detection purposes; however, watermark is needed for this test. A blind watermarking approach for medical image protection was proposed by [26]. This method uses combination of DWT and SVD for embedding data into the cover image. SVD increases computational complexity significantly and is not a suitable choice for resource-constrained IoT devices and real-time scenarios.

Table 1 summarizes the articles that have been mentioned in literature review section in chronological order of year.

3. Preliminaries

3.1. Integer Wavelet Transform (IWT). Wavelets are basis functions used to represent signals. Integer Wavelet Transforms (IWT) are the wavelet transforms that map integers to integers [27]. The procedure of calculating IWT using lifting technique is illustrated in Figure 1 and described as follows.

Step 1. The image matrix is separated odd and even columns. Frequency subbands, HF (high-frequency components), and LF (low-frequency components) are calculated using the following equations:

$$\begin{aligned} \mathrm{HF} &= \mathrm{odd}(i, j) - \mathrm{even}(i, j), \\ \mathrm{LF} &= \mathrm{even}(i, j) + \frac{\mathrm{HF}}{2}. \end{aligned} \tag{1}$$

Step 2. $\mathrm{LF}_{\mathrm{even}}$ and $\mathrm{LF}_{\mathrm{odd}}$ show the even and odd rows of LF. $\mathrm{HF}_{\mathrm{even}}$ and $\mathrm{HF}_{\mathrm{odd}}$ are also even and odd rows of HF. First level decomposition of the image is calculated as follows:

$$\begin{aligned} \mathrm{HH} &= \mathrm{HF}_{\mathrm{odd}} - \mathrm{HF}_{\mathrm{even}}, \\ \mathrm{HL} &= \mathrm{HF}_{\mathrm{even}} + \frac{\mathrm{HH}}{2}, \end{aligned} \tag{2}$$

$$\mathrm{LH} = \mathrm{LF}_{\mathrm{odd}} - \mathrm{LF}_{\mathrm{even}}.$$

$$\mathrm{LL} = \mathrm{LF}_{\mathrm{even}} + \frac{\mathrm{LH}}{2}. \tag{3}$$

3.2. Logistic-Sine Map. Chaotic map is a mathematical concept which is equal to evolution function. These functions are extremely sensitive to their initial conditions and exhibit chaotic behavior that means a small change in input parameter leads to an unpredictable change in output. Logistic-Sine map (LS) is introduced in [28] as an intensive chaotic map defined as follows:

$$\begin{aligned} x_{n+1} &= \left(r.x_n(1 - x_n) + \frac{(4 - r)\sin(\pi x_n)}{4} \right) \mathrm{mod} 1 r \in (0, 4] \\ x_n &\in [0, 1]. \end{aligned} \tag{4}$$

The analyses of bifurcation diagram and Lyapunov exponent prove that the chaotic behaviors of the LS map exist in the whole range of parameter settings and its chaotic sequences have a uniform distribution within [0, 1] [28]. These properties of LS make it impossible for an adversary to predict the chaotic sequence. In other words, regenerating chaotic sequence without having the key (concatenation of r and x_n parameters) is not possible which makes LS a proper option to increase the security of the proposed method significantly.

4. Proposed Method

The proposed method can be divided into two main phases, embedding and extraction phase, which take place, respectively, on the sender and receiver side. An overview of the proposed method has been shown in Figure 2. At the sender side, the patient's medical image and EHR are obtained as inputs from which the Integrity Check Code (ICC) is calculated. Then EHR and ICC are encrypted using a chaotic sequence generated by LS map. IWT is applied to the medical image as a cover image, and then four frequency subbands are produced and picked to embed encrypted ICC

Table 1: Summary of articles in chronological order of year.

Article	Description	Year
[23]	Data hiding method based on interpolation	2009
[24]	Data hiding method based on interpolation	2012
[20]	Health status monitoring (monitoring the risk of bedsores)	2015
[18]	Health status monitoring (fall detection)	2017
[19]	Health status monitoring (home care systems)	2017
[21]	Health status monitoring (applications of wearable technologies)	2017
[9]	Reversible data hiding method based on interpolation	2018
[25]	Watermarking scheme with digital signature based on SVD	2019
[26]	Watermarking scheme with embedded hash based on SVD	2021

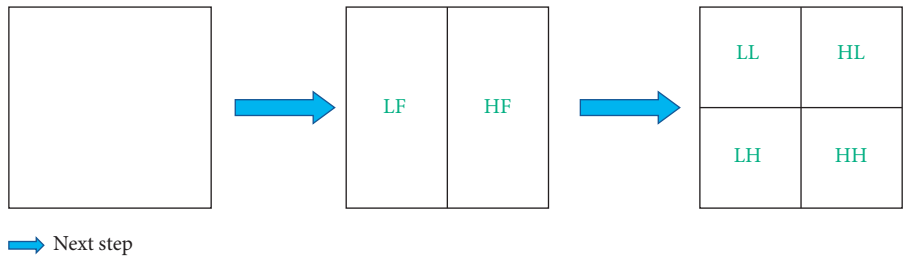


Figure 1: IWT calculation.

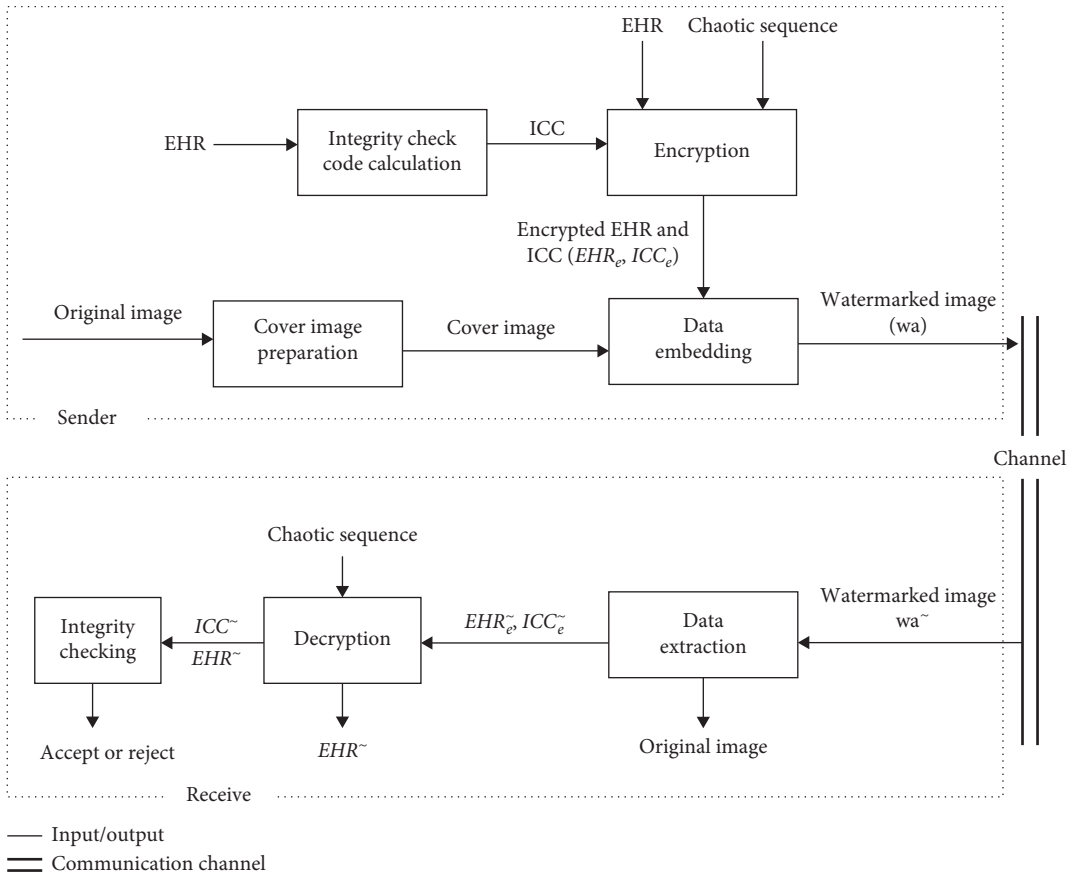


Figure 2: An overview of the proposed method.

and EHR into the LSB of coefficients. The data embedding procedure is described in Section 4.1 in detail. At the receiver side, the watermarked image is obtained and inverse IWT is applied to it. The four frequency subbands are picked to extract encrypted EHR and ICC from pixels LSB. The receiver regenerates chaotic sequence by using an agreed key

and then decrypts EHR and ICC which gives EHR′ and ICC′. Receiver recalculates ICC from EHR′, names it ICC″, and compares it to the ICC′ which was extracted and decrypted from the watermarked image. If ICC″ = ICC′ then the watermarked image is verified. The extraction procedure is described in Section 4.2 in detail.

4.1. Data Embedding Procedure. Data embedding procedure contains five subsections: cover image preparation, chaotic sequence generation, ICC calculation, EHR and ICC encryption, and data embedding, which are described in detail. Overview of data embedding is illustrated in Figures 3 and 4 .

4.1.1. Cover Image Preparation. To prepare the cover image, IWT is applied to the patient's medical image which produces 4 frequency subbands: LL, HL, LH, and HH. The following steps are taken to prepare the cover image:

(1) IWT is applied to the original image which produces LL, HL, LH, and HH frequency subbands

(2) HL, LH, and HH subbands are picked for EHR embedding

(3) LL subband is picked for ICC embedding

4.1.2. Chaotic Sequence (CS) Generation. To generate a chaotic sequence (CS), a key is needed. Concatenation of LS map parameters, x_0 and r, creates the key. Using this key, a chaotic sequence is generated, and the initial value effect can be faded by generating a sequence with a length three times bigger than that needed and picking the last segment. This sequence is used for encrypting EHR and ICC. The order of choosing subbands and pixels of each subband as host positions is based on the chaotic sequence. This leads to high confusion and diffusion which are the result of chaotic sequence features that make them appropriate for security aspects. The following steps are taken to generate CS:

(1) Choosing values for x_0 and r as the key

(2) Using the key as initial values of LS map to generate chaotic sequence (CS) with the size of $3n$, where $n = \text{size}(\text{EHR}||\text{ICC})$ is the length of the needed sequence

4.1.3. ICC Calculation. To detect tampering on a watermarked image, an Integrity Check Code (ICC) is calculated and embedded into the cover image.

The ASCII form of EHR is considered as sequence $W = p_1, p_2, p_3, p_4, p_5, \ldots, p_n$ where each p_n is an ASCII code of a letter. We divide this code into segments with length of five letters. The i^{th} segment of W is denoted by w_i, where $1 \le i \le (n/5)$. It should be noticed that if n is not the multiple of five, we add some padding. Corresponding to each segment $w_i = p_1p_2p_3p_4p_5$ which contains 40 bits, a four-bit code is produced by comparing each element of w_i with the central element p_3 as follows:

$$b_i = \begin{cases} 0, & p_j < p_3, \\ 1, & p_j > p_3, \end{cases} \quad 1 \le j \le 5. \tag{5}$$

All b_i come together to create B which is a compressed Integrity Check Code (ICC).

4.1.4. EHR and ICC Encryption. The CS that has been generated in step 4.1.2 is divided into two parts, CS_1 and CS_2, with the binary size of EHR and ICC, respectively. Then these parts are used to encrypt EHR and ICC to improve security. The encryption process consists of two stages:

(1) Scrambling: the order of the bits of EHR and ICC is changed based on new indices obtained from the sorted subsequences CS_1 and CS_2.

(2) Applying XOR operation: the binary forms of EHR and ICC are XOR with binary form of CS_1 and CS_2, respectively.

The following equations define encrypted EHR and ICC denoted by EHR_e and ICC_e, respectively.

$$\begin{aligned} \text{EHR}_e &= \text{EHR} \oplus \text{CS}_1, \\ \text{ICC}_e &= \text{ICC} \oplus \text{CS}_2. \end{aligned} \tag{6}$$

4.1.5. Data Embedding. The cover image was prepared in 4.1.1 subsection and ready to use for embedding. EHR_e is embedded into LSBs of coefficients in HL, LH, and HH. Moreover, LSBs of coefficients in LL band are hosts of bits in ICC_e. The order of choosing frequency subbands and coefficients for embedding EHR_e and ICC_e is based on chaotic sequence (CS). This technique makes it impossible for an attacker to recognize the host positions without having CS. At the end of this subsection, EHR_e is embedded into the cover image, and the cover image is watermarked with ICC_e. Then inverse IWT is applied to the watermarked image and the result will be sent. In summary for data embedding the following steps are taken:

(1) EHR_e bits are embedded into two LSBs of HL, LH, and HH coefficients of the cover image

(2) ICC_e bits are embedded into two LSBs of LL coefficients of cover image which produces the watermarked image

(3) Inverse IWT is applied to the watermarked image

4.2. Data Extraction Procedure. On the receiver side, at first, EHR is extracted from the watermarked image and then the integrity checking is done to ensure whether the watermarked image has been tampered or not. The data extraction process has been shown in Figures 5 and 6 . EHR and ICC extraction and integrity checking subsections are going to be described.

4.2.1. EHR and ICC Extraction. The receiver regenerates CS using the agreed key between the sender and the receiver which is the concatenation of x_0 and r. IWT applies to the

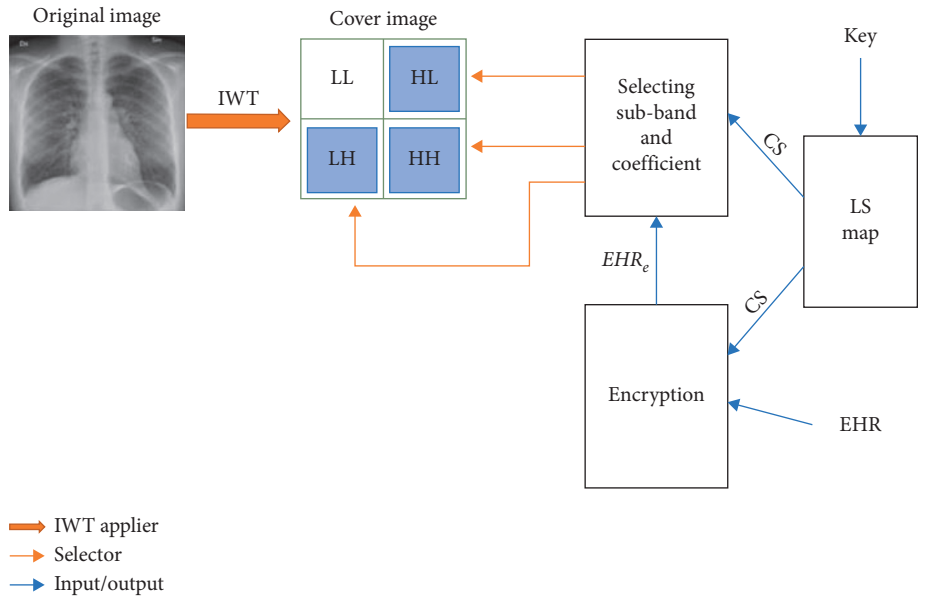


Figure 3: Preparing the cover image and embedding EHR into it.

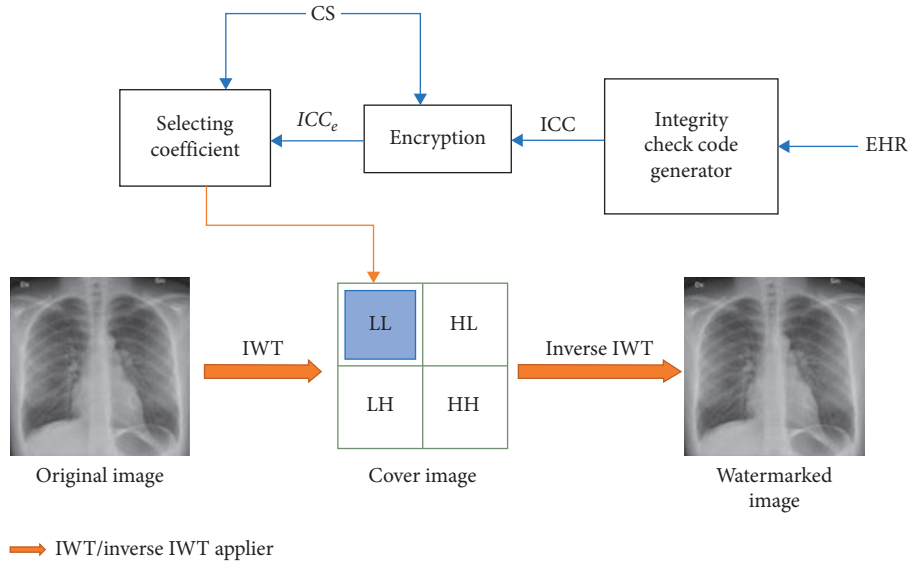


Figure 4: Calculating ICC and embedding it into the cover image.

watermarked image to produce LL, HL, LH, and HH sub-bands. According to the CS, similar to the sender side, the receiver can determine the selection order of host positions and extracts $EHR_e^{\sim}$ and $ICC_e^{\sim}$ from two LSBs of coefficients.

Regenerated CS is also used for decrypting $EHR_e^{\sim}$ and $ICC_e^{\sim}$ by equations $EHR^{\sim} = EHR_e^{\sim} \oplus CS_1$ and $ICC^{\sim} = ICC_e^{\sim} \oplus CS_2$ and then returning the scrambled bits to their first positions.

4.2.2. Integrity Checking. The ASCII code of $EHR^{\sim}$ is put in W and the receiver recalculates the ICC by using the same algorithm as described in 4.1.3 subsection. The integrity checking process has been demonstrated in Figure 6. The receiver compares recalculated ICC to $ICC^{\sim}$; if $ICC = ICC^{\sim}$ this means the watermarked image is valid, and otherwise, it is rejected. After checking the validity of the received watermarked image, inverse IWT is applied to the cover image to recover the patient's medical image. At this point, the receiver has both EHR and the medical image of the patient. Integrity checking is done by taking the following steps:

(1) ICC is calculated

(2) If $ICC = ICC^{\sim}$ then the received watermarked image and EHR are both valid

(3) If $ICC \neq ICC^{\sim}$ then the received watermarked image is rejected

(4) Inverse IWT is applied to the received watermarked image to recover the patient's medical image

5. Results and Discussion

In this section, a comprehensive investigation has been made on the proposed method in terms of performance and security analysis. The experiments have been carried out using MATLAB R2019a [29] on a Windows 10 PC with Intel® Core™ i7-2670QM CPU and 4 GB RAM. Various metrics such as Peak Signal to Noise Ratio (PSNR), Structural Similarity Index Matrix (SSIM), Mean Square Error (MSE), Normalized Cross-correlation (NCC), Mean Absolute Error (MAE), and Bit Error Rate (BER) are measured to evaluate the performance of the proposed method. PSNR and SSIM are used to measure the quality of an image while MSE, NCC, MAE, and BER are used to evaluate the error in extracted secret data bits. All experimental results are reported for both grayscale and color medical/general test images.

5.1. Metrics Explanation. A brief explanation and mathematical definition for each criterion is given in the following.

5.1.1. Peak Signal to Noise Ratio (PSNR). PSNR is a measure of the ratio of signal to noise power. This criterion is used to evaluate the imperceptibility of a watermarked image. PSNR is defined by the following equation [30]:

$$\text{PSNR} = 10\log_{10}\frac{255^2}{\text{MSE}}. \tag{7}$$

5.1.2. Structural Similarity Index Matrix (SSIM). SSIM is a measure for evaluating the similarity of two images related to perceptual quality in terms of the human visual system. SSIM is defined by the following equation:

$$\text{SSIM}(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)}, \tag{8}$$

where μ_X and μ_Y are averages of pixels values of X and Y, respectively. Moreover, σ_X^2 and σ_Y^2 are variances of X and Y and σ_{XY} is the covariance of X and Y. $C_1 = (K_1\ L)^2$ and $C_2 = (K_2 L)^2$ are two variables to stabilize the division with a weak denominator. Variable L is the dynamic range of pixel values and variables K_1 and K_2 are 0.01 and 0.03 by default [31].

5.1.3. Normalized Cross-Correlation (NCC). NCC measures the degree of similarity between the cover image and the watermarked image [32]. The range of NCC is [1, −1]. When two images are completely the same, NCC is equal to 1, and NCC will be −1 if they are completely opposite. If NCC is equal to 0, this means two images are uncorrelated. Assume $X(i, j)$ and $Y(i, j)$ represent a cover image and a watermarked image, respectively. NCC is defined by the following equation:

$$\text{NCC} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[X(i,j) - \mu_X][Y(i,j) - \mu_Y]}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[(X(i,j) - \mu_X)^2\right]}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[(Y(i,j) - \mu_Y)^2\right]}}, \tag{9}$$

where μ_X and μ_Y are averages of matrices X and Y, respectively.

5.1.4. Mean Absolute Error (MAE). This criterion measures the average absolute errors between the cover image and the watermarked image. Assuming X and Y are cover image and watermarked image, respectively, for a grayscale image, MAE is defined by the following equation [33]:

$$\text{MAE} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}|X(i,j) - Y(i,j)|}{M \times N}. \tag{10}$$

5.1.5. Bit Error Ratio (BER). When data is transmitting over a communication channel, during transmission data could get damaged or altered intentionally by an attacker or unintentionally by noise. The number of damaged bits is divided by the total number of transmitted bits to calculate the ratio of bit error. BER is defined by the following equation [34]:

$$\text{BER} = \frac{n_e}{n_s}, \tag{11}$$

where n_e is the number of damaged bits and n_s is the total number of transmitted bits.

5.2. Imperceptibility Analysis. The changes that are made to cover image must be imperceptible for the human visual system (HVS). A watermark is completely imperceptible if humans cannot distinguish the original image from the watermarked image when they are laid side by side [35].

The proposed method has been tested on various color and grayscale medical and commonly used images with a size of 512 × 512 pixels. All testing images and corresponding watermarked images are shown in Figure 7. The proposed method has been simulated for two modes, such that two and three LSBs of each coefficient of the cover image are

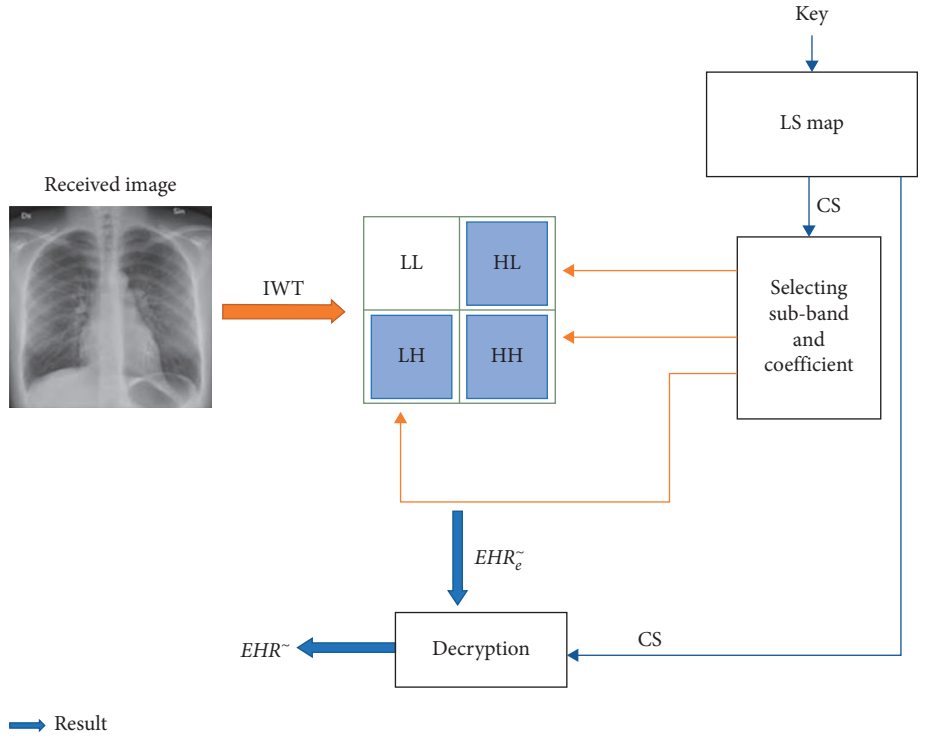


Figure 5: Extracting EHR from the watermarked image.

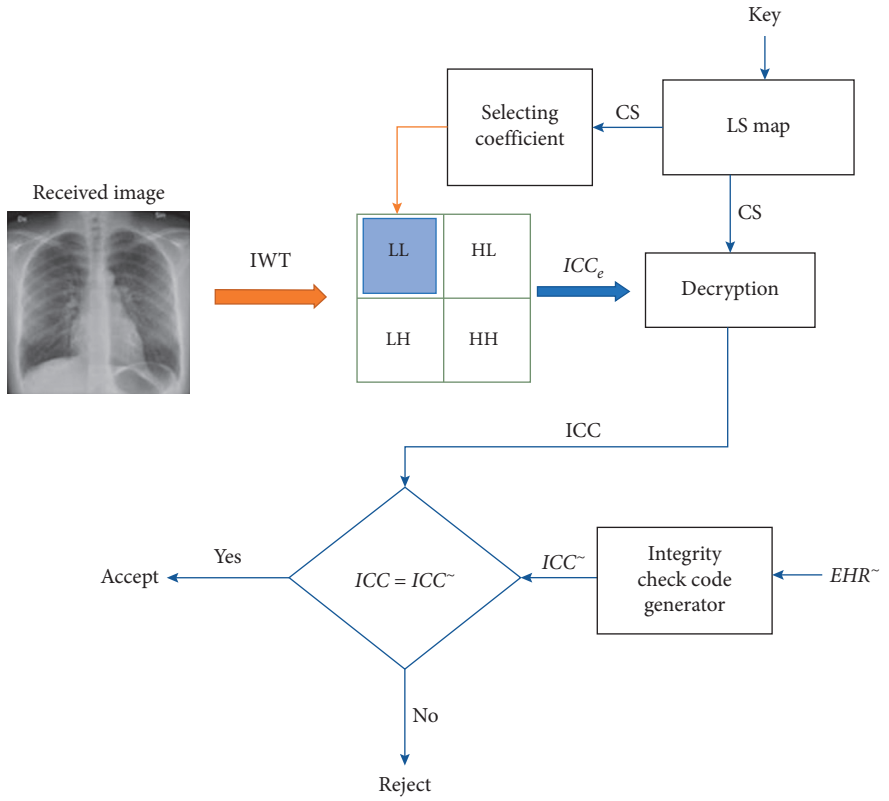


Figure 6: Integrity check test.

selected for embedding secret message in first and second mode, respectively. For grayscale images, the capacities are 432,538 bits (393,216 bits for EHR_e and 39,322 bits for ICC_e) and 648,808 bits (589,824 bits for EHR_e and 58,984 bits for ICC_e), respectively, for the first and second mode. Color images consist of three layers, Red, Green, and Blue (RGB);

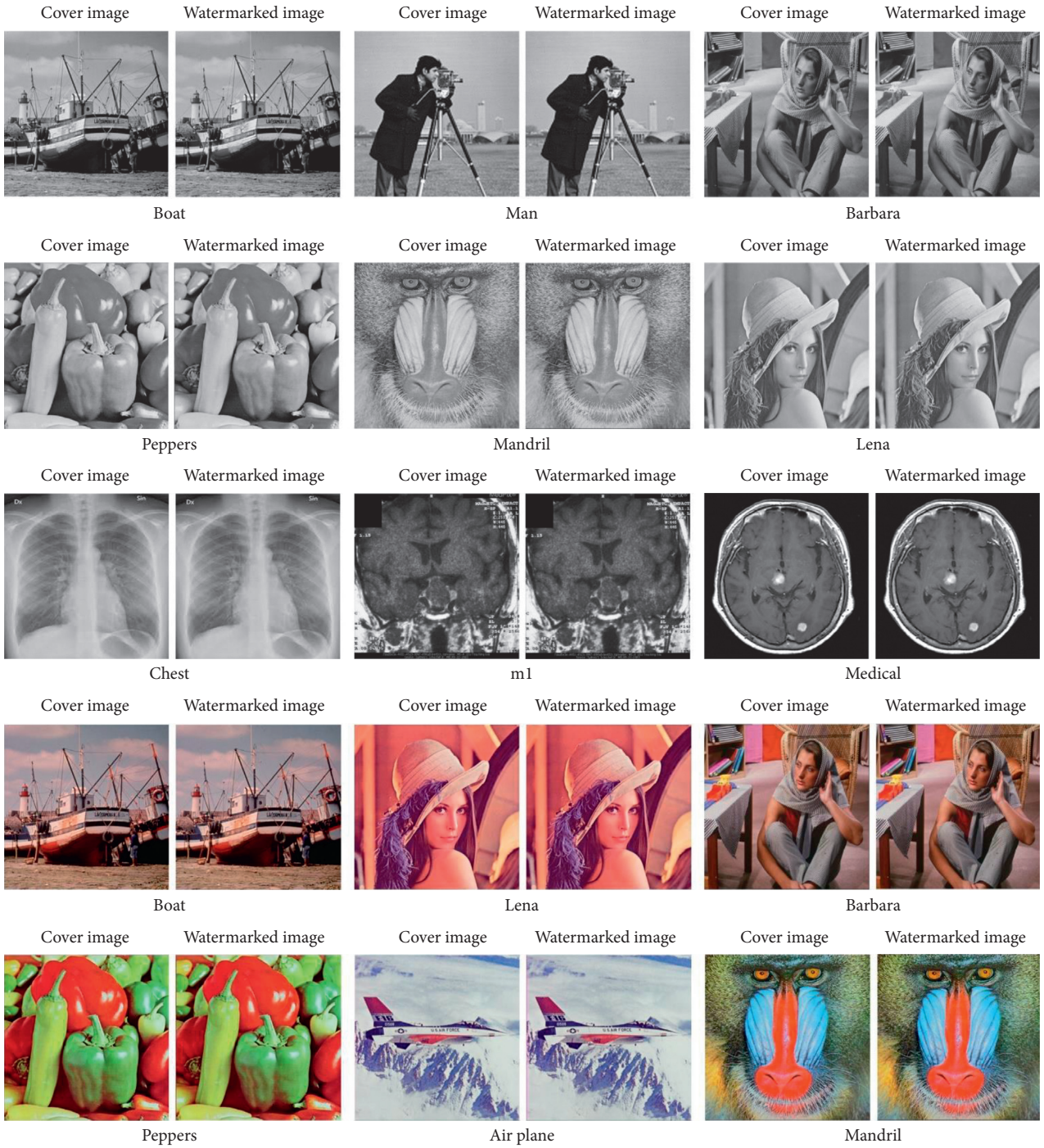


Figure 7: Test and watermarked images.

thus the capacities are 1,297,613 bits (1,179,648 bits for EHR_e and 117,965 bits for ICC_e) and 1,946,424 bits (1,769,472 bits for EHR_e and 176,952 for ICC_e), respectively. Since the length of ICC_e is equal to (1/10) of the length of EHR_e, the full capacity of LL is not occupied and could be used for other purposes. The results of PSNR, SSIM, NCC, and MAE metrics are reported for payloads of 432,538 and 1,297,613 bits (first mode) for grayscale and color images in Tables 2 and 3 , respectively.

PSNR values above 45 dB, MAE values close to zero, and NCC values close to 1 besides outstanding SSIM results indicate that the proposed method is capable of providing imperceptible and high quality watermarked images. PSNR, SSIM, NCC, and MAE results for second mode are given in Tables 4 and 5 for grayscale and color images, respectively. In second mode, the capacity for grayscale images is 648,808 bits (589,824 bits for EHR_e and 58,984 bits for ICC_e) and for color images the capacity is 1,946,424 bits (1,769,472 bits for

Table 2: Criterions results for grayscale images (432,538 bits' payload).

Cover image	PSNR	SSIM	MSE	NAE	NCC
Lena	45.41571356	0.98529705	0.93436050	0.00377006	0.99959233
Peppers	45.40055233	0.98616943	0.94027328	0.00388811	0.99967717
Mandrill	45.37948723	0.99539030	0.94287109	0.00371620	0.99950085
Barbara	45.37739511	0.98971939	0.94022369	0.00401877	0.99968409
Boat	45.28194767	0.98932682	0.96976089	0.00363709	0.99955803
Man	45.40891853	0.98356864	0.94287109	0.00392399	0.99975098
Chest	45.39985440	0.97885607	0.93158340	0.00357256	0.99944428
Medical	44.93731418	0.97929658	0.66186523	0.01098185	0.99970931
M1	45.35225447	0.98161973	1.01231002	0.00653188	0.99968356

Table 3: Criterions results for color images (1,297,613 bits' payload).

Cover image	PSNR	SSIM	MSE	NAE	NCC
Lena	45.39667538	0.99919942	0.31209437	0.00393097	0.99948778
Peppers	45.31623951	0.99911326	0.33829159	0.00450784	0.99962612
Barbara	45.40010469	0.99749971	0.31376139	0.00436215	0.99963265
Boat	45.35461716	0.99125248	0.33135774	0.00368493	0.99972065
Mandrill	45.39058876	0.99861759	0.31411319	0.00372696	0.99966761
Airplane	45.36217708	0.98705524	0.31885867	0.00256708	0.99942052

Table 4: Criterions results for grayscale images (648,808 bits' payload).

Cover image	PSNR	SSIM	MSE	NAE	NCC
Lena	39.16776366	0.94241805	3.92137145	0.00795723	0.99828481
Peppers	39.15785072	0.94528910	3.96346664	0.00817899	0.99864270
Mandrill	39.16214677	0.98142495	3.93774414	0.00781284	0.99791707
Barbara	39.17000866	0.95986360	3.92034149	0.00839860	0.99868275
Boat	39.15454443	0.95880376	3.83931350	0.00778576	0.99819215
Man	39.15081530	0.93682292	3.96741104	0.00830203	0.99894934
Chest	39.16677934	0.92011517	3.88083267	0.00753641	0.99767058
Medical	38.38476078	0.90927220	2.82612609	0.02433453	0.99871588
M1	39.02895827	0.93090727	4.45757293	0.01355195	0.99864375

Table 5: Criterions results for color images (1,946,424 bits' payload).

Cover image	PSNR	SSIM	MSE	NAE	NCC
Lena	39.16788707	0.99666985	1.30826441	0.00825880	0.99785790
Peppers	39.04715772	0.99621930	1.45020294	0.00942305	0.99842121
Barbara	39.17016370	0.98975923	1.31314341	0.00916486	0.99846046
Boat	39.08809440	0.96543779	1.43398072	0.00762411	0.99881913
Mandrill	39.15882182	0.99425752	1.31775029	0.00783808	0.99860516
Airplane	39.15136430	0.94956322	1.33884726	0.00535757	0.99759740

EHR_e and 176,952 bits for ICC_e). Similar to first mode, full capacity of LL subband is not occupied; thus, it could be used for other purposes.

The proposed method has been compared to several state-of-the-art techniques and results are reported in Tables 6 and 7 to show that the proposed method outperforms other schemes under comparison. Although the capacity of the proposed method is higher compared to other techniques, the results are showing significantly better performance of the proposed scheme in terms of PSNR, MAE, NCC, and SSIM. This is because of taking advantage of IWT which is a lossless transform and does not deteriorate cover image at all. In Figure 8, PSNR metric for different test images is compared to the other state-of-the-art methods and BPP is abbreviation of Bits Per Pixel.

5.3. Reversibility. As already mentioned, reversibility is a requirement for IoT based healthcare system. Therefore, we proposed a reversible technique in this paper. Usually, the capacity of irreversible techniques is more in comparison to the reversible techniques. On the other side, increasing capacity decreases imperceptibility. Thus, we tried to establish a trade-off between capacity and imperceptibility while providing reversibility in the proposed scheme. The procedure of extracting secret data from the cover image and

Table 6: PSNR, SSIM, NCC, and NAE criterions comparison.

Cover image	[10] payload: 327,680 bits				The proposed method payload: 432,538 bits			
	PSNR	SSIM	NCC	NAE	PSNR	SSIM	NCC	NAE
Lena	43.8838	0.97572	1	0.0100	45.4157	0.98529	1	0.0037
Mandrill	43.9171	0.98761	1	0.0098	45.3794	0.99539	1	0.0037
Peppers	43.8796	0.97559	1	0.0104	45.4005	0.98616	1	0.0038
Chest	43.8909	0.96547	1	0.0088	45.3998	0.97885	1	0.0035
Average	43.8935	0.97964	1	0.0100	45.3985	0.98894	1	0.0037

Table 7: Comparing the proposed method to the other state-of-the-art schemes.

Cover image	Method	Capacity	PSNR (dB)
Lena	[36]	196,608	46.3661
	[10]	327,680	43.8838
	Proposed	432,538	45.4157
Mandrill	[36]	196,608	46.3725
	[10]	327,680	43.9171
	Proposed	432,538	45.3794
Chest	[37]	10,882	48.4208
	[38]	36,060	48.9464
	[39] (GA scheme)	38,700	49.0119
	[39] (PSO scheme)	38,390	49.0047
	[36]	196,608	46.3685
	[10]	327,680	43.8909
	Proposed	432,538	45.3998

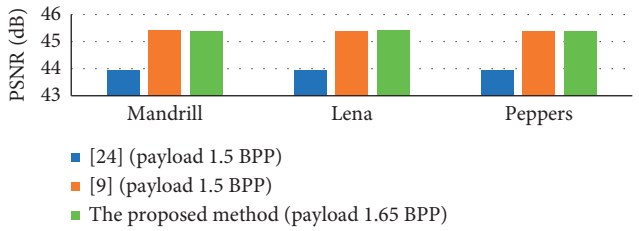


Figure 8: PSNR comparison.

recovering the patient's medical image has been described in Section 4.2 step by step. Test cover images and their corresponding watermarked images in Figure 7 are almost identical, and results of criterions such as SSIM, NCC, and MAE for these images prove their similarity. Because of close similarity, the watermarked image is reusable on the receiver side completely. Since LSBs of the cover image are used for embedding into, high quality of the patient's medical image is preserved.

5.4. Computational Complexity Analysis. As already mentioned, computational complexity is an important factor for IoT based healthcare system. One of the most important features of the proposed method is a low computational overhead which leads to less time and energy consumption. The cover image generation and data embedding stages of the proposed method are compared to cover image generation and data embedding of other state-of-the-art techniques in Figures 9 and 10, respectively. Cover image preparation is considerably faster in the proposed method

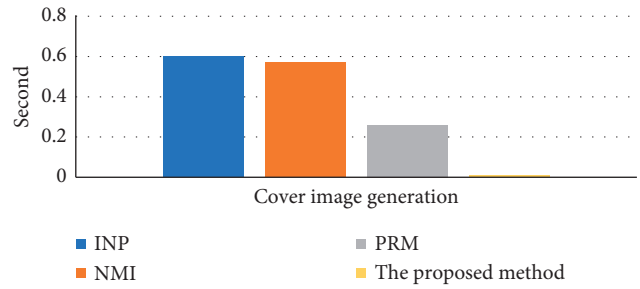


Figure 9: Required time comparison for CIG.

comparing to other schemes, because of taking advantage of IWT instead of calculating several equations in other state-of-the-art techniques. Using various algebraic equations for generating a cover image from the original image is more time consuming and complex than IWT. The data embedding stage of the proposed method is done by embedding secret data directly into LSBs of cover image pixels which does not require any calculation; therefore it is faster in comparison to other state-of-the-art schemes.

5.5. Security Analysis. One of the main goals of this paper is to address security challenges. For this purpose, several security levels are designed and included in the proposed method. In summary, the following steps are taken to ensure the proposed method meets security requirements:

(1) Choosing the order of pixels for embedding is based on a chaotic sequence (CS) which is impossible to regenerate without knowing the agreed key between the sender and the receiver.

(2) Secret data (EHR) is first encrypted and then embedded into the cover image.

(3) A fragile watermark is calculated and encrypted and then embedded into the cover image in order to detect any tampering or altering.

(4) Both encrypted EHR and the watermark are embedded into the transformed layer of the original image using IWT which brings noticeably more resistance against different attacks.

The proposed scheme has been tested against various signal processing and geometric attacks and BER (%) results are presented in Table 8 for payload of 432,538 bits and 1,297,613 bits for grayscale and color images, respectively. Moreover, obtained security analysis results are compared to [10] in Figure 11. It is evident from Figure 11 that the proposed

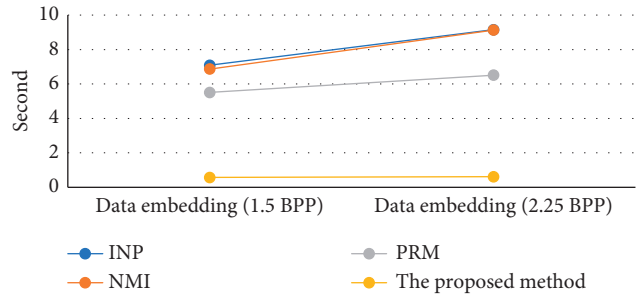


Figure 10: Required time for DE.

Table 8: BER results after performing various attacks for payload of 432,538 and 1,297,613 bits.

Attack	Lena (grayscale)	Mandrill (grayscale)	Lena (color)	Mandrill (color)	Tampering
Salt and pepper (0.1)	0.055	0.085	0.104	0.089	Detected
Gaussian noise (0.02)	0.277	0.165	4.893	2.592	Detected
Median filter	0.328	0.256	0.202	0.245	Detected
Low pass filtering	0.035	0.008	0.214	0.610	Detected
Sharping	0.059	0.236	0.199	0.233	Detected
Histogram equalization	0.041	0.128	0.084	0.215	Detected

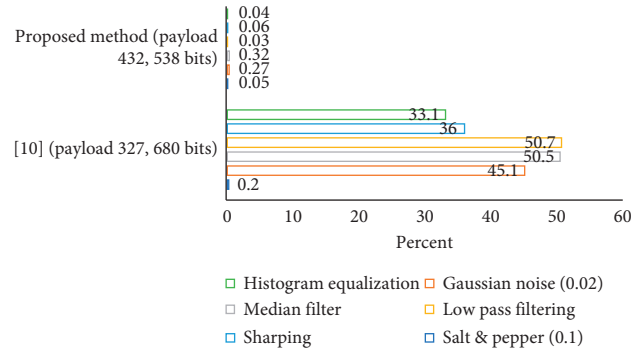


Figure 11: BER comparison for lena.

Table 9: Comparison of the main features of the proposed method and the other state-of-the-art schemes.

Method	Imperceptibility	Capacity	Computational cost	Tamper detection
[40]	Low	Medium	Medium	No
[39]	Low	Low	High	No
[36]	High	Low	Low	Yes
[10]	High	High	Low	Yes
[9]	High	High	Low	No
Proposed	High	High	Low	Yes

method, because of taking advantage of IWT, has impressive resistance against various attacks. Tamper detection analysis is also included in Table 8 to demonstrate the functionality of the fragile watermark. The results prove that the fragile watermark can detect any kind of tampering and altering. All security tests have been done for both grayscale and color images.

In Table 9, the proposed method is compared to other state-of-the-art schemes in terms of the main features.

5.6. Histogram Analysis. Histogram analysis is performed on watermarked images by adversaries with the intension to find a clue about what has been embedded. Histogram analysis is done by comparing the histograms of the cover image and watermarked image with each other. To withstand this attack, histograms of the cover image and watermarked image must be similar to each other closely. In Figure 12 histograms of various cover images (12(a), 12(c),

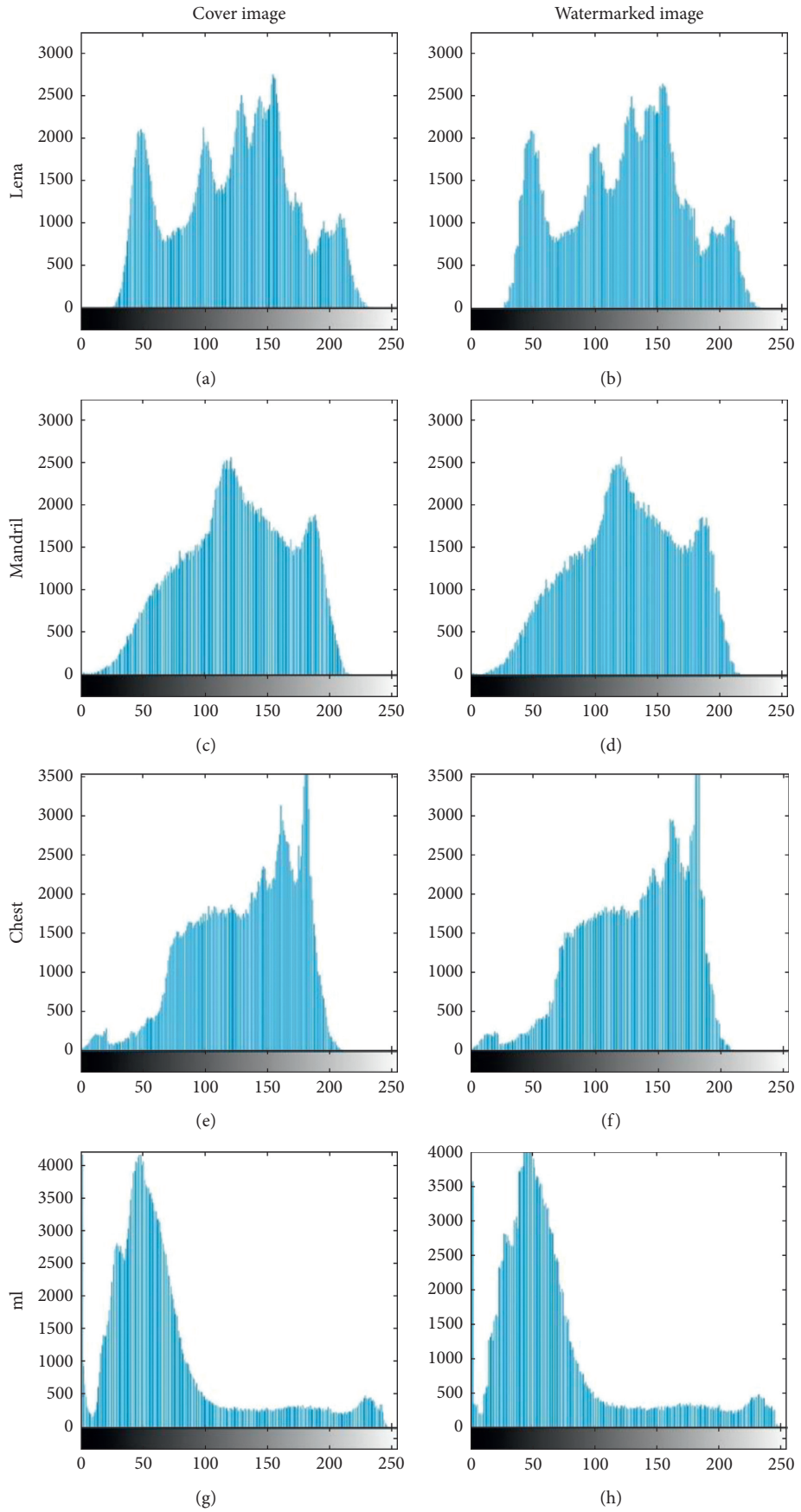


Figure 12: Histograms of cover and watermarked images.

12(e), and 12(g)) and corresponding watermarked images (12(b), 12(d), 12(f), and 12(h)) are presented for a payload of 432,538 bits. It is evident from Figure 12 that the proposed method can resist this attack since the histograms are nearly the same.

6. Conclusion

In this paper, considering security and IoT requirements, a secure, reversible, lightweight watermarking method with the capability of tamper detection has been introduced for IoT based healthcare systems. Usually, the capacity of reversible techniques is less than irreversible ones. Thus, we tried to establish a trade-off between capacity and preserving image quality to introduce a high capacity reversible scheme. In the proposed method for preparing a cover image, IWT was employed which is a fast and lossless transform with low computational complexity. EHR firstly is encrypted by a chaotic sequence and then embedded into LSBs of IWT subbands coefficients. A fragile watermark is calculated and embedded for tamper detection capability. Comprehensive investigations and analyses that have been made to the experimental results demonstrate high performance in terms of imperceptibility and image quality. The proposed method reduces the computational complexity significantly in comparison to the other state-of-the-art techniques by taking advantage of IWT. Security analyses in Section 5.5 prove that the proposed method is noticeably resistant against various signal processing attacks and the tamper detection feature is working properly.

Data Availability

All of the image samples and underlying data that support the results of our study are given in the paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Special thanks are due to Iman Dorostkar Ahmadi who helped us by implementing this work.

References

[1] H. Ahmadi, G. Arji, L. Shahmoradi, R. Safdari, M. Nilashi, and M. Alizadeh, "The application of internet of things in healthcare: a systematic literature review and classification," *Universal Access in the Information Society*, vol. 18, no. 4, pp. 837–869, 2019.

[2] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: techniques, Evaluations, and trends in future research," *Neurocomputing*, vol. 335, pp. 299–326, 2019.

[3] Y. Qiu, H. Duan, J. Sun, and H. Gu, "Rich-information reversible watermarking scheme of vector maps," *Multimedia Tools and Applications*, vol. 78, no. 17, pp. 24955–24977, 2019.

[4] M. Turuk and A. Dhande, "A novel reversible multiple medical image watermarking for health information system," *Journal of Medical Systems*, vol. 40, no. 12, pp. 1–13, 2016.

[5] W. Wang and J. Zhao, "Hiding depth information in compressed 2D image/video using reversible watermarking," *Multimedia Tools and Applications*, vol. 75, no. 8, pp. 4285–4303, 2016.

[6] I. J. Kadhim, P. Premaratne, and P. J. Vial, "High capacity adaptive image steganography with cover region selection using dual-tree complex wavelet transform," *Cognitive Systems Research*, vol. 60, pp. 20–32, 2020.

[7] T. Rabie, M. Baziyad, and I. Kamel, "Enhanced high capacity image steganography using discrete wavelet transform and the Laplacian pyramid," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 23673–23698, 2018.

[8] D. K. Sarmah and A. J. Kulkarni, "Improved Cohort Intelligence-A high capacity, swift and secure approach on JPEG image steganography," *Journal of Information Security and Applications*, vol. 45, pp. 90–106, 2019.

[9] S. A. Parah, J. A. Sheikh, J. A. Akhoon, and N. A. Loan, "Electronic Health Record hiding in Images for smart city applications: a computationally efficient and reversible information hiding technique for secure communication," *Future Generation Computer Systems*, vol. 108, pp. 935–949, 2020.

[10] J. A. Kaw, N. A. Loan, S. A. Parah, K. Muhammad, J. A. Sheikh, and G. M. Bhat, "A reversible and secure patient information hiding system for IoT driven e-health," *International Journal of Information Management*, vol. 45, pp. 262–275, 2019.

[11] K.-H. Jung, "A survey of interpolation-based reversible data hiding methods," *Multimedia Tools and Applications*, vol. 77, no. 7, pp. 7795–7810, 2018.

[12] A. Malik, G. Sikka, and H. K. Verma, "Image interpolation based high capacity reversible data hiding scheme," *Multimedia Tools and Applications*, vol. 76, no. 22, pp. 24107–24123, 2017.

[13] A. A. Mohammad, A. Al-Haj, and M. Farfoura, "An improved capacity data hiding technique based on image interpolation," *Multimedia Tools and Applications*, vol. 78, no. 6, pp. 7181–7205, 2019.

[14] A. Shaik and V. Thanikaiselvan, "High capacity reversible data hiding using 2D parabolic interpolation," *Multimedia Tools and Applications*, vol. 78, no. 8, pp. 9717–9735, 2019.

[15] E. K. Naeini, I. Azimi, A. M. Rahmani, P. Liljeberg, and N. Dutt, "A real-time PPG quality assessment approach for healthcare Internet-of-Things," *Procedia Computer Science*, vol. 151, pp. 551–558, 2019.

[16] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, "A survey on internet of things security: requirements, challenges, and solutions," *Internet of Things*, vol. 2019, Article ID 100129, 2019.

[17] W. A. Kassab and K. A. Darabkh, "A-Z survey of Internet of Things: architectures, protocols, applications, recent advances, future directions and recommendations," *Journal of Network and Computer Applications*, vol. 163, Article ID 102663, 2020.

[18] C. C.-H. Hsu, M. Y.-C. Wang, H. C. Shen, R. H.-C. Chiang, and C. H. Wen, "FallCare+: an IoT surveillance system for fall detection," in *Proceedings of the 2017 International Conference on Applied System Innovation (ICASI)*, pp. 921-922, IEEE, Sapporo, Japan, May 2017.

[19] Y. Zhuang, "Query customization and trigger optimization on home care systems," in *Proceedings of the 2017 International*

Conference on Applied System Innovation (ICASI), pp. 668–671, IEEE, Sapporo, Japan, May 2017.

[20] R. Zgheib, R. Bastide, and E. Conchon, "A semantic web-of-things architecture for monitoring the risk of bedsores," in *Proceedings of the 2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 318–323, IEEE, Las Vegas, NV, USA, December 2015.

[21] D. Wilson, "An overview of the application of wearable technology to nursing practice," *Nursing Forum*, Wiley Online Library, vol. 52, no. 2, pp. 124–132, 2017.

[22] H. Sajedi, "Applications of data hiding techniques in medical and healthcare systems: a survey," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 7, no. 1, pp. 1–28, 2018.

[23] K.-H. Jung and K.-Y. Yoo, "Data hiding method using image interpolation," *Computer Standards & Interfaces*, vol. 31, no. 2, pp. 465–470, 2009.

[24] C.-F. Lee and Y.-L. Huang, "An efficient image interpolation increasing payload in reversible data hiding," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6712–6719, 2012.

[25] T. K. Araghi and A. A. Manaf, "An enhanced hybrid image watermarking scheme for security of medical and non-medical images based on DWT and 2-D SVD," *Future Generation Computer Systems*, vol. 101, pp. 1223–1246, 2019.

[26] N. Zermi, A. Khaldi, R. Kafi, F. Kahlessenane, and S. Euschi, "A DWT-SVD based robust digital watermarking for medical image security," *Forensic Science International*, vol. 320, Article ID 110691, 2021.

[27] V. Thanikaiselvan, P. Arulmozhivarman et al., "Comparative analysis of (5/3) and haar IVVT based steganography," *Information Technology Journal*, vol. 13, no. 16, pp. 2534–2543, 2014.

[28] Y. Zhou, L. Bao, and C. L. P. Chen, "A new 1D chaotic system for image encryption," *Signal Processing*, vol. 97, pp. 172–182, 2014.

[29] T. Jack, *MATLAB. 9.6.0.1072779 (R2019a)*, The MathWorks Inc., Natik, MA, USA, 2019.

[30] M. Saidi, H. Hermassi, R. Rhouma, and S. Belghith, "A new adaptive image steganography scheme based on DCT and chaotic map," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 13493–13510, 2017.

[31] M. Nazari and I. Dorostkar Ahmadi, "A novel chaotic steganography method with three approaches for color and grayscale images based on FIS and DCT with flexible capacity," *Multimedia Tools and Applications*, vol. 79, no. 19-20, pp. 13693–13724, 2020.

[32] K. Muhammad, J. Ahmad, N. U. Rehman, Z. Jan, and M. Sajjad, "CISSKA-LSB: color image steganography using stego key-directed adaptive LSB substitution method," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8597–8626, 2017.

[33] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. S. Ho, and K.-H. Jung, "Image steganography in spatial domain: a survey," *Signal Processing: Image Communication*, vol. 65, pp. 46–66, 2018.

[34] S. A. El_Rahman, "A comparative analysis of image steganography based on DCT algorithm and steganography tool to hide nuclear reactors confidential information," *Computers & Electrical Engineering*, vol. 70, pp. 380–399, 2018.

[35] M. Staring, "Analysis of quantization based watermarking," *Compare*, vol. 500, no. 2, pp. 3–15, 2002.

[36] S. A. Parah, F. Ahad, J. A. Sheikh, and G. M. Bhat, "Hiding clinical information in medical images: a new high capacity and reversible data hiding technique," *Journal of Biomedical Informatics*, vol. 66, pp. 214–230, 2017.

[37] S. Lee, C. D. Yoo, and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *Institute of Electrical and Electronics Engineers Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 321–330, 2007.

[38] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," *Institute of Electrical and Electronics Engineers Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, 2009.

[39] T. Naheed, I. Usman, T. M. Khan, A. H. Dar, and M. F. Shafique, "Intelligent reversible watermarking technique in medical images using GA and PSO," *Optik*, vol. 125, no. 11, pp. 2515–2525, 2014.

[40] J. Hu and T. Li, "Reversible steganography using extended image interpolation technique," *Computers & Electrical Engineering*, vol. 46, pp. 447–455, 2015.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5546334, 12 pages
https://doi.org/10.1155/2021/5546334

Research Article

New Efficient and Secured Authentication Protocol for Remote Healthcare Systems in Cloud-IoT

Mourade Azrour,[1] **Jamal Mabrouki**,[2] **and Rajasekhar Chaganti**[3]

[1]*Computer Sciences Department, Faculty of Sciences and Techniques, Moulay Ismail University, Errachidia, Morocco*
[2]*Laboratory of Spectroscopy, Molecular Modeling Materials Nanomaterial, Water and Environment, CERNE2D, Mohammed V University, Faculty of Science, Rabat, Morocco*
[3]*Expedia Group Inc, Seattle 98119, USA*

Correspondence should be addressed to Mourade Azrour; azrour.mourade@gmail.com

Received 11 March 2021; Accepted 24 April 2021; Published 8 May 2021

Academic Editor: Azeem Irshad

Recently, Internet of Things and cloud computing are known to be emerged technologies in digital evolution. The first one is a large network used to interconnect embedded devices, while the second one refers to the possibility of offering infrastructure that can be used from anywhere and anytime. Due to their ability to provide remote services, IoT and cloud computing are actually integrated in various areas especially in the healthcare domain. However, the user private data such as health data must be secured by enhancing the authentication methods. Recently, Sharma and Kalra projected an authentication scheme for distant healthcare service-based cloud-IoT. Then, authors demonstrated that the proposed scheme is secure against various attacks. However, we prove in this paper that Sharma and Kalra's protocol is prone to password guessing and smart card stolen attacks. Besides, we show that it has some security issues. For that reason, we propose an efficient and secured authentication scheme for remote healthcare systems in cloud-IoT. Then, we prove informally that our projected authentication scheme is secure against multiple attacks. Furthermore, the experimental tests done using Scyther tool show that our proposed scheme can withstand against known attacks as it ensures security requirements.

1. Introduction

The Internet of Things (IoT) and cloud computing are revolutionizing many industries such as health and transportation. The IoT is a large network that interconnects objects, computers, and human individuals. These devices are able to sense, process, and communicate data from one end to another one. In addition, cloud computing is a system that allows the customers to access computing resources via the network. The cloud computing provider ensures the protection of a given number of servers that can be used according to the customer needs. Indeed, IoT's growth has been particularly dynamic and has truly revolutionized human personal and professional daily activities. Some of the IoT areas include agricultural [1–4], industrial [5], education [6], healthcare [7–9], and environmental fields [10–13].

Remote patient monitoring relies on computer systems that retrieve health information from individuals in one location and communicate it digitally to health professionals in another location for assessment and advice, as shown in Figure 1. With this kind of service, healthcare provider can continue processing patient's medical data even if the patient stays at home or care facility and reduces the patient readmission rates.

The question of health is systematically at the heart of the human race's concern, even though there are technological advancements in health treatment. Recently, healthcare has taken on great significance, as witnessed by the most recent coronavirus epidemic. Indeed, in areas where the epidemic is spreading, it is increasingly wise to monitor people remotely using health monitoring technology. A variety of monitoring platforms which allow collecting a large volume of healthcare data at the site of treatment, such as patient's vital signs,

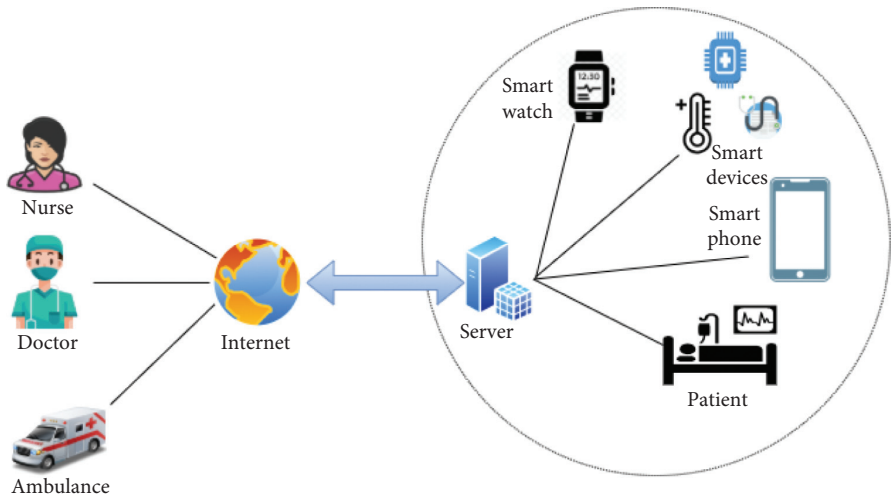


Figure 1: Internet of Thing and healthcare system.

patient's weight, heart rate, blood pressures, blood glucose, blood O2 levels, and electrocardiographs, can be used to monitor the patient's health.

Because of the performance and computing limitations of IoT equipment in handling the massive volume of collected data, it may be appropriate to employ a cloud service to overcome such challenges. Nevertheless, this approach will require warranty of confidentiality, integrity, and security of exchanged data. Subsequently, data are only accessible by authorized entities.

The commonly used solution to assure the confidentiality of data is using such authentication protocols. Hence, Sharma and Kalra [14] proposed a trivial authentication protocol for cloud-IoT-based healthcare system. Formerly, they demonstrated that their proposed protocol is efficient, trivial, and secured against various attacks including DoS attack, man in the middle attack, offline password guessing attack, user impersonation attack, replay attack, and parallel session attack. Authors also use AVISPA tool to evaluate their protocol formally. In this research work, we demonstrated that Sharma and Kalra's scheme poses security vulnerabilities, in particular vulnerable to password guessing attacks. Moreover, some private values described in the protocol are not very secured; it can be obtained basically by any attacker. With the aim to improve the security of cloud-IoT-based healthcare, we propose a new efficient and secured authentication protocol. After proving theoretically that our protocol can resist against various attack, we have done simulation tests under Scyther tool. The obtained results confirm that our scheme can deal against famous attacks, and it guarantees security requirements.

The remaining part of the paper is organized as follows. Related works have been debated in Section 2. Section 3 is reserved for reviewing Sharma and Kalra's protocol. In Section 4, weaknesses of Sharma and Kalra's protocol are discussed. Our proposed efficient and secured authentication scheme is presented in Section 5. In Section 6, informal and formal analyses are detailed. Section 7 provides the performance and comparative analysis. In Section 8, we conclude our paper.

2. Related Works

Due to the quick growth and development of various new technologies, personal security, the system for controlling access and the procedures for checking the authenticity of data are gaining significance everyday, particularly since the birth of the IoT. As a consequence, in this section, we discuss some authentication protocols that have been previously presented in literature.

Watro et al. [15] proposed a secured authentication scheme based on RSA for WSNs. Wong et al. [16] proposed authentication scheme that is funded on one way hash functions. This protocol was considered to be secure against many possible attacks, including replay attack, man-in-the-middle attack, forgery attack, and key impersonation attack. Nonetheless, this protocol is proved prone to insider attack and man-in-the-middle attack. As result, Das et al. [17] planned an enhance protocol for gaining more security. Moreover, Xu et al. [18] and Song [19] proposed independently two authentication protocols in 2009 and 2010, respectively. The two proposed protocols are both based on RSA cryptography.

Based on elliptic curve cryptography (ECC), Xu et al. [20] proposed mutual authentication and key convention scheme as a solution of computational problem. Then, he demonstrated that the proposed protocol guarantees confidentiality by using a dynamic identity. Hence, Yan et al. [21] proposed a user authentication system based on biometric detection. However, this protocol cannot resist against replay attacks and is not able to guarantee user anonymity. Furthermore, Mishra et al. proved that Yan's protocol is vulnerable against offline password guessing attacks. Based on those issues, Mishra et al. [22] suggested a new reinforced biometric authentication protocol that uses random digits. Afterword, Tan [23] proposed three-factor mutual authentication protocol.

Yoon and Kim [24] presented user authentication protocol based on a biometric parameter to enhance security of wireless sensors' networks. The proposed scheme is demonstrated secure against some attacks such as DoS attack and sensor impersonation attack.

In 2012, He et al. [25] proposed an authentication protocol, which is efficient for actual medical applications that are based on sensor network. Nevertheless, the scheme is prone to forgery attack and password guessing attack. In addition, it is not capable to offer forward privacy service. In 2014, Mishra et al. [26] rely on chaotic map computation for presenting an authentication and key exchanging protocol for healthcare information organisms. However, this scheme is vulnerable to against password guessing attack.

In 2015, Jiang et al. [27] proved that the protocol proposed by Chen et al. [28] is not secured against password guessing attack. Consequently, with the goal to resolve this issue, Jiang et al. projected a different authentication scheme. Nonetheless, the planned solution is still vulnerable to password guessing and user impersonation attack.

In 2019, Azrour et al. [29] demonstrated that Ye et al.'s [30] protocol is not secured and it has some security issues. In the same year, Cheng et al. [31], based on elliptical curve cryptography and biometrics, proposed a public node identity authentication scheme for numerous categories of devices. In 2020, Azrour et al. [32] proposed a new authentication protocol for IoT devices. Then, authors proved formally and informally that their protocol is efficient and can resist against several attacks.

3. Evaluation of Sharma and Kalra's Protocol

In the present section, we present a brief review of three main phases of Sharma and Kalra's scheme, namely, registration, login, and authentication phases. Used notations and their significations are described in Table 1.

3.1. Registration Phase

Step 1: user U_i selects her/his identity **I d**, password **pw**, and arbitrary number **R**. He/she computes $\mathbf{MPS} = \mathbf{h}(\mathbf{pw} \| \mathbf{R})$ and sends $\langle \mathbf{I\ d}, \mathbf{MPS} \rangle$to the server via secured channel.

Step 2: server checks received Id. If it exists in database, the server requests a new Id. In other case, the server computes $= H(\text{MPS}\|\text{Id})$, $b = A\ (\text{Id}\|K)$, $c = H(K) \oplus H(\text{MPS}\|b)$, and $d = b \oplus H(\text{MPS}\|a)$. Afterwards, server sends back to user a, c, and d.

Step 3: user saves $\langle a, c, d, R \rangle$ in it smart card.

3.2. Login Phase.

Sharma and Kalra's scheme login phase contains three steps:

Step 1: user U_i inserts her/his identity **I d** and password **pw** in the smart device.

Step 2: the smart device calculates $\mathbf{MPS} = \mathbf{h}(\mathbf{pw} \| \mathbf{R})$based on input pw and stored R.

Step 3: the smart device computes $a' = H(\text{MPS} \| \text{Id})$ and compares its value with stored a. In this case, it equals the login phase which is a success.

Table 1: Notations and their significations.

Symbol	Signification
U_i	User (medical professional)
Id_i	User's U_i identity
pw	User's U_i password
SN_i	The sensor node
GN	Gateway node
Id_{sn}	Identity of sensor node
CS	Cloud server
X_s	Secret key of CS
$K_{\text{CS-SN}i}$	Shared session key between CS and SN_i
$T1, T2, T3, T4$	The current time
A, B, C, D	High entropy random numbers
h	Hash function
$\oplus$	XOR operator
$\|\|$	Concatenation operator

3.3. Authentication Phase.

In this phase, the user U_i, the sensor, and the gateway node have to authenticate each other mutually and produce the session key. The steps of this phase are

Step 1: the smart device calculates $b = d \oplus H(\text{MPS}\|a)$ and $H(K) = c \oplus H(\text{MPS} \|b)$. It generates timestamp T_1 and computes $V_1 = \text{Id} \oplus H(H(K)\|T_1)$. Then, it chooses random Ni and computes $V_2 = N \oplus H(b\| T_1)$ and $V_3 = H(V_1\|V_2\| Ni \|T_1)$. Next, it transfers this message to gateway node (GN) $\langle V_1, V_2, V_3, T_1, \text{Id}_{\text{SN}} \rangle$.

Step 2: after receiving user's massage, the GN verifies the timestamp. Then, it calculates $\text{MID}_{\text{SN}} = \text{Id}_{\text{SN}} \oplus H(H(K\|z_2))$. It generates random numberNj, which is used for computing $V_4 = H(X_{\text{GN-SN}} \| T_1\|T_2) \oplus Nj$ and $V_5 = H(\text{Id}_{\text{SN}}\|V_4\|T_1 \|T_2\|Nj)$. The GN then communicates this message $\langle V_1, V_2, V_3, T_1, T_2, \text{MID}_{\text{SN}}, V_4, V_5 \rangle$ to the SN.

Step 3: upon receiving GN message, the SN verifies the timestamp. Then, it calculates $\text{Id}_{\text{SN}}' = \text{MId}_{\text{SN}} \oplus H$ $(H(K\|T_2))$, $X'_{\text{GN-SN}} = (\text{Id}_{\text{SN}})$, and $Nj' = V_4 \oplus H(X'_{\text{GN-SN}} \|T_1\|T_2)$. Formerly, it checks $V_5' = H(\text{Id}_{\text{SN}}\|V_4\|T_1 \| T_2\|Nj')$. If it is correct, the GN is authenticated. Next, SN computes $\text{Id}' = V_1 \oplus H(H(K\|T_1))$, $b' = H(\text{Id}\|K)$, and $Ni' = V_2 \oplus H(b'\|T_1)$. Then, it checks the validity of $V_3' = H(V_1\|V_2\|Ni'\|T_1)$. If it is ok, the user was authenticated. Therefore, the SN computes its parameters V_6, V_7, V_8, and V_9 that will be sent to GN. $V_6 = Nj' \oplus H(b'\|T_3)$, $V_7 = Ni' \oplus H(X'_{\text{GN-SN}}\|T_3)$, $V_8 = H(V_6\| b'\|T_3)$, and $V_9 = H(V_7\|X'_{\text{GN-SN}}\|T_3)$.

Step 4: once the GN receives SN response, it verifies the timestamp. Then, it checks the correctness of $V_9' = H(V_7\|X'_{\text{GN-SN}}\| T_3)$. It computes $\prime = V_7 \oplus H$ $(X'_{\text{GN-SN}} \|T_3)$, $\text{Sk} = H(Ni' \oplus Nj)$, and $V_{10} = H(\text{Sk}$

$\| V_6 \| V_8 \| T_3 \| T_4$). Next, the GN sends to the user this message: $\langle V_6, V_8, V_{10}, T_3, T_4 \rangle$.

Step 5: in this step, the user verifies the validity of timestamp. Then, he/she checks the authenticity of $V_8' = H(V_6 \| b' \| T_3)$ and calculates $Nj' = V_6 \oplus H(b' \| T_3)$ and $\mathrm{Sk} = H(Ni \oplus Nj')$. Finally, the user checks the correctness of $V_{10}' = H(\mathrm{Sk} \| V_6 \| V_8 \| T_3 \| T_4)$.

4. Weaknesses of Sharma and Kalra's Protocol

In this section, we demonstrate that user authentication scheme for cloud-IoT-based healthcare services suggested by Sharma and Kalra is defenceless against offline password guessing attacks, and it has some security issues.

4.1. User Password Guessing Attack. Sharma and Kalra proved that their protocol can resist against offline password guessing attack even if the smart card is stolen. In opposition to this, we can prove her that an adversary can guess user's password. To do that, he/she has to steal the user's smart card and then recover the value of R_1 and a_i. Afterword, the adversary runs the dictionary attack to guess the correct password. As it is clear in Figure 2, adversary selects a guessed password from passwords dictionary. Then, he/she computes the value of $\mathrm{MPS}'' \leftarrow h(\mathrm{pw} \| R_1)$ and the value of $h(\mathrm{MPS}'' \| \mathrm{Id}_i)$; if the second value equals a_i, the guessed password is correct. Otherwise, the adversary selects another password until discovering the correct one.

4.2. Impersonation Attack. Sharma and Kalra demonstrated that their proposed protocol can resist against numerous attacks including impersonation attack. However, in this section, we demonstrate that the user impersonation attack is still operative in Sharma and Kalra's authentication scheme. Accept that an adversary has obtained the contents of a smartcard. He/she can execute the pervious attack to get user's parameters (login and password); then, he/she makes a forged authentication request.

The pirate inserts stolen smart card. Next, he/she enters the deduced ID' and Pw'. Subsequent, the smart device computes the values of $\mathrm{MPS}' = h(\mathrm{Pw}' \| R)$ and $a' = H(\mathrm{MPS}' \| \mathrm{Id}')$. Then, it verifies if $a' \stackrel{?}{=} a$. The equality will be true because the guessed parameters are verified in the previous attack. Afterward, the smart device will execute the authentication phase. It computes $b' = d \oplus H(\mathrm{MPS}' \| a')$ and $H(K) = c \oplus H(\mathrm{MPS}' \| b')$. It generates timestamp T_1 and computes $V_1 = \mathrm{Id} \oplus H(H(K) \| T_1)$. Then, it chooses random Ni and computes $V_2 = N \oplus H(b \| T_1)$ and $V_3 = H(V_1 \| V_2 \| Ni \| T_1)$. Next, it transfers this message to gateway node (GN) $< \langle V_1, V_2, V_3, T_1, \mathrm{Id}_{\mathrm{SN}} \rangle$.

The remainder of the protocol will run normally. The gateway node and sensor node will authenticate the pirate as the valid user and then share with them the session key successfully. Therefore, the pirate can execute user impersonation attack successfully if he has stolen the smart card contents.

Algorithm1 offline_PW_Guessing

```
Begin
  input : Id_i, R_1, a_i, D_PW (Password_dictionary)
  output : PW

for pw' in D_PW do
   MPS'' ← h (pw'||R_1)
   if a_i = h (MPS''||Id_i) then
        return pw'
   end if
end for
end.
```

Figure 2: Algorithm for guessing password offline.

4.3. Other Security Issues. Authentication phase is an essential and main phase in all authentication protocols. Indeed, it is a step that assures verification of user identity, allowing authorization and constructing session keys. In healthcare field, it is a key for establishing a secured connection with the remote server and for protecting patient private data. Therefore, it must receive much importance. In our cryptanalysis of Sharma and Kalra authentication protocol for cloud-IoT-based healthcare service, we have observed that there are double serious mistakes. Initially, the sensor node utilizes value of K which is the secret master key of gateway node (GN). Normally, the secret master key of gateway node is a private key. Accordingly, it must be a secret and should not be known to anyone. Otherwise, all systems are at risk and all secret messages will be discovered by any attacker. Secondly, authors propose that the secret shared between the gateway node and sensor node (SN_i) is $X_{\mathrm{GN-SN}_i}$ which is computed as $X_{\mathrm{GN-SN}_i} = h(\mathrm{Id}_{\mathrm{SN}_i})$. However, the value of $\mathrm{Id}_{\mathrm{SN}_i}$ is clearly (not encrypted or hashed) in the first message sent by user (U_i) to gateway node (GN). Consequently, if an adversary intercepts this message, he can get easily $\mathrm{Id}_{\mathrm{SN}_i}$. Then, he is able to compute $X_{\mathrm{GN-SN}_i}$.

5. Our Proposed Scheme

In this section, we present our new efficient and secured authentication protocol for remote healthcare systems in cloud-IoT. The proposed scheme entails five phases, including system setup phase, new sensor registration phase, user registration phase, login and authentication phase, and password changing phase.

5.1. System Setup Phase. In this phase, the superadmin chooses secret key of cloud server X_s and one way hash function h. Finally, the server publishes h and saves its private key secretly.

5.2. New Sensor Registration Phase. To implement a newly connected sensor node (SN_i) in an already functioning healthcare system, the cloud server has to randomly select both specific $\mathrm{Id}_{\mathrm{sn}}$ and $K_{\mathrm{CS-SN}i}$ as the identification and unique key of the added sensor, respectively. The gateway subsequently uploads $\mathrm{Id}_{\mathrm{sn}}$ and $\mathrm{SK} = h(\mathrm{Id}_{\mathrm{sn}} \| K_{\mathrm{CS-SN}i})$ data to

the sensor memory before running it. In addition, it saves $\mathrm{Id}_{\mathrm{sn}}$ and $\mathrm{HSK} = \mathrm{SK} \oplus h(X_s \| \mathrm{Id}_{\mathrm{SN}})$ in its local database for eventual future utilization.

5.3. User Registration Phase. In order to create a count in the cloud server, a medical professional has to perform registration steps with the cloud server. The details of this phase are illustrated in Figure 3.

Step 1: the medical professional selects freely appropriate identity Id_i and suitable password pw_i. Afterword, he picks arbitrarily two numbers a and b. Next, he calculates $\mathrm{MID} = h(\mathrm{Id}_i \| a)$ and $\mathrm{MPW} = h(\mathrm{Id}_i | i | b)$. The two last values are transferred to the cloud server using a secured canal.

Step 2: the cloud server CS picks c randomly and computes $V = h(\mathrm{MID} \| X_s) \oplus h(\mathrm{MPW} \| c)$. Next, the CS saves MID and c in local database and transmits V to medical professional.

Step 3: the medical professional memorizes that information (V, a, b, MID) in a smart card.

5.4. Login and Authentication Phase. Once medical professional has accomplished successfully the registration process, he can connect to any sensor node. For doing that, the medical professional must accomplish the login phase by inserting his smart card. Subsequently, the authentication phase is executed. After successful login and authentication, the medical professional is allowed to interact with medical sensor nodes in real time. The steps of login and authentication phase are described below and are depicted in Figure 4.

Step_Auth1: $\mathbf{U} \longrightarrow \mathbf{CS} : \{\mathbf{V}_1, \mathbf{MI\ D}, \mathbf{A}, \mathbf{Id}_{\mathrm{SN}}, \mathbf{T}_1\}$

Firstly, medical professional user types his/her Id_i and pw_i, and the smart card verifies user's identity by checking $\mathrm{MID} \stackrel{?}{=} h(\mathrm{Id}_i \| a)$. If it is not OK, the process stops. Otherwise, the smart card picks randomly an integer A. Then, it computes $x = V \oplus h(h(\mathrm{Id}_i | \mathrm{pw}_i | b) \| c)$ and $V_1 = h(x \| A)$. Subsequently, it sends to the cloud server this message $\{V_1, \mathrm{MID}, A, \mathrm{Id}_{\mathrm{SN}}, T_1\}$.

Step_Auth2: $\mathbf{CS} \longrightarrow \mathbf{SN} : \{\mathbf{V}_2, \mathbf{B}, \mathbf{MI\ D}, \mathbf{T}_2\}$

After receiving user's communication, the cloud server checks the timestamp $T_2 - T_1 \leq T$. Then, it computes $w_1 = h(\mathrm{MID} \| X_s)$ and verifies if $V_1 \stackrel{?}{=} h(w_1 \| A)$. In the case it is validate, the cloud server generates randomly an integer B. Hence, it calculates $w_2 = \mathrm{HSK} \oplus h(\mathrm{Id}_{\mathrm{sn}} \| X_s)$, $\mathrm{HID} = h(\mathrm{MID} \| \mathrm{Id}_{SN})$, and $V_2 = h(\mathrm{HID} \| w_2 \| T_2 \| B)$ Finally, the cloud server forwards this message to the sensor node $\{\mathbf{V}_2, \mathbf{B}, \mathbf{MI\ D}, \mathbf{T}_2\}$.

Step_Auth3: $\mathbf{SN} \longrightarrow \mathbf{CS}$: $\{\mathbf{V}_3, \mathbf{C}, \mathbf{HI\ D}, \mathbf{Id}_{\mathbf{SN}}, \mathbf{T}_3\}$

Once the sensor node SN obtains the message sent by CS. First of all, it checks the authenticity of timestamp $T_3 - T_2 \leq \Delta T$. Next, it calculates the value of $\mathrm{HID}' = h(\mathrm{MID} \| \mathrm{Id}_{SN})$. Then, it checks whether $V_2 \stackrel{?}{=} h(\mathrm{HID}\prime \| \mathrm{SK} \| \mathrm{T}_2 \| \mathrm{B})$ is valid or not. If it is OK, the sensor node SN chooses random integer C and computes $V_3 = h(|\mathrm{MID}||\mathrm{Id}_{\mathrm{SN}}||\mathrm{SK}|\mathrm{T}_3 \| C)$ which is sent back to the cloud server with other values $\{\mathbf{V}_3, \mathbf{C}, \mathbf{HI\ D}, \mathbf{Id}_{\mathbf{SN}}, \mathbf{T}_3\}$.

Step_Auth4: $\mathbf{CS} \longrightarrow \mathbf{U}$: $\{\mathbf{V}_4, \mathbf{D}, \mathbf{Id}_{\mathbf{SN}}, \mathbf{T}_4\}$

When sensor's response is reaching to the cloud server, this last one verifies the validity of timestamp $T_4 - T_3 \leq \Delta T$. Subsequently, it checks if $V_3 \stackrel{?}{=} h(\mathrm{MID} \| w_2 \| T_3 \| C)$ is correct or not. In the case that it is reasonable, and the cloud server picks randomly an integer D. At that moment, it computes $V_4 = h(|w_1||\mathrm{MID}||\mathrm{Id}_{\mathrm{SN}}||T_4 \| D)$ and the session key $S_{\mathrm{Key}} = h(w_1 \| \mathrm{MID} \| \mathrm{Id}_{\mathrm{SN}})$. After that, the cloud server sends back the response to the medical professional user $\{\mathbf{V}_4, \mathbf{D}, \mathbf{Id}_{\mathbf{SN}}, \mathbf{T}_4\}\}$.

Step_Auth5:

After reception of cloud server reply, the medical professional user checks the correctness of the timestamp $T_5 - T_4 \leq \Delta T$. Hence, it authenticates the cloud server's message by checking if $V_4 \stackrel{?}{=} h(|x|||\mathrm{MID}|| \mathrm{Id}_{\mathrm{SN}} \| T_4 \| D)$ is true or false. In the case that it is OK, the medical professional user generates the session key $S_{\mathrm{Key}} = h(x||\mathrm{MID}||\mathrm{Id}_{\mathrm{SN}})$.

5.5. Password Changing Phase. Naturally, our proposed authentication protocol gives to the medical professional user the possibility to alter his/her password spontaneously. This operation can be completed in a public channel. The steps of this phase are illustrated in Figure 5. Besides, they are detailed in the following.

Step_Chang1: $\mathbf{U} \longrightarrow \mathbf{CS} : \{\mathbf{M}_u\}$

In this step, medical professional user U types in it login and password (Id_i and pw_i) . Afterwards, he/she verifies $\mathrm{MID} \stackrel{?}{=} h(\mathrm{Id}_i \| a)$. If it is all right, the user selects freely his/her new password pw_i^* and chooses two arbitrary numbers $a^*$and b^*. Next, he/she computes $\mathrm{MID}^* = h(\mathrm{Id}_i \| a^*)$ and $\mathrm{MPW}^* = h(\mathrm{Id}_i || \mathrm{pw}_i^* | b^*|)$. Finally, he/she encrypts the message $M_u = E_{SK}(\mathrm{MPW}||\mathrm{MPW}^*|| \mathrm{MID}||\mathrm{MID}^*||V)$ which will be sent to the cloud server.

Step_Chang2: $\longrightarrow \mathbf{U} : \{\mathbf{M_s}\}$

After receiving user's request the server decrypts the received message $M_u' = D_{SK}(\mathrm{MPW}||\mathrm{MPW}^*||\mathrm{MID} ||\mathrm{MID}^*||V)$. Then, it checks whether $V \stackrel{?}{=} h(\mathrm{MID} \| X_s) \oplus h(\mathrm{MPW} \| c)$ is correct or not. If it is OK, the server selects randomly c^*. Then, it replaces MID and c by MID^* and c^*, respectively. Next, it computes$V^* = h(\mathrm{MID}^* \| X_s) \oplus h(\mathrm{MPW}^* \| c^*)$ and $M_s = E_{\mathrm{SK}}(V^*)$. Finally, the server sends back to the user the message M_s.

Step_Chang3

Once the server response was received by the user, this last one decrypts the message $M_s' = D_{\mathrm{SK}}(V^*)$ and replaces V, a, b, and MID by V^*, a^*, b^* and MID^* respectively.

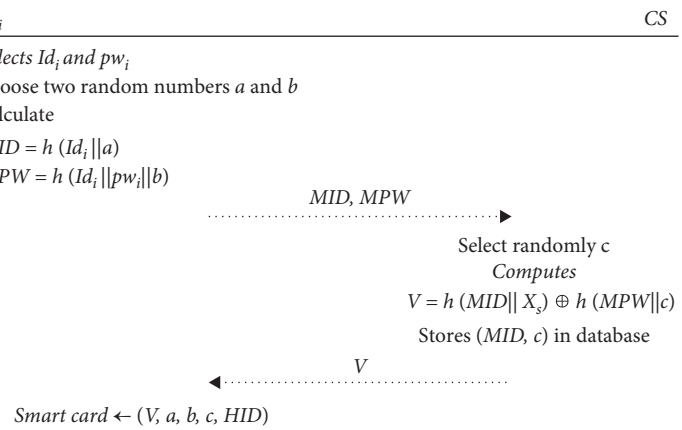


Figure 3: Registration phase.

6. Informal and Formal Analyses

In this section, we will analyze the security of our proposed protocol against numerous security attacks. As well as, we present its security properties, including mutual authentication, data integrity, user anonymity, session key exchanging, and forward secrecy (Table 2). The security of our proposed scheme is proved by both informal and formal analysis.

6.1. Informal Analyses

6.1.1. Session Key Exchanging. In our proposed protocol, the session key is generated by the user and cloud server as $S_{\text{Key}} = h(x||\text{MID}||\text{Id}_{\text{SN}})$, where $x = V \oplus h(h(|\text{Id}_i||\text{pw}_i||b)\|c) = h(\text{MID}\|X_s)$. Thanks to the reason that X_s and MID are secret, and the session key cannot be known at the end of login and authentication phase except for the medical professional user and the cloud server. As a result, we can say that the proposed scheme guarantees session key secrecy.

6.1.2. Mutual Authentication. In a public unsecured channel, each entity has authenticated each other before authentic communication takes place. So, owing to the advantages of mutual authentication in a network environment, our planned protocol reassures mutual authentication. Hence, the cloud server authenticates both the medical professional user and the sensor node. To verify users authenticity in Step 2, the server checks the correctness of receivedV_1. For checking the sensor identity, in Step 4, the cloud server verifies the exactness of $V_3 \stackrel{?}{=} h(\text{MID}\|w_2\|T_3\|C)$. Additionally, the medical professional user is able to authenticate the cloud server identity in Step 5, through examination of the legitimacy of $V_4 \stackrel{?}{=} h(|x||\text{MID}||\text{Id}_{\text{SN}}\|T_4\|D)$. Hereafter, in Step 3, the sensor node also authenticates cloud servers' message by checking the accuracy of $V_2 \stackrel{?}{=} h(\text{HID}'\|\text{SK}\|T_2\|B)$.

6.1.3. Data Integrity. It is very essential, while forwarding information between the various IoT terminals, to ensure that the data are correct and belong to the authenticated sender. Besides, it is very indispensable to ensure that the data have not been falsified by the authorities during the transfer by invoking fraudulent acts or forgery attacks. In the proposed protocol, if we suppose that an attacker captures V_1, V_2, V_3, or V_4. Then, he tries to alter their values. However, the receivers will detect this modification using the timestamp. In addition, the modification timestamps will be identified since the timestamps are embedded into V_1, V_2, V_3, and V_4.

6.1.4. DoS Attack. Our proposed authentication protocol can resist against DoS attack. Accordingly, the user is able to know if her/his message has passed the authentication phase or not, especially, after getting server's response which can be validated or rejected message. With goals to verify the freshness of received message, our protocol uses timestamps. Furthermore, arbitrary numbers are produced in every stage and in every session. Besides, since the duplicated messages are unacceptable, the attacker is not able to perform the DoS attack. Thus, our suggested method can resist against DoS attack.

6.1.5. No Verification Table. According to the proposed protocol, the confidential information of each user, including the password, is not stored by the cloud server or the sensors. In case an attacker succeeds in the hacking cloud server or node, he/she would not be actually capable of obtaining the password checking data. Therefore, the hacker will not be able to retrieve any authenticating details.

6.1.6. Off-Line Password Guessing Attack. Assume that an attacker has got the smart card. Then, he/she extracts all stored parameters in it. If he/she wants to guess the password, he/she cannot, since the only value that contains password is $V = h(\text{MID}\|X_s) \oplus h(\text{MPW}\|c)$. Therefore, the attacker has to know the value of X_s and Id_i. However, Id_i is encrypted using one-way hash function, and X_s is server's private key. Consequently, we can conclude that our proposed scheme is secure against offline password guessing attack.

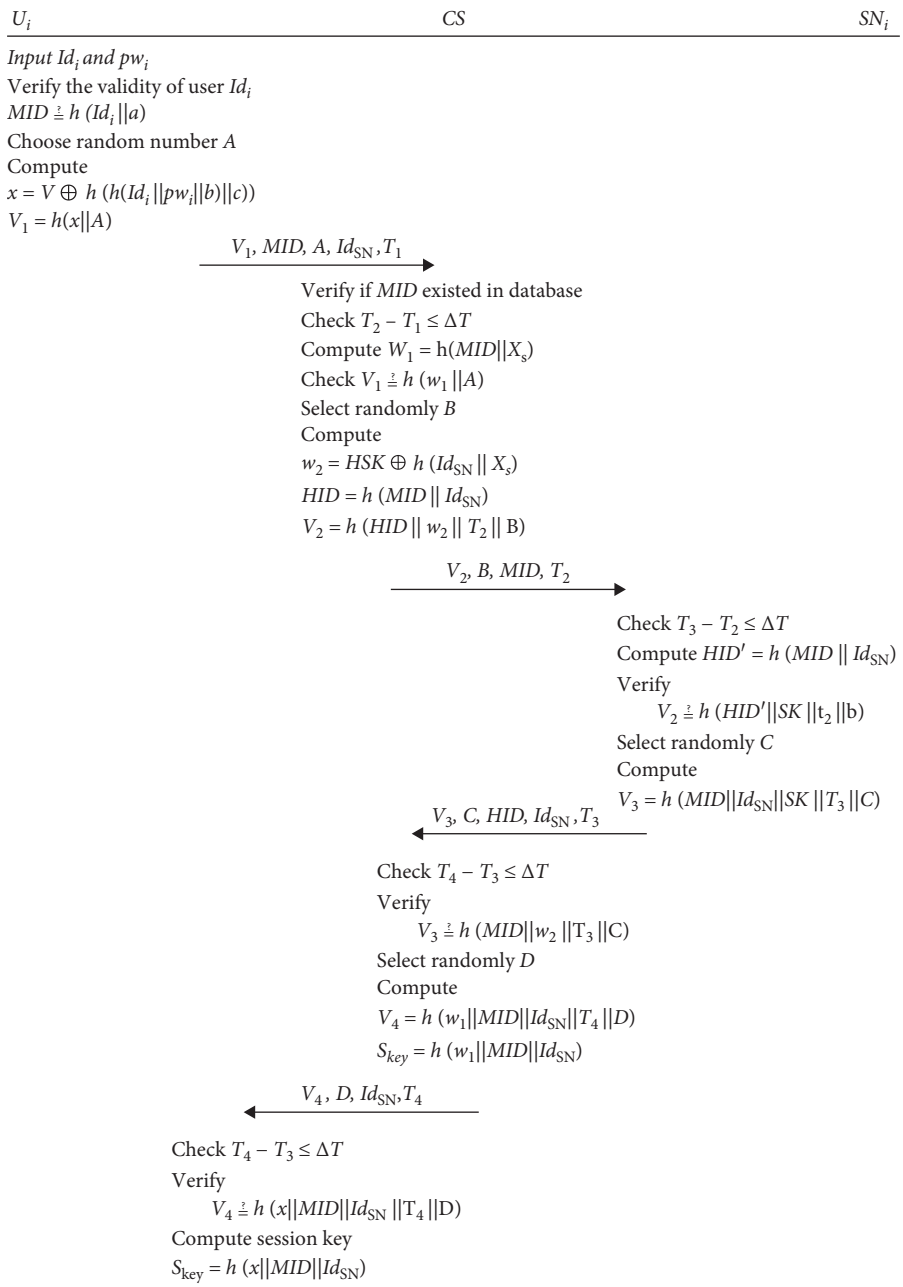


Figure 4: Login and authentication phase.

6.1.7. Replay Attack. Assume that an adversary replays the old message to the server. In our proposed protocol, the cloud server discovers that this message is not fresh. Originally, the cloud server checks the validity of timestamp $T_2 - T_1 \leq \Delta T$; in the case that it is noted valid, the session stops. The same thing happens after receiving sensor node's message $T_4 - T_3 \leq \Delta T$. The sensor node and user use $T_3 - T_2 \leq \Delta T$ and $T_5 - T_4 \leq \Delta T$, respectively, to check the newness of cloud server's message. Consequently, our proposed protocol can withstand against replay attack.

6.1.8. Insider Attack. Our proposed scheme can resist against privileged insider attack. Assume that a malicious or pirate has an access the registration data {MID,c }. Even if we have those data, the attacker can neither guess the password nor initiate any kind of counterfeit attack. Furthermore, he/she must fight the secrecy of one way hash function if he/she wants to have just the user Id. On the contrary, for initiating impersonation attack, the attacker must have access to cloud server's secret key. Consequently, our proposed protocol can resist against privileged insider attack.

6.1.9. Perfect Forward Secrecy. In our proposed protocol, the session key S_{Key} is computed as $S_{Key} = h(w_1||MID||Id_{SN})$, where $w_1 = h(MID||X_s)$. The session key contains server's secret key X_s and users MID that depend on user's encrypted

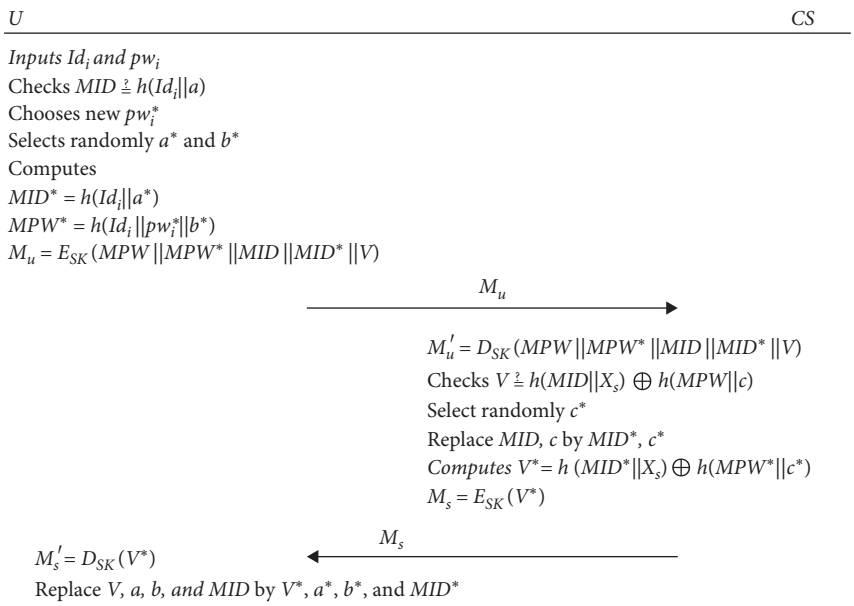


Figure 5: Password changing phase.

Table 2: Security feature's comparison.

	Kumar et al. [33]	He et al [34]	Amin et al [35]	Sharma and Kalra [14]	Ours
Mutual authentication	✓	✓	✓	✓	✓
Data integrity	✓	✓	✓	✓	✓
No verification table	✗	✗	✓	✓	✓
Session key exchanging	✓	✓	✓	✓	✓
DoS attack	✓	✓	✓	✓	✓
Perfect forward secrecy	✗	✗	✗	✓	✓
Off-line password guessing attack	✗	✗	✗	✗	✓
Replay attack	✗	✗	✗	✓	✓
Insider attack	✗	✗	✗	✓	✓

✓: secured against attack. ✗: not secured against attack.

Id_i. In other words, the session key S_{Key} depends on secure parameters which are not accessible to the attacker. Consequently, our proposal assures perfect forward secrecy.

7. Formal Analyses

7.1. Security Examination Using Scyther. In this section, we initially clarify the usefulness of Scyther tool [36], which was useful for formal security examination of our proposed scheme. Formerly, we showed gained outcomes by using this tool. Absolutely, it is software for automatically checking of security protocols. It is based on a reverse analysis method. The requests correspond to the knowledge of the participants (source and destination) and also to the wishes of a possible hacker. Symmetric and asymmetric encryption, hash functions, and encryption keys are also embedded in Scyther.

Our planned protocol is then written in the security protocol description language (SPDL). This specification allows us to specify different roles of the medical professional user, the cloud server, and the sensor node. In each role, event sequences are embedded including sending, receiving, declarations, and complaints. According to Figure 6, which details the obtained results, we can notice that our protocol is secure against many attacks. Besides, it meets the necessary security-related fundamentals.

7.2. Security Verification Using Random Oracle Model. Actually, the random oracle model is used for formal security analysis. In this analysis, we verify that a given attacker is not in measure to recover important secret values including Id, Pw, V_1Idi, and session key S_{Key}. In our study, the similar procedure presented in [37–39] is adopted. The random oracle is detailed in the following.

Reveal: it produces the input of a hash function; let λ be absolutely from a specified hash output ψ, where $\psi = f(\lambda)$.

Formula 1. Let us suppose that an attacker has stool user's smart device; in addition, he has the knowledge about the transactions $\{V_1 - V_4, T_1 - T_4, Id_{SN}, MID\}$ that has been transmitted over an untrusted network. While, h is the hash function that is understood as a random oracle, and the introduced protocol is secured against the attacker to retrieve the Id, Pw, V_1, and the session key S_{Key} of legitimate user U.

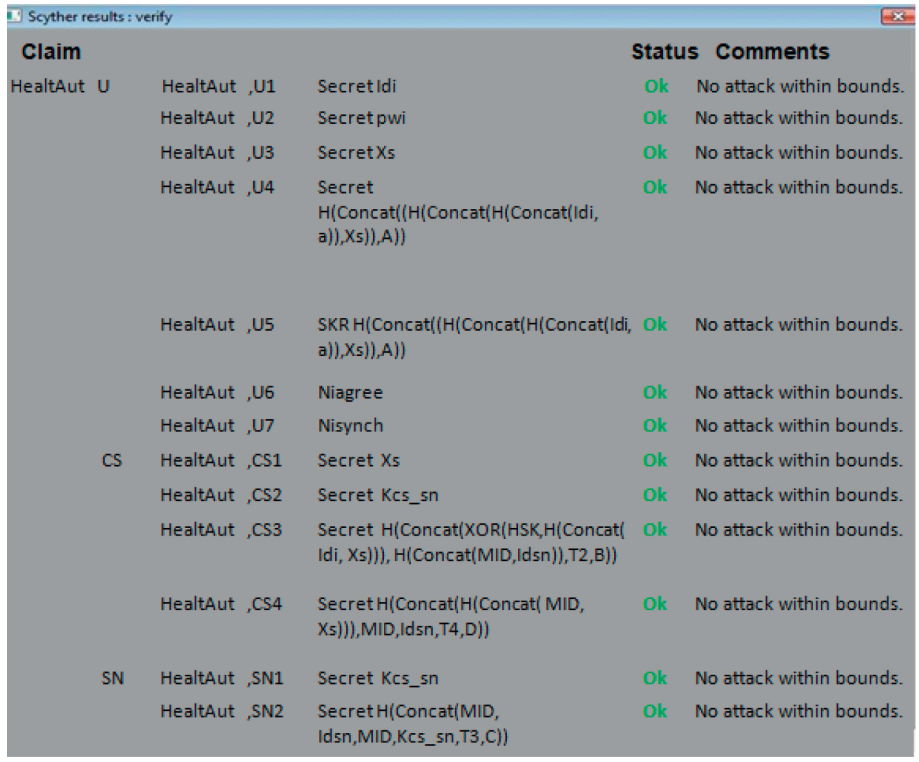


Figure 6: Experimental test results.

Proof. Suppose that the attacker $\mathscr{A}$ has obtained user's parameters Id and Pw by using smart card data $\{V, a, b, \text{MID}\}$ and the intercepted messages $\{V_1 - V_4, T_1 - T_4, \text{Id}_{\text{SN}}, \text{MID}\}$, and the tentative algorithm $\text{Tent1}^{\text{hash}}_{\mathscr{A},\text{RPMAP}}$ defines the possibility of achievement Ach_1 for $\text{Exp1}^{\text{hash}}_{\mathscr{A},\text{RPMAP}}$ by means of the following specified function:

$$\text{Ach}_1 = \left|\text{Pb} \cdot \left[\text{Tent}\,1^{\text{hash}}_{\mathscr{A},\text{RPMAP}} = 1 - 1\right]\right|. \quad (1)$$

$\text{Pb}[E]$ denotes the probability for a given event E. In this experiment, we define the advantageous condition as $F_{Av1}(t_1, rq_1) = \text{Max}_{\mathscr{A}}\{\text{Ach1}\}$, where Max is determined based on totally adversaries taking the execution time t_1 and rq_1 is the total maximum requests transmitted to the reveal oracle. Our presented protocol is secure against the adversary to discover Id, Pw, and V_1 if $F_{Av1}(t_1, \text{rq}_1) \le \varepsilon$ for a small value of $\varepsilon > 0$. The trial model $\text{Tent}\,1^{\text{hash}}_{\mathscr{A},\text{RPMAP}}$ indicates that if the hacker is able to compute the inverse of the hash function, it can recover user's parameters Id, Pw, and V_1. Nevertheless, it is impossible to determine the reverse of this one-way hash function in polynomial period, such that $F_{Av1}(t_1, rq_1) \le \varepsilon$ for a small value of $\varepsilon > 0$. Accordingly, we can say that our proposed scheme is safe from the attacker for obtaining Id, Pw, V_1, and S_{Key}. □

Formula 2. If we suppose that the hash function behaves as random oracle and the attacker have intercepted the message forwarded on unsecured channel $\{V_1 - V_4, T_1 - T_4, \text{Id}_{\text{SN}}, \text{MID}\}$, the attacker may not be able to calculate user's session key S_{Key}.

Proof. If an attacker that intercepts the transferred message $\{V_1 - V_4, T_1 - T_4, \text{Id}_{\text{SN}}, \text{MID}\}$ tries to generate user's session key S_{Key}, the probability of achievement Ach_1 in this calculation is defined by the tentative algorithm $\text{Tent2}^{\text{hash}}_{\mathscr{A},\text{RPMAP}}$ as

$$\text{Ach}_2 = \left|\text{Pb} \cdot \left[\text{Tent}\,2^{\text{hash}}_{\mathscr{A},\text{RPMAP}} = 1 - 1\right]\right|. \quad (2)$$

$\text{Pb}[E]$denotes the probability for a given event E. In this experiment, we define the advantageous function as $F_{Av2}(t_2, rq_2) = \text{Max}_{\mathscr{A}}\{\text{Ach2}\}$, where Max is determined based on totally adversaries taking the execution time t_1 and rq_1 is the total maximum requests transmitted to the reveal oracle. Our presented protocol is secure against the adversary to discover S_{Key} if $F_{Av2}(t_2, rq_2) \le \varepsilon$ for a small value of $\varepsilon > 0$. The trial model $\text{Tent2}^{\text{hash}}_{\mathscr{A},\text{RPMAP}}$ indicates that if the hacker is able to compute the inverse of the hash function, it can recover user's parameters Id, Pw, and V_1. Nevertheless, it is impossible to determine the reverse of this one-way hash function in legal period, such that $F_{Av2}(t_2, rq_2) \le \varepsilon$ for a small value of $\varepsilon > 0$. Accordingly, we can close that our proposed scheme is safe from the attacker for computing S_{Key} (Algorithm 1 and 2). □

8. Performance and Comparative Analysis

This section details the results of the performance analysis of our protocol. In the first place, we display the performance of our proposed protocol in point of view of the ability to resist against security attacks. Secondly, our protocol is compared

Begin
(1) Intercept the transmitted values $\{\mathbf{V}_1, \mathbf{V}_4, \mathbf{MI\ D}, \mathbf{A}, \mathbf{Id_{SN}}, \mathbf{T}_1, \mathbf{T}_4\}$
(2) Call reveal oracle on $\mathbf{V}_4$ for getting value of $\{(x\|MI\ D\|\mathrm{Id}_{\mathrm{SN}}\|T_4\|D)\}$ *as* $(x\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}}\|T_4\|D)) \leftarrow reveal1\ (\mathbf{V}_4)$
(3) Call reveal oracle on $\mathbf{V}_1$ for getting value of $\{(x\|A)\}$ *as* $(x\|A) \leftarrow reveal1\ (\mathbf{V}_1)$
(4) Calculate $x' = V \oplus h((h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c))$
(5) If $(x' \stackrel{?}{=} x)$, then
(6) Extract the parameters $\{\mathbf{V}, \mathbf{a}, \mathbf{b}, \mathbf{MI\ D}\}$ from the mobile device.
(7) Calculate $h(h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c) = V \oplus x$
(8) Call reveal oracle on input $h(h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c$ to discover $\{h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c\}$ as $(h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c \leftarrow reveal1(h(h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c)$.
(9) Call reveal oracle on input $h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)$ to discover $\{(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\}$ as $(\mathrm{Id}_i\|\mathrm{pw}_i\|b) \leftarrow reveal1\ (h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)$.
(10) Ten, compute MID' $= h(\mathrm{Id}_i\|a)$
(11) If (MID′$\stackrel{?}{=}$ MID), then
Accept Id_i', pw_i', and a' as valid identity, password, and random number.
Return (true)
(12) Else
Return (false)
(13) End if
End.

Algorithm 1: Tent $1^{\mathrm{hash}}_{A,\mathrm{RPMAP}}$.

Begin
(14) Intercept the transmitted values $\{\mathbf{V}_1, \mathbf{V}_4, \mathbf{MI\ D}, \mathbf{A}, \mathbf{Id_{SN}}, \mathbf{T}_1\}$
(15) Call reveal oracle on $\mathbf{V}_4$ for getting value of $\{(x\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}}\|T_4\|D)\}$ *as* $(x\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}}\|T_4\|D) \leftarrow reveal1\ (\mathbf{V}_4)$
(16) Call reveal oracle on $\mathbf{V}_1$ for getting value of $\{(x\|A)\}$ *as* $(x\|A) \leftarrow reveal1\ (\mathbf{V}_1)$
(17) Calculate $w_1 = h(\mathrm{MID}\|X_s)$
(18) Generate the session key $S_{\mathrm{key}}' = h(w_1\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}})$
(19) Compute $V_4' = (x\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}}\|T_4\|D)$
(20) If $(V_4' \stackrel{?}{=} \mathbf{V}_4)$, then
Accept $S_{\mathrm{key}}' = h(w_1\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}})$ as effective user's session key.
Return (true)
(21) Else
Return (false)
(22) End if
End.

Algorithm 2: Tent $2^{\mathrm{hash}}_{A,\mathrm{RPMAP}}$.

to other related ones according to the computational complexity. Hence, the results of the first comparison are illustrated in Table 2. As it is very clear, we can realize that our protocol can resist against various attacks, and it is able to guarantee several security requirements including perfect forward secrecy, session key exchange, and mutual authentication.

As we have mentioned above, the calculation charges of our proposed scheme is compared to other correlated protocols specifically Alzahrani et al. [40], Li et al. [41], Azrour et al. [32], and Sharma and Kalra [14]. In this calculation, very weedy procedures such as string concatenation operation and XoR procedure are ignored. The sign T_h personifies the time charge of one way hash operation, whereas T_{ED} characterizes the time complexity of symmetric key operations and T_{pm} denotes the computational charge of elliptic curve point multiplication.

In our protocol, the medical professional user calculates $6T_h$ and the cloud server computes $8T_h$, while the sensors node calculates $3T_h$. Consequently, the whole calculation complexity of our scheme is only $17T_h$. As displayed in Table 3, one can remark that our scheme is based only on one-way hash functions which do not consume much time if it is compared to the symmetric key operations. In case we compare the number of time that T_h uses, we will find that our protocol uses only 17. Hence, we confirm that our proposed protocol is appropriate for remote healthcare applications based one cloud-IoT.

Table 3: Comparative analysis.

	Alzahrani et al [40]	Li et al. [41]	Sharma and Kalra [14]	Azrour et al. [32]	Ours
User	$1T_{ED} + 11T_h$	$2T_{ED} + 6T_h$	$11T_h$	$5T_h$	$6T_h$
Server/gateway	$7T_h$	$6T_{ED} + 7T_h$	$7T_h$	$6T_h + 4T_{pm}$	$8T_h$
Sensor node	$1T_{ED} + 5T_h$	$2T_{ED} + 5T_h$	$5T_h$	$2T_h + 2T_{pm}$	$3T_h$
Total	$2T_{ED} + 23T_h$	$10T_{ED} + 18T_h$	$23T_h$	$13T_h + 6T_{pm}$	$17T_h$

9. Conclusions

Modern technologies are currently having a great impact on the healthcare world. Thus, healthcare professionals can have access to patient confidential data online, which mandates strong authentication protocols for home patient monitoring. So, it is necessary to consider a lightweight authentication scheme to guarantee secure communication between the healthcare system-based cloud-IoT. In the present study, we firstly demonstrated that the protocol proposed by Sharma and Kalra is vulnerable and exposes some security issues. Then, we have proposed our authentication protocol to mitigate the prior work vulnerable issues. Afterwards, we have demonstrated informally that our protocol can resist against various attacks and can provide security requirement. In addition, the simulation done under Scyther tools confirms that our protocol is formally secured and meets security fundamentals.

Data Availability

Experimental results, obtained using Scyther tool, are available and will be shared with authors at https://sites.Google.com/umi.ac.ma/azrour.

Conflicts of Interest

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

References

[1] D. Pivoto, P. D. Waquil, E. Talamini, C. P. S. Finocchio, V. F. Dalla Corte, and G. de Vargas Mores, "Scientific development of smart farming technologies and their application in Brazil," *Information Processing in Agriculture*, vol. 5, no. 1, pp. 21–32, 2018.

[2] P. Visconti, N. I. Giannoccaro, R. d. Fazio, S. Strazzella, and D. Cafagna, "IoT-oriented software platform applied to sensors-based farming facility with smartphone farmer app," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1095–1105, 2020.

[3] G. I. Hapsari, G. Andriana Mutiara, L. Rohendi, and A. Mulia, "Wireless sensor network for monitoring irrigation using XBee Pro S2C," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1345–1356, 2020.

[4] M. S. U. Chowdury, T. B. Emran, S. Ghosh et al., "IoT based real-time river water quality monitoring system," *Procedia Computer Science*, vol. 155, pp. 161–168, 2019.

[5] Z. Mohd Yusoff, Z. Muhammad, M. S. I. Mohd Razi, N. F. Razali, and M. H. C. Hashim, "IOT-Based smart street lighting enhances energy conservation," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, p. 528, 2020.

[6] J. Poushter, "Smartphone ownership and internet usage continues to climb in emerging economies," *Pew Research Center*, vol. 9, no. 4, pp. 1–44, 2016.

[7] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.

[8] M. S. Hossain, G. Muhammad, and A. Alamri, "Smart healthcare monitoring: a voice pathology detection paradigm for smart cities," *Multimedia Systems*, vol. 25, no. 5, pp. 565–575, 2019.

[9] Y.-T. Park, "Emerging new era of mobile health technologies," *Healthcare Informatics Research*, vol. 22, no. 4, pp. 253-254, 2016.

[10] J. Mabrouki, M. Azrour, G. Fattah, D. Dhiba, and S. E. Hajjaji, "Intelligent monitoring system for biogas detection based on the Internet of Things: mohammedia, Morocco city landfill case," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 10–17, 2021.

[11] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, "IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.

[12] J. Mabrouki, M. Azrour, Y. Farhaoui, and S. El Hajjaji, "Intelligent system for monitoring and detecting water quality," in *Big Data and Networks Technologies*, Y. Farhaoui, Ed., vol. vol. 81, pp. 172–182, Springer International Publishing, Cham, Switzerland, 2020.

[13] J. Mabrouki, M. Azrour, and S. El Hajjaji, "Use of internet of things for monitoring and evaluation water's quality: comparative study," *International Journal of Cloud Computing*, 2021, In press.

[14] G. Sharma and S. Kalra, "A lightweight user authentication scheme for cloud-IoT based healthcare services," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 43, no. 1, pp. 619–636, 2019.

[15] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proceedings Of the 2nd ACM Workshop on Security Of Ad Hoc and Sensor Networks*, pp. 59–64, San Diego, CA, USA, 2004.

[16] K. H. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," *Sensor Networks, Ubiquitous, and Trustworthy Computing*, vol. 1, p. 8, 2006.

[17] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[18] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.

[19] R. Song, "Advanced smart card based password authentication protocol," *Computer Standards & Interfaces*, vol. 32, no. 5-6, pp. 321–325, 2010.

[20] X. Xu, Z. P. Jin, H. Zhang, and P. Zhu, "A dynamic ID-based authentication scheme based on ECC for telecare medicine information systems," *In Applied Mechanics And Materials*, vol. 457, pp. 861–866, 2014.

[21] X. Yan, W. Li, P. Li, J. Wang, X. Hao, and P. Gong, "A secure biometrics-based authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 37, no. 5, pp. 1–6, 2013.

[22] D. Mishra, S. Mukhopadhyay, S. Kumari, M. K. Khan, and A. Chaturvedi, "Security enhancement of a biometric based authentication scheme for telecare medicine information systems with nonce," *Journal of Medical Systems*, vol. 38, no. 5, pp. 1–11, 2014.

[23] Z. Tan, "A user anonymity preserving three-factor authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, no. 3, pp. 1–9, 2014.

[24] E.-J. Yoon and C. Kim, "Advanced biometric-based user authentication scheme for wireless sensor networks," *Sensor Letters*, vol. 11, no. 9, pp. 1836–1843, 2013.

[25] D. He, C. Chen, S. Chan, J. Bu, and A. V. Vasilakos, "ReTrust: attack-resistant and lightweight trust management for medical sensor networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 4, pp. 623–632, 2012.

[26] D. Mishra, J. Srinivas, and S. Mukhopadhyay, "A secure and efficient chaotic map-based authenticated key agreement scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, no. 10, p. 120, 2014.

[27] Q. Jiang, J. Ma, G. Li, and X. Li, "Improvement of robust smart-card-based password authentication scheme," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 383–393, 2015.

[28] Z.-Y. Cheng, Y. Liu, C.-C. Chang, and S.-C. Chang, "An improved protocol for password authentication using smart cards," vol. 22, no. 4, 2012.

[29] M. Azrour, M. Ouanan, Y. Farhaoui, and A. Guezzaz, "Authentication Protocol for Internet of Things," *Studies in Big Data*, vol. 53, pp. 67–74, 2019.

[30] N. Ye, Y. Zhu, R.-c. Wang, R. Malekian, and L. Qiao-min, "An efficient authentication and access control scheme for perception layer of internet of things," *Applied Mathematics & Information Sciences*, vol. 8, no. 4, pp. 1617–1624, 2014.

[31] X. Cheng, Z. Zhang, F. Chen et al., "Secure identity authentication of community medical internet of things," *IEEE Access*, vol. 7, pp. 115966–115977, 2019.

[32] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for Internet of Things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.

[33] P. Kumar, S.-G. Lee, and H.-J. Lee, "E-SAP: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," *Sensors*, vol. 12, no. 2, pp. 1625–1647, 2012.

[34] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, and S.-S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Systems*, vol. 21, no. 1, pp. 49–60, 2015.

[35] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud Computing environment," *Future Generation Computer Systems*, vol. 78, pp. 1005–1019, 2018.

[36] C. J. F. Cremers, "The scyther tool: verification, falsification, and analysis of security protocols," *Computer Aided Verification*, pp. 414–418, 2008, In press.

[37] N. Koblitz and A. J. Menezes, "The random oracle model: a twenty-year retrospective," *Designs, Codes and Cryptography*, vol. 77, no. 2-3, pp. 587–610, 2015.

[38] J.-S. Coron, J. Patarin, and Y. Seurin, "The random oracle model and the ideal cipher model are equivalent," *In Advances in Cryptology*, Springer, Berlin, Germany, pp. 1–20, 2008.

[39] M. Bellare and P. Rogaway, "Random oracles are practical," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 62–73, New York, NY, USA, 1993.

[40] B. A. Alzahrani, A. Irshad, K. Alsubhi, and A. Albeshri, "A secure and efficient remote patient-monitoring authentication protocol for cloud-IoT," *International Journal of Communication Systems*, vol. 33, no. 11, p. e4423, 2020.

[41] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, and M. K. Khan, "A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity," *Security and Communication Networks*, vol. 9, no. 15, pp. 2643–2655, 2016.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9918697, 11 pages
https://doi.org/10.1155/2021/9918697

Research Article

Implementation of Blockchain Consensus Algorithm on Embedded Architecture

Tarek Frikha,[1] **Faten Chaabane**,[2] **Nadhir Aouinti**,[1] **Omar Cheikhrouhou**,[3] **Nader Ben Amor**,[1] **and Abdelfateh Kerrouche**[4]

[1]*Université de Sfax, CES Lab, 3038, Sfax, Tunisia*
[2]*Université de Sfax, Regim Lab, Sfax, 3038, Tunisia*
[3]*College of CIT, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*
[4]*C82c School of Engineering and the Built Environment Edinburgh Napier University, 10 Colinton Road, Edinburgh, EH10 5DT, UK*

Correspondence should be addressed to Tarek Frikha; tarek.frikha@enis.tn

Received 6 March 2021; Revised 29 March 2021; Accepted 8 April 2021; Published 23 April 2021

Academic Editor: Azeem Irshad

The adoption of Internet of Things (IoT) technology across many applications, such as autonomous systems, communication, and healthcare, is driving the market's growth at a positive rate. The emergence of advanced data analytics techniques such as blockchain for connected IoT devices has the potential to reduce the cost and increase in cloud platform adoption. Blockchain is a key technology for real-time IoT applications providing trust in distributed robotic systems running on embedded hardware without the need for certification authorities. There are many challenges in blockchain IoT applications such as the power consumption and the execution time. These specific constraints have to be carefully considered besides other constraints such as number of nodes and data security. In this paper, a novel approach is discussed based on hybrid HW/SW architecture and designed for Proof of Work (PoW) consensus which is the most used consensus mechanism in blockchain. The proposed architecture is validated using the Ethereum blockchain with the Keccak 256 and the field-programmable gate array (FPGA) ZedBoard development kit. This implementation shows improvement in execution time of 338% and minimizing power consumption of 255% compared to the use of Nvidia Maxwell GPUs.

1. Introduction

The global IoT market is expected to reach a value of USD 1,386.06 billion by 2026 from USD 761.4 billion in 2020 at a CAGR of 10.53%, during the period 2021–2026 [1].

The IoT technology is connecting various devices such as mobile phones, sensors, and household appliances together for collecting and sharing data for the next industrial revolution of intelligent connectivity. The fourth industrial revolution (Industry 4.0) interconnects smart digital technology with real worlds to create smart manufacturing and supply chain management [1]. In the current context, the emergence of Industry 4.0 and the adoption of IoT devices require manufacturers to implement innovative ways to advance production with intelligent connectivity that uses more robotics and avoids industrial accidents and machines' downtime failure. Therefore, industries, hospitals, supply chains, governments, banks, and logistics need to be connected using Distributed Ledger Technologies (DLT) such as blockchain technology to react quickly for a more connected world. This will enable more secured process dealing with big data analysis generated by IoT devices.

Blockchain is mainly dealing with data storage and management and a distribution technology that is transparent and secure and operates regardless of a central control body [2].

Unlike traditional methods, blockchain allows peer-to-peer transfer of digital assets without the need for an intermediary. This technology was inspired by Bitcoin [3] cryptography and then has emerged, evolved, and spread in

several applications including finance [4], health [5], administration [6], industry [7], agriculture [8], and smart cities [9]. It affects also other sectors such as the transfer of goods (supply chain), digital media transfer (sale of works of art), remote service delivery (travel and tourism), distributed intelligence (graduation), electricity generation and distribution, startup fundraising, electronic voting, identity management, crowdfunding (increasing startup funds), and crowd-operation (remote voting).

The first blockchain success notified with Bitcoin, was followed by other blockchains such as Ethereum [10], Hyperledger Fabric [11], Azur, Grid+, IOTA [12], and Tezos [13, 14].

Representing the new generation of blockchain, Ethereum can play a major role of a public blockchain like Bitcoin, or a private blockchain such as Hyperledger Fabric. It is also the basis of other blockchains which are specific frameworks for applications, such as the Azur. For example, the blockchain proposed by Microsoft, which was optimized to take advantage of the characteristics of the cloud. Another example is the Grid + blockchain which is used in energy management applications.

To preserve the security of the blockchain, a specific algorithm, known as consensus, is used. It allows a new block to be added to the blockchain without compromising the integrity of data stored in the distributed ledger.

Moreover, some blockchains are defined with intelligent contracts and software platforms to play the role of links in the blockchain. However, all these blockchains are using consensus to preserve their security. In this context, several types of consensus are proposed in the literature such as the Proof of Work (PoW), the Proof of Stake (PoS), the Proof of Authority (PoA), the PBFT, and the Ripple and the Raft [14]. These consensus algorithms have different complexity levels. One of the most complex and energy-intensive consensuses is PoW which was used in several blockchains such as Bitcoin, Ethereum, and IoTA [15]. As an example, the mining process time is approximately 10 minutes for Bitcoin [16] and 15 seconds for Ethereum using Nvidia RTX 3080 GPU [17]. Regardless of the number of miners, it still takes about 10 minutes to mine one Bitcoin. At 600 seconds (10 minutes), all else being equal it will take 72,000 GW (or 72 terawatts) of power to mine a Bitcoin using the average power usage provided by ASIC miners [16].

The use of blockchain, particularly the mining part, requires significant computing resources. In this paper, a feasibility study of implementing the blockchain on an embedded system and particularly on field-programmable gate array FPGA is presented taking into consideration all the resource requirements to validate this approach.

An embedded architecture is proposed to implement the PoW consensus, especially on FPGA-based architecture. This optimized architecture should accelerate the classical PoW process and consequently minimize the energy consumption. This proposed architecture is chosen according to a comparison between different software (SW), hardware (HW), and mixed architectures.

More precisely, the contribution of this paper is as follows.

The main contribution of this paper consists of two parts. First, an embedded architecture is proposed to implement the PoW consensus algorithm on FPGA. This part is called the off-chain system block. And, the second part is dedicated to the design of an off-chain/on-chain system. The PoW implementation and particularly the hash algorithm were off-chain (on FPGA). The node smart contract, transactions, and blocks where on-chain (they are implemented on the Raspberry Pi 3 platform).

The remainder of this paper is as follows. In Section 2, we describe the basic notions of the blockchain, particularly its different consensus followed by a study on embedded technologies and mixed HW/SW architectures [18]. In Section 3, a description of the PoW used in the blockchain Ethereum will be dissected. The profiling of this function will allow to describe the embedded architecture to be chosen. Section 4 is reserved for the choice of the architecture and the different parts of our system containing the consensus implementation. The last part will be reserved for the results obtained and the comparison between SW on GPU and HW-implemented architecture from execution time and energy consumption point of view. Finally, in Section 5, we conclude and give potential perspectives.

2. Background

2.1. Blockchain Overview. In this section, we give an overview of the blockchain technology and its different classes and main components.

2.2. Security-Based Blockchain Classification. From the security point of view, blockchain can be classified as public, consortium, and private.

2.3. Public Blockchain. The blockchain is said to be public because it is open to everyone. Thus, it is assimilated to a marketplace, where anyone can open a store to offer any products and services. In this case, there are no restrictions on the comings and goings of visitors who are free to visit the different stores to make purchases.

Consequently, a public blockchain has several characteristics, such as a decentralized network which is open to all actors without any restriction, data can be consulted by all without any restriction, and data can be consulted by all without any restriction, but it is indelible, forgery-proof and cannot be modified afterwards. In this class of blockchain, the use of the PoW consensus makes the blockchain's transactions impossible to falsify and very easy to manipulate.

There are many examples of public blockchains: Bitcoin, Ethereum, Ripple [14], Litecoin [19], and Dash.

2.4. Consortium Blockchain. It consists of a permitted blockchain which is partially decentralized and differs from public blockchains because its network is only accessible to a limited number of users.

New members must be validated by the nodes and already existing members in the consortium, and the

accessibility of the data depends on the access rights granted to each node. It can be compared to a corporate marketplace (here, the "consortium") for which only consortium members would be allowed to open a store to offer products and services. However, the consortium may grant some exemptions to open additional stores. The comings and goings in this marketplace are normally restricted by the rules defined by the consortium.

It should be noted that the vast majority of existing consortium blockchains operate under the Proof of Authority (PoA) system. As examples of public blockchains, we can cite Ripple [20], Funds DLT, etc.

2.5. Private Blockchain. In contrast to public blockchains, private blockchains (of which permitted blockchains are a special case) are like distributed databases.

Their characteristics are as follows:

(i) The network is accessible to a limited number of users. New entrants must be validated by a central decision-making body.

(ii) The accessibility of the data depends on the access rights of each node. This is defined by the central decision-making body.

(iii) On a private blockchain, the consensus is based on the trust placed in all the validator nodes.

A private blockchain can be compared to a marketplace where all members authorized to launch a store, or to sell products and services, are only members of this same structure.

As a result, the cases of use are very frequent. As for distributed databases, they are useful for sharing confidential or important data within an organization or within the different entities of a group.

There are many examples of private blockchains. We can cite Hyperledger Fabric, Grid+, Azur, Ethereum (both private and public blockchains), etc.

2.6. Consensus Algorithms. It consists of the transition from centralized systems where the administrator or the central system can validate or invalidate transactions such as the banking system and database management systems.

In this kind of systems, the administrator is the valid or invalid manager. In decentralized systems such as blockchains, the absence of an administrator requires another protocol for verification and validation. The intermediary functions are moved to the periphery participating pair in the infrastructure of the chain. Since the peers do not necessarily know each other, it is a decentralized system.

The consensus algorithm consists of firstly setting up a process to validate, verify, and confirm transactions, then recording the transactions in a large distributed directory, creating a block record (a chain of blocks), and finally implementing a consensus protocol.

Thus, validation, verification, consensus, and immutable recording lead to trust and security of the blockchain.

Several types of consensus are used in the blockchain including PoW [21], PoS [21], PoA [21], PBFT [21], Ripple [20], and DAC [21]. In this paper, we will describe only the PoW algorithm that will be implemented in HW (FPGA platform). In the next part, we will describe the state-of-the-art of embedded systems.

3. Overview of Embedded Architectures' Solutions

The evolution of electronics and microelectronics has made it possible to minimize the size of transistors to increase the number of electronic components integrated on the same chip. The main component is the microprocessor. Microprocessors consist of one or more central processing units (CPUs), as well as other modules required for their operation such as memory controllers, cache memory, and I/O controllers.

However, in some systems, the integrated circuit contains not only the microprocessor but also other components such as microcontrollers and GPUs. Such a system is called System on Chip (SoC). These SoCs are based on the minimization of space and power consumption, while preserving the necessary performance for the constraints of the appropriate applications.

For example, a typical modern SoC contains the CPU, the GPU, the communication modules (Wi-Fi, Bluetooth, etc.), a module for localization, as well as other subsystems and coprocessors providing various functions such as device security [9].

These SoCs are used in applied computer systems generally called embedded systems. Although there is no formal definition of the latter, they are generally information systems designed for well-defined tasks [22] and are integrated in other products [23].

The use of embedded systems has also touched the blockchain technology. Thus, e-health, agriculture, light and heavy industry, e-learning, and augmented reality [24] applications are often based on SoCs to set up systems that meet their different needs.

Thus, we find different architectures that are in adequacy with the different needs. We can find single processor systems whose performance is enhanced by HW accelerators (IPs) [24], or massively parallel architectures that take advantage of the large number of processors operating in perfect parallelism [25].

If the use of embedded systems has touched several domains, its use in the blockchain domain has remained rather limited, especially for FPGAs' technology. In fact, despite its various internal resources such as embedded high-speed memory, parallel computing blocks, and flexible architecture, which are suitable for computationally complex applications, it is still limited to the use of the PoW consensus.

Such idea is rarely discussed in the literature. We mention particularly in the work presented in [18], where the authors presented the possibility of implementing an embedded robotics application managed by blockchain.

In the work by Chaari [26], an embedded system based on a Raspberry Pi 3 platform was used. One of the problems encountered in this work is essentially that the Raspberry is unable to run all the PoW consensus software functions due to its limited capabilities.

In this paper, the main target is to propose an embedded architecture suitable for blockchain applications and able to support the implementation of the PoW consensus. Hence, we will show the feasibility as well as the gain realized by using such architecture adopted at Ethereum PoW on FPGAs.

3.1. Ethereum Blockchain Components. In this section, we are interested in blockchain components, especially Ethereum blockchain and its different components.

The blockchain is based on specific terminology representing important concepts. Among the frameworks of the blockchain, there are the following.

3.1.1. Transactions. These are the exchanges of data between different users. Each transaction is signed by the sender's private key. Thanks to this signature, the security of the transactions is guaranteed. Therefore, any modification of these transactions during transmission can be avoided.

3.1.2. Blocks. A block is a record in the blockchain which contains the confirmed transactions. Thus, each open transaction will be added to a block. After a period, for a new block containing transactions to be added to the blockchain, it must be validated by a selected person called a minor. This validation operation is called mining.

3.1.3. The Block Chains. Each block in the blockchain is linked to the previous block. This link is done by inserting the hash specific to the previous block. Therefore, the hash of each block includes not only its own hash but also the hash of the previous block. Figure 1 illustrates what has been described. This way we can protect the blockchain from any form of corruption.

3.1.4. Smart Contracts. A smart contract is a software "installed" on a blockchain solution. It is the most important link in the blockchain. It runs automatically as soon as the various preprogrammed constraints are checked. Even though it is not a legal document, the intelligent contract automates the execution of a contractual commitment.

A consensus algorithm is a process through which all the nodes of the blockchain network achieve a common agreement about the actual state of the distributed ledger [26]. A well-designed consensus protocol can ensure the fault tolerance, authenticity, and security of a blockchain system.

3.1.5. Ethereum Consensus Algorithm. The Ethereum consensus is based on the Ethash algorithm, also known as the Dagger Hashimoto algorithm. The simplified diagram [28] described in Figure 2 represents this algorithm structure and particularly the main one [29].

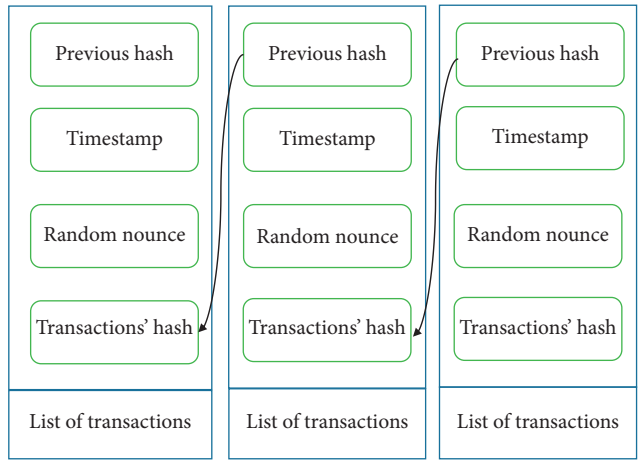


Figure 1: Blockchain illustration.

The profiling of the Ethash algorithm shows that the most used and consuming part is the Keccak 256 part. Therefore, we will implement this part in HW.

4. From SW to HW Architecture

We notice that the implementation of new technologies (IoT, identification, recognition, virtual reality, etc.) is no longer carried out on traditional platforms (PCs, servers, GPUs, etc.) but on embedded systems that can be either generic or well-tailored to the specific requirements of these emerging applications.

To set up a customized solution, it is important to use a mixed SW/HW design allowing adequate mixture of programmability and computing power.

Unlike the development of computer-based software and systems, which is very resource-intensive, the implementation of a System-on-Chip is based on a specific methodology to meet the limitations imposed by the target platforms. In this section, we will characterize the methodology used to realize the design flow of system-on-chip.

The development can be carried out according to several models. The V model presents the development cycle of a system.

This approach is based on two axes:

An axis of specification and design: this axis has as a parameter realization time

An axis of realization and integration: its parameters are the systems and components

Starting from a defined need, the first stage, which is the specification stage, consists of defining the system to be generically realized and then specifying the performances to be respected. Then, the design stage must be implemented. As for the specification, the design is based on two parts: a first generic followed by a second one which is detailed and during which the system is subdivided into different blocks. This conceptual approach leaves room for the realization of the components of our system.

Once the system realization part is completed, a battery of tests is necessary before obtaining our finished product. We start

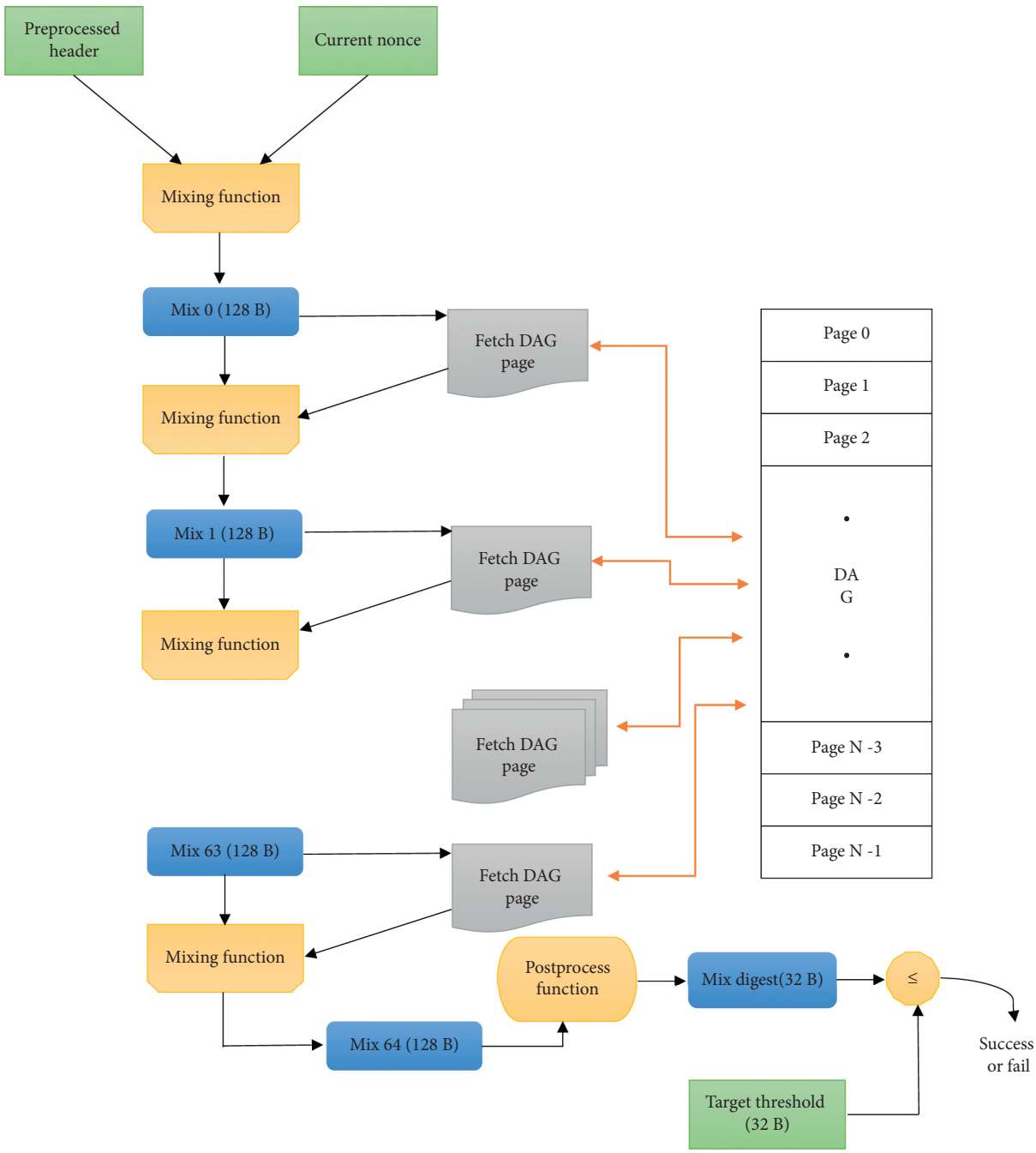


Figure 2: Flow diagram of the Ethash algorithm used by Ethereum with a DAG size of 2.37 GB of late 2018 [27].

with unit tests to verify the functioning of the previously defined blocks. Then, an experiment of integration of these different blocks is carried out. After that, a performance verification is set up to meet the specification presented in the first part. Then, the system integration is done for validation. Finally, an operational test is carried out to verify compliance with the expected specification. This being completed, our product is finalized. It thus meets the need defined previously [21, 30].

5. Embedded System Fields of Application

The use of embedded systems emerges in several fields such as agriculture, industry 4.0, smart cities, and e-health. To design efficient embedded architectures for blockchain applications, we need to profile the consensus algorithm to design an architecture on the FPGA platform. It is possible to have as a result a monoprocessor or a multiprocessor architecture. Different tasks are subdivided on processors during program execution.

In other systems, it is possible to have a monoprocessor architecture with coprocessors (also named IPs). These coprocessors are designed using a HW language such as VHDL, Verilog, System Verilog, and System C.

Such an approach was used for example in the study by Frikha et al. [31], where the authors implemented an adaptive multimedia multiconstraints' system based on

dynamic reconfiguration on FPGAs with augmented reality as a case study. In the work by Boutekkouk [32], the author presented the design of an intelligent embedded system. This system can be used in many artificial intelligence-based systems such as expert systems, neural networks, and other sophisticated artificial intelligence (AI) models to guarantee some important characteristics such as self-learning, self-optimising, and self-adaptation.

Among the embedded systems' application fields, we can also mention smart cities [33], smart agriculture [34], and e-health [35]. All these fields based on IoT use embedded systems mainly for their adaptability in designing systems with low energy consumption.

In this paper, we choose a monoprocessor system coupled with hardware accelerators that executes the most complex part of the application. Using the same approach proposed in the study by Frikha et al. [31], we profiled the consensus algorithm proposed by Ethereum. Thanks to this profiling, we will implement the best architecture to minimize the resources and improve the SW execution time.

This will allow us to choose the best possible architecture. We propose to implement an embedded architecture for the Ethereum hash algorithm. This algorithm named Ethash is a SHA 3. The implemented part is the Keccak 256 algorithm.

To the best of our knowledge, this blockchained approach has not been previously implemented. Additionally, the key idea of the work is to address the problem of important energy consumption of public blockchains.

6. The Proposed Consensus Embedded Architecture

Since the PoW consensus algorithm is the most time-consuming and energy-intensive part of the blockchain process, the aim of this paper is to reduce its execution time.

This proposed approach is based on a mixed on-chain and off-chain implementation. Only one part of the implementation (on-chain part) is connected to the blockchain. The other part (off-chain part) is connected directly to the on-chain part, and it is responsible for giving the consensus result.

More precisely, the PoW consensus and, more specifically, the part of the Keccak 256 algorithm on FPGA will do the off-chain encryption.

Inspired by Baklouti and Abid [25], we have set up this system to implement the PoW consensus and more specifically the part of the Keccak 256 algorithm on FPGA to do the off-chain encryption.

Keccak 256 is a part of Ethash which is the consensus of the PoW repetition.

Figure 3 represents the Keccak deployment architecture.

In this section, we are going to compare the software implementation and the hardware implementation of the Keccak hash algorithm. After profiling, Keccak is the more complex, energy consuming, time consuming, and repeated function.

As input of the Keccak system, we have the proposed new block, the head of the most recent block, and finally the nonce value. The hash and the combination of different blocks give a hash number. If this number is less than the target value, then we solved the PoW, else we must increment with a new nonce value and try the whole process again.

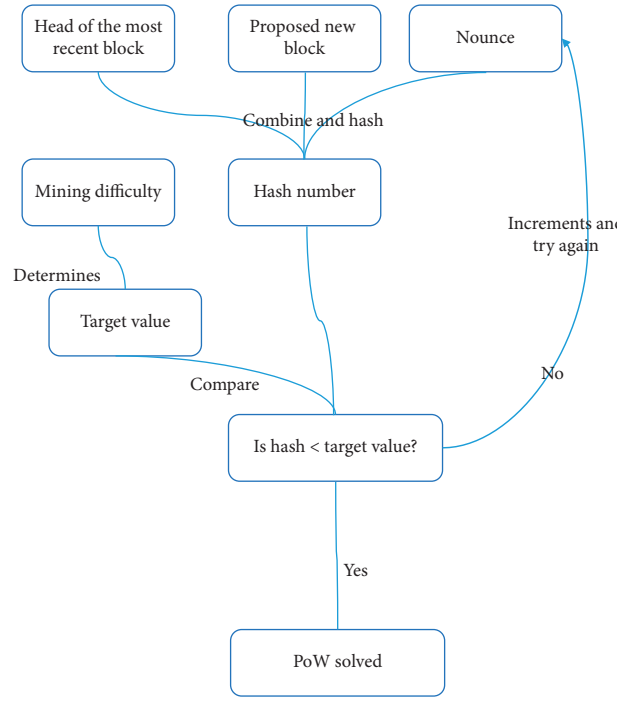


Figure 3: Keccak implementation algorithm.

The mining difficulty was determined by comparing the hash number and the target value. As mentioned in the work by Chaari [26], the implementation of the blockchain Ethereum node on a resource-constrained platform such as the Raspberry PI3 shows that the implementation of PoW leads to the platform crash.

As a first contribution, we present here the study we carried out in order to divide our node on two parts: a node without PoW that works on-chain: it runs on the ARM processors of the Raspberry Pi 3, and an off-chain verification part implemented on FPGAs.

In the following section, we will describe the obtained results and the implemented system.

7. Experimental Results

7.1. Initial System. After writing our genesis file and running the init command on the Raspberry Pi 3, the initialization of our blockchain was successful. Then, we were able to execute the node and access the JavaScript console where we performed some basic ether transfer transaction between the predefined accounts which were successfully submitted. But the moment the mining is being started, the Raspberry Pi 3 would overheat and stopped functioning. For that, we executed another node from the same blockchain on a computer that was able to mine the transactions and synchronize the results with the node running on the Raspberry Pi 3 as illustrated in Figure 4. Therefore, using Proof of Work, a Raspberry Pi3 can only synchronize the mined blocks but not mine new ones. That is why we decided to implement consensus system off-chain.

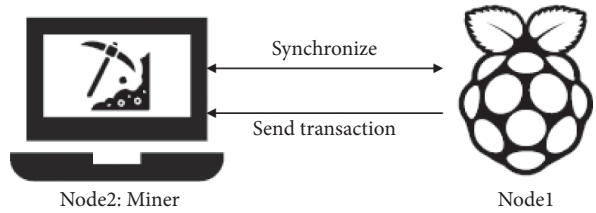


Figure 4: Private Ethereum blockchain using PoW consensus.

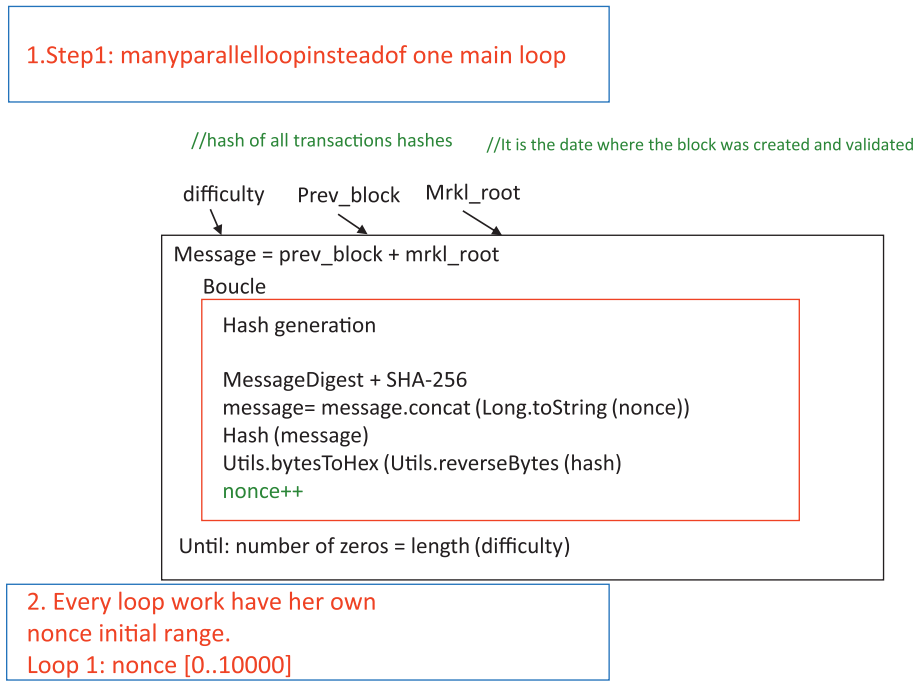


Figure 5: SW profiling result.

7.2. Keccak FPGA Implementation

7.2.1. Code Profiling Result. By taking the code implemented in the Java language related to the Ethereum node, we managed to isolate the part corresponding to the PoW consensus. This code has also been profiled to obtain the result of Figure 3. The result of this profiling is described in Figure 5.

Several loops are present: the relative loop to the nonce is repetitive and independent of any other input. We can consequently implement any VHDL system and create several generators of nonce values.

7.2.2. VHDL Keccak Implementation. Due to the health crisis and the impossibility to have more performant platforms, we choose to use the available ones. Henceforth, we use the Raspberry Pi 3. For the ZedBoard, we can explain it to its outperformance compared to the Virtex 5 ML 507 one. The implementation of the Keccak code in VHDL has been done to create an ASIC allowing the working off-chain to do the hash and to set up the PoW consensus. We used the Xilinx ZedBoard FPGA as a prototyping platform to realize the Keccak [29]. This board is an evaluation and a development board based on the Xilinx Zynq 7000.

Combining a dual Cortex-A9 Processing System (PS) with 85,000 Series-7 Programmable Logic (PL) cells, the Zynq-7000 AP SoC can be targeted for broad use in many applications. The ZedBoard's robust mix of on-board peripherals and expansion capabilities make it an ideal platform for both novice and experienced designers [29].

To improve this system, we have added 4 independent IPs to generate the nonce values. As an example, in [0.10000] interval, we are able to allocate to the IPs 1, 2, 3, and 4, respectively, the intervals [0.249], [250.499], [500.749], and [750.1000].

Figure 6 represents the proposed architecture of Keccak RTL implementation architecture. It contains different inputs and outputs but also the logic gates, Fifo, Padder bloc, Hash bloc, and different RTL registers.

7.3. Simulation Results and Comparison

7.3.1. Simulation Results. After the simulation results of the Keccak RTL implementation, the VHDL code simulation is proposed in Figure 7. The value of nonce to obtain the hash value is indicated in the figure by the arrow. Note that the nonce value used to obtain the proper hash is 239327.

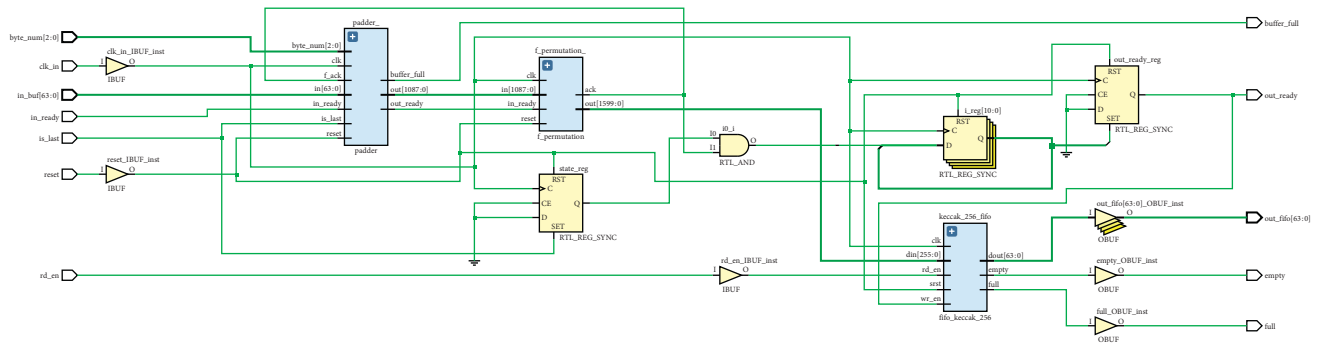


Figure 6: KECCAK RTL implementation.

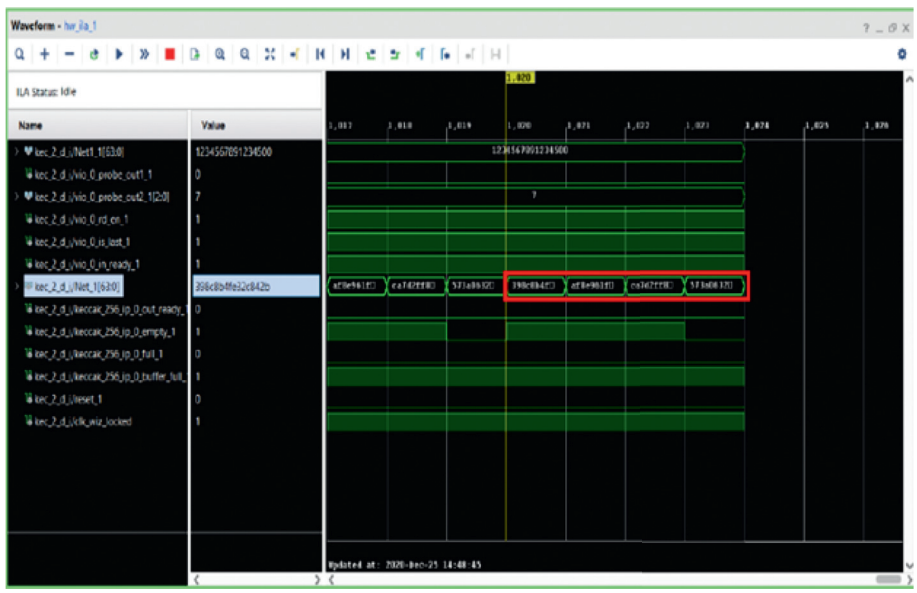


Figure 7: KECCAK simulation result.

Table 1: HW/SW comparison.

	SW	HW1	HW2
Execution time (ms)	21	3.98	2.78
Energy consumption (W)	3.7	1.2	1.7

7.3.2. SW and HW Comparison. After implementing the code, we tried to compare the SW version of the code implemented in Java running on Raspberry PI 3 and the two architectures. The HW1 architecture represents the complete implementation on the Keccak code presented in Figure 3. HW2 consists of using 4 nonce-generating IPs working in parallel in order to parallelize the code and to minimize the execution time.

We notice that the HW1 gain compared to the SW is approximately 5.25x. The HW2 gain compared to the SW is approximately 7.55x. The energy consumption on the Raspberry PI 3 is 3.7 W; however, in the HW version, we note that the HW1 requires 1.2 W, while the second requires 1.7 W.

The difference in consumption despite the use of the same platform (ZedBoard) for the HW1 and HW2 is due to the duplication of the IPs of nonce generators.

Table 1 illustrates the obtained result of HW/SW comparison.

The system obtained after this implementation is described in the figure. We can find there a description of the classical architecture of Ethereum followed by the on-chain/off-chain architecture that has been adopted.

Figure 8 presents the proposed part in the paper with an on-chain architecture implemented on Raspberry Pi3, whereas the offchain one is set up on FPGA.

Figure 9 represents a comparison between the classical blockchain architecture and the proposed architecture.

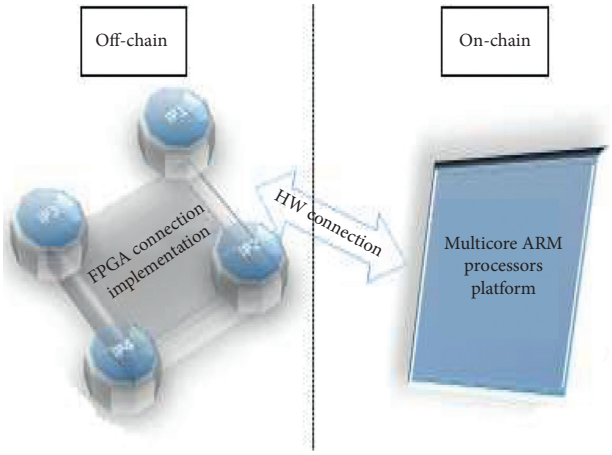


Figure 8: Mixed architecture for PoW algorithm implementation.

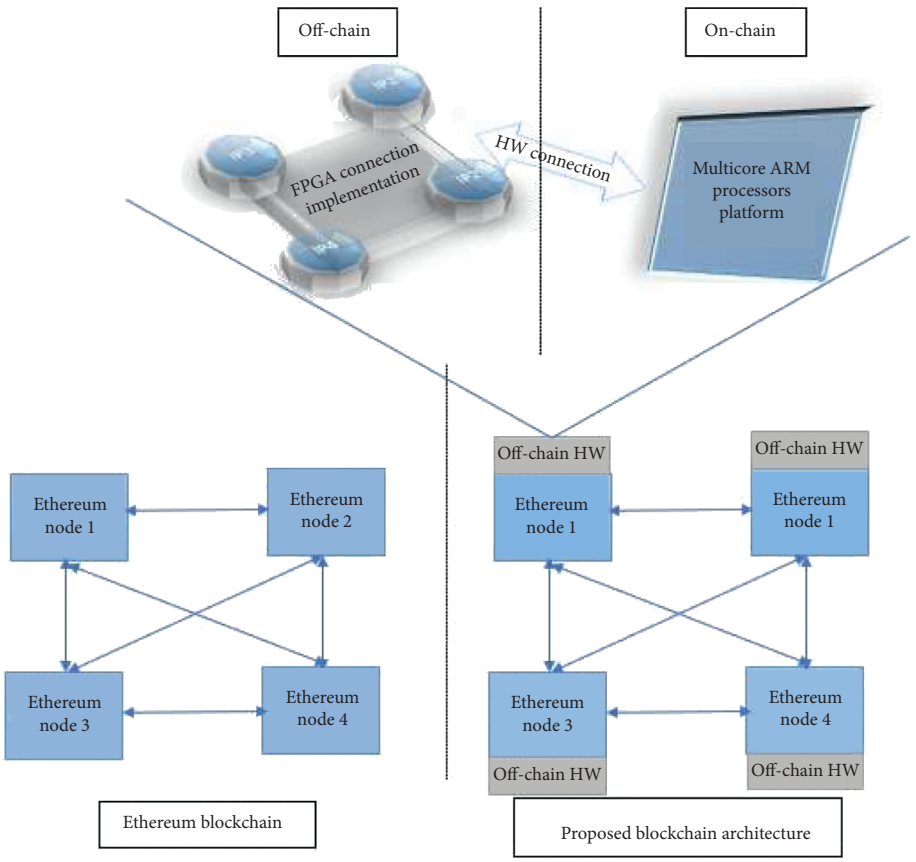


Figure 9: Whole proposed architecture implementation.

8. Conclusion

In this paper, we have highlighted the HW implementation of the PoW consensus. This consensus is used in the Ethereum blockchain. We were able to demonstrate that, to successfully implement this consensus on low-resource platforms, it is possible to use an on-chain system to successfully transfer and receive data and an off-chain system to implement the consensus and send the result to the on-chain node. This system, despite its complexity, allows a gain of at least 5 times compared to a pure SW system in execution time, while minimizing energy consumption. It can also be improved and accelerated by playing on the different blocks of the consensus. Indeed, we have added 4 IPs of nonce generators, but we could improve the result even more by adding more Keccak 256 and or 512 IPs to have a more efficient and faster system.

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

Dr. Omar Cheikhrouhou thanks Taif University for its support under the project Taif University Researchers Supporting Project (no. TURSP-2020/55), Taif University, Taif, Saudi Arabia.

References

[1] H. Treiblmaier, A. Rejeb, and A. Strebinger, "Blockchain as a driver for smart city development: application fields and a comprehensive research agenda," *Smart Cities*, vol. 3, no. 3, pp. 853–872, 2020.

[2] G. Bloom, B. Alsulami, E. Nwafor, and I. C. Bertolotti, "Design patterns for the industrial internet of things," in *Proceedings of the 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 1–10, Imperia, Italy, 2018.

[3] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *SSRN Electronic Journal*, 2008.

[4] A. Polyviou, P. Velanas, and J. Soldatos, "Blockchain technology: financial sector applications beyond cryptocurrencies," *Proceedings*, vol. 28, no. 1, p. 7, 2019.

[5] C. Agbo, Q. Mahmoud, and J. Eklund, "Blockchain technology in healthcare: a systematic review," *Healthcare*, vol. 7, no. 2, p. 56, 2019.

[6] V. Paliwal, S. Chandra, and S. Sharma, "Lockchain technology for sustainable supply chain management: a systematic literature review and a classification framework," *Sustainability*, vol. 12, 2020.

[7] J. Lee, M. Azamfar, and J. Singh, "A blockchain enabled Cyber-Physical System architecture for Industry 4.0 manufacturing systems," *Manufacturing Letters*, vol. 20, pp. 34–39, 2019.

[8] G. Pau, M. Collotta, A. Ruano, and J. Qin, "Smart home energy management," *Energies*, vol. 10, no. 3, pp. 382–386, 2017.

[9] G. Mirabelli and V. Solina, "Blockchain and agricultural supply chains traceability: research trends and future challenges," *Procedia Manufacturing*, vol. 42, pp. 414–421, 2020.

[10] K. Cho and Y. Cho, "HyperLedger fabric-based proactive defense against inside attackers in the WSN with trust mechanism," *Electronics*, vol. 9, 2020.

[11] N. Wang, X. Zhou, X. Lu et al., "When energy trading meets blockchain in electrical power system: the state of the art," *Applied Sciences*, vol. 9, 2019.

[12] D. Siswantoro, R. Handika, and A. F. Mita, "The requirements of cryptocurrency for money, an Islamic view," *Heliyon*, vol. 6, no. 1, 2020.

[13] B. Tavares and F. Figueiredo Correia, "A survey on blockchain technologies and research," *Journal of Information Assurance and Security*, vol. 14, pp. 118–128, 2019.

[14] K. Christodoulou, E. Iosif, A. Inglezakis, and M. Themistocleous, "Consensus crash testing: exploring ripple's decentralization degree in adversarial environments," *Future Internet*, vol. 12, 2020.

[15] M. Salimitari, M. Chatterjee, and Y. P. Fallah, "A survey on consensus methods in blockchain for resource-constrained IoT networks," *Internet of Things*, vol. 11, 2020.

[16] https://www.thebalance.com/how-much-power-does-the-bitcoin-network-use-391280#:%7E:text=Regardless%20of%20the%20number%20of,usage%20provided%20by%20ASIC%20miners.

[17] https://zipmex.com/learn/how-long-to-mine-ethereum/#:%7E:text=Successful%20mining%20on%20the%20Ethereum,a%20block%20of%20Bitcoin%20transaction.

[18] Y. Sakakibara, Y. Tokusashi, and H. Matsutani, "Accelerating blockchain transfer system using FPGA-based NIC," in *Proceedings of the 2018 IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications*, pp. 171–178, Melbourne, Australia, December 2018.

[19] Z. Tu and C. Xue, "Effect of bifurcation on the interaction between Bitcoin and Litecoin," *Finance Research Letters*, vol. 31, 2019.

[20] S. Bamakan, A. Motavali, and A. Bondarti, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Systems with Applications*, vol. 154, 2020.

[21] T. Frikha, H. Choura, N. Abdennour, O. Ghorbel, and M. Abid, "ESP2: embedded smart parking prototype," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 6, pp. 1569–1576, 2020.

[22] P. Marwedel, *Embedded System Design: Embedded Systems, Foundations of Cyber-Physical*, Springer International Publishing, Berlin, Germany, 2018.

[23] T. Noergaard, "Embedded systems architecture: a comprehensive guide for engineers and programmers," in *Embedded Systems Architecture*, pp. 261–293, Elsevier Science, Amsterdam, Netherlands, 2013.

[24] T. Frikha, N. Ben Amor, J.-P. Diguet, and M. Abid, "A novel Xilinx-based architecture for 3D-graphics," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14947–14970, 2019.

[25] M. Baklouti and M. Abid, "Multi-softcore architecture on FPGA," *International Journal of Reconfigurable Computing*, vol. 2014, Article ID 979327, 13 pages, 2014.

[26] A. Chaari, "*Storing health and fitness data on an ethereum based blockchain*," M.S. thesis, National Engineering School of Sfax, Sfax, Tunisia, 2019.

[27] Unknown, ETHASH., 2018, https://miningbitcoinguide.com/mining/sposoby/ethash.

[28] M. Stachowski, A. Fiebig, and T. Rauber, "Autotuning based on frequency scaling toward energy efficiency of blockchain algorithms on graphics processing units," *The Journal of Supercomputing*, vol. 77, no. 1, pp. 263–291, 2020.

[29] Xilinx, Zeadboard User Guide, 2013..

[30] S. Falcone, J. Zhang, A. Cameron, and A. Abdel-Rahman, "Blockchain design for an embedded system," *Ledger*, vol. 4, no. 1, 2019.

[31] T. Frikha, N. Ben Amor, K. Lahbib, J. P. Diguet, and M. Abid, "A data adaptation approach for a HW/SW mixed architecture (case study: 3D application)," *WSEAS Transactions on Circuits and Systems*, vol. 12, no. 9, pp. 263–272, 2013.

[32] F. Boutekkouk, "Embedded systems codesign under artificial intelligence perspective: a review," *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, vol. 32, no. No. 4, 2019.

[33] A. Kumar and V. Nath, "Study and design of smart embedded system for smart city using internet of things," in *Nano-electronics, Circuits and Communication Systems*, V. Nath and J. Mandal, Eds., Springer, Singapore, Singapore, 2019.

[34] M. Mahbub, "A smart farming concept based on smart embedded electronics, internet of things and wireless sensor network," *Internet of Things*, vol. 9, 2020.

[35] A. J. Bokolo, "Application of telemedicine and eHealth technology for clinical services in response to COVID-19 pandemic," *Health and Technology*, vol. 11, no. 2, pp. 359–366, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6677896, 13 pages
https://doi.org/10.1155/2021/6677896

Research Article

V2X-Based Mobile Localization in 3D Wireless Sensor Network

Iram Javed,[1] **Xianlun Tang**,[1] **Kamran Shaukat**,[2] **Muhammed Umer Sarwar**,[3] **Talha Mahboob Alam**,[4] **Ibrahim A. Hameed**,[5] **and Muhammad Asim Saleem**[6]

[1]*School of Computer Science and Technology, Chongqing University of Post and Telecommunication, Chongqing, China*
[2]*School of Electrical Engineering and Computing, The University of Newcastle, Callaghan, Australia*
[3]*Department of Computer Science, Government College University, Faisalabad, Pakistan*
[4]*Department of Computer Science, University of Engineering and Technology, Lahore, Pakistan*
[5]*Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Trondheim, Norway*
[6]*School of Information and Software Engineering, University of Electronic Science and Technology, Chengdu, China*

Correspondence should be addressed to Iram Javed; iram.javed1@hotmail.com and Ibrahim A. Hameed; ibib@ntnu.no

Received 3 December 2020; Revised 11 January 2021; Accepted 27 January 2021; Published 11 February 2021

Academic Editor: Shehzad Chaudhry

In a wireless sensor network (WSN), node localization is a key requirement for many applications. The concept of mobile anchor-based localization is not a new concept; however, the localization of mobile anchor nodes gains much attention with the advancement in the Internet of Things (IoT) and electronic industry. In this paper, we present a range-free localization algorithm for sensors in a three-dimensional (3D) wireless sensor networks based on flying anchors. The nature of the algorithm is also suitable for vehicle localization as we are using the setup much similar to vehicle-to-infrastructure- (V2I-) based positioning algorithm. A multilayer C-shaped trajectory is chosen for the random walk of mobile anchor nodes equipped with a Global Positioning System (GPS) and broadcasts its location information over the sensing space. The mobile anchor nodes keep transmitting the beacon along with their position information to unknown nodes and select three further anchor nodes to form a triangle. The distance is then computed by the link quality induction against each anchor node that uses the centroid-based formula to compute the localization error. The simulation shows that the average localization error of our proposed system is 1.4 m with a standard deviation of 1.21 m. The geometrical computation of localization eliminated the use of extra hardware that avoids any direct communication between the sensors and is applicable for all types of network topologies.

1. Introduction

The advancement in electronic systems and wireless communication makes wireless sensor networks (WSNs) a great asset for the Internet of Things (IoT). This development provides a way to design a low-power, low-cost, and tiny sensor module, with the ability to process data, sense physical structure, and provide communication within the networks [1–4]. A sensor node consists of a processor, sensing hardware, a transceiver, and a power supply. A node always has limited energy, limited computational power, and memory; thus, a large distributed WSN is required to accomplish a specific task [5]. In WSNs, a lot of sensor nodes operated on a battery are deployed over a sensing region. All of these sensors are used to collect data from other nodes and measure the ecological conditions such as sound, temperature, and vibration. The process data is then forwarded to the base station depending on the system requirements [6, 7].

In the past few decades, the use of IoT became more popular, which allows the user to transfer information from many applications. WSNs also gain much popularity in many applications because of their cost-effective nature. Such applications include military applications [8, 9], civil process monitoring [10, 11], habitat environment monitoring [12–14], health applications [15], home automation [16, 17], and vehicular networks [18]. Hence, WSNs provide a new way for proactive computing by getting real-time data from the physical environment. Depending on the system structure and application requirement sensor position is

required to be known. For this purpose, a sensor node must contain a positioning device [3]. But this is not possible in some scenarios, so a localization system should be able to self-localize and provide an accurate solution without the use of any extra hardware. Localization is a challenging issue for many applications especially for mobile-based WSNs where path planning is the main concern. An extensive research work has been placed for static and 2D based systems, but none of the algorithms provides an accurate solution for mobile-based WSNs. One de facto solution is the Global Positioning System (GPS), but due to its strict requirement of the line of sight scenario, GPS is not a perfect solution [19]. Furthermore, GPS is not always available and also not working in indoor environment. Furthermore, the cost of GPS receiver is very high, and it consumes more power than a tiny sensor module. Moreover, a mobile node always keeps changing its position so equipping a positioning device does not make it feasible to accomplish its task in a given time span.

A wireless sensor network has four basic components including (1) localized or distributed nodes; (2) a wire/wireless interconnected network; (3) an information cluster located in a central point; and (4) a set of application systems to process correlation data. Doubtlessly, the main computation is mostly done within the network, because of the large amount of data, algorithms, and techniques implemented within the system. Most of the sensor network applications require measuring the position of the sensor nodes so every system requires an algorithm which is free from extra hardware and communication cost. Furthermore, a localization system should be able to self-localize and calibrate in case of any environmental changes. A lot of WSN localization algorithms are proposed in the literature under range-based and range-free localization categories [1]. It is to be noted that range-based solutions provide high accuracy which use the distance information from the neighbor nodes. A simple example of such a system is triangulation that uses three further nodes to form a triangle. In a range-free system, the information of absolute distance is not available; instead, the node position is determined through radio connectivity information. Range-free approaches mostly used anchor node deployment for geometrical positioning of the sensor nodes which also provide a low-cost solution. Doubtlessly, each algorithm has its own merits and demerits; its main target is to estimate the node position with accuracy and high efficiency. The flow diagram of the range-based and range-free localization technique is presented in Figure 1 taken from [20].

The range-based method on the other side used extra hardware that not only increased the cost but also increased the communication and computation overhead. Distance measurement approaches which include time of arrival (ToA) [21], angle of arrival (AoA) [22], time difference of arrival (TDoA) [23], and received signal strength (RSSI) [24] are the best examples of range-based localization algorithms. A brief comparison between range-based and range-free localization algorithms is presented in Table 1.

In the literature, most of the authors have discussed 3D-based static network localization. However, only a few ideas are available which address the localization of sensor nodes in mobile environment. Therefore, there is a significant need for some automated process to help discover, identify, and locate the mobile nodes within an indoor facility after initial deployment. This motivates us to propose a new localization algorithm consisting of static sensor nodes and flying anchor nodes which helps to locate the position of the sensor nodes even in a changing environment. The self-calibration factor and the use of LQI values make our system more robust and efficient. The rest of the paper is organized as follows. Section 2 discusses the state of the artwork done in WSN localization along with the problem statement and detailed algorithm description. Section 3 presents the simulation results, computational complexity, and lower bound of the proposed localization algorithm. Section 4 concludes the paper with possible future work.

2. Materials and Methods

2.1. Related Work. The localization of mobile ad hoc system is very problematic and challenging because it is almost impossible to install an infrastructure with a tiny sensor module especially in a dense environment. Various ad hoc applications also require deploying some anchor nodes with a known position. The aim of this work is to design a localization algorithm based on a mobile anchor node flying on a C-shaped path. To do so, a mobile anchor node always requires some sort of trajectories which is basically not our main consideration for this research. However, a fixed random walk will be considered for simulation purposes. Path planning may be either static or dynamic. In static based path-planning schemes, the execution is taken place before the real process whereas in mobile sensor always follows the predefined path in the entire localization algorithm. In a dynamic path-planning localization, trajectories are not fixed and predefined but particularly drawn according to the environmental conditions.

In a static based path-planning algorithm, the well-known trajectories are SCAN, DOUBLE SCAN, and HILBERT proposed in [25] and provide the high network coverage. A series of straight lines are proposed in SCAN and DOUBLE SCAN algorithms on which a mobile node is moved in the entire network. In a HILBERT algorithm, a curve is divided into 2D space and trajectories which are a mixture of vertical and horizontal path. The HILBERT algorithm identifies more beacons during the random walk as compared to SCAN and DOUBLE SCAN trajectories. In another work, two trajectories based on CIRCLE and S-CURVE are proposed [26]. A mobile anchor node is traveling on a circle to cover the entire network. The disadvantage of this scheme is that if the node is not lying in the circle, it may not be localized, and this leads to the network coverage problem. S-CURVE is much similar to SCAN algorithm due to its fixed trajectory and random walk. The structure of S shape also makes it very similar to SCAN algorithm. Furthermore, the authors in S-CURVE algorithm also try to reduce the problem of collinearity during the localization process. In another algorithm, five different trajectories were proposed in [27]. The trajectories

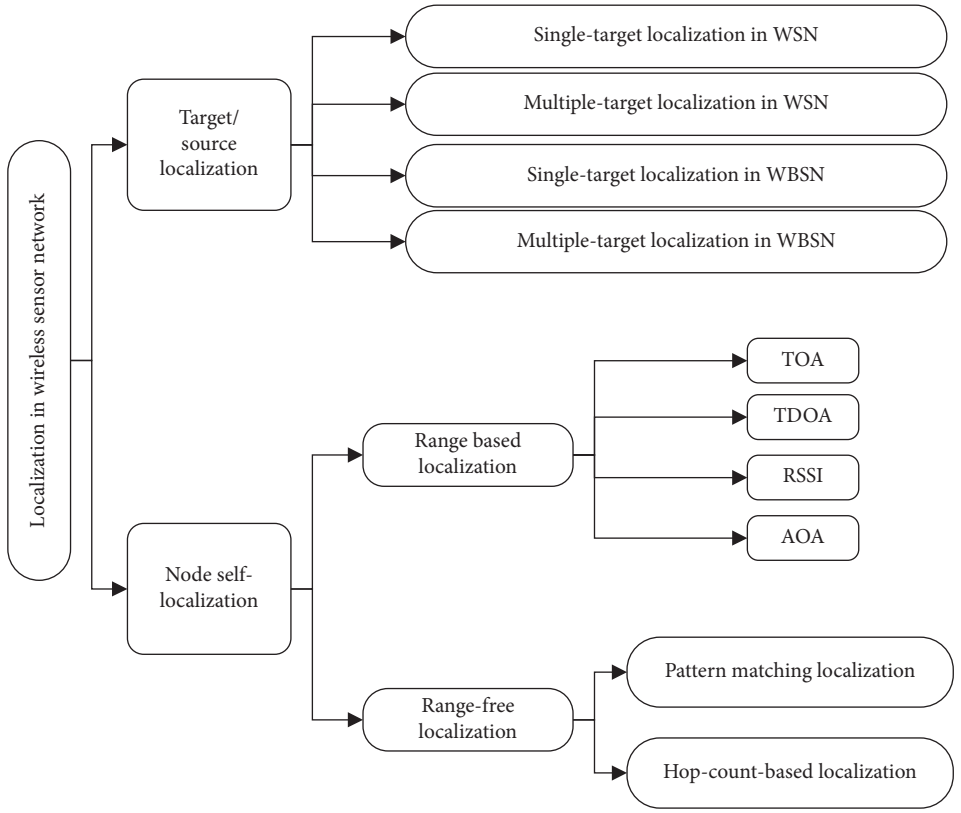


Figure 1: A flow diagram of localization algorithms.

Table 1: Comparison between range-based and range-free localization algorithm.

Algorithm types	Measurement methods	Localization error	Hardware cost
Range-based	AOA	High	High
	TDoA	High	High
	RSSI	Medium	Low
Range-free	Hop-based techniques	Medium	Medium
	Single neighbor	Low	Low
	Multineighbor	Low	Low

LAYERED-SCAN and LAYEREDCURVE divided the 3D region into different layers. The mobile node travels in the form of SCAN and DOUBLE SCAN format but within the different layered. A variety of other path-planning algorithms were proposed based on dynamic trajectories for the real distribution of WSNs. An undirected graph is generated in [28] based on the artificial intelligence scheme Breadth-First (BRF) algorithm that transforms mobile walk into the form of spanning tree structure. The movement of the node is dynamic and the mobile node has several different walks that even increase the computation of localization algorithm. The detailed summary of the literature review is presented in Table 2.

2.2. Problem Statement and Assumptions. Let us consider that a set of nodes are deployed on a sensing region where some nodes are static, which are going to localize with the help of mobile anchor nodes. Mobile nodes are the anchor nodes whose position is known with the help of GPS system.

Table 2: Summary of literature review.

Algorithm	Accuracy	Node density	Traveling speed	Energy
[8]	Average	Average	N/A	Low
[14]	Low	Average	High	High
[25]	Average	High	High	High
[26]	Low	High	Low	High
[27]	Average	High	Average	Low
[28]	High	Average	Average	High

In brief, a localization process in WSNs will be accomplished in several steps including the computation of distance, reconstruction of sampling distance matrix in presence of noise, and finally computing the localization error using some triangulation techniques.

In a proposed algorithm, we assume that the nodes are deployed on a nonoverlapped network. Let us assume that $\boldsymbol{N}$ number of sensor nodes are deployed on a 3D space and $\boldsymbol{M}$ number of mobile anchor nodes are moving in a C-shaped

trajectory. The reason to choose C-CURVE trajectory is that most of the network topologies are in this form so we do not fall into a network coverage problem [29]. The nodes have the same communication range as shown in Figure 2.

The system environment includes a fewer number of flying anchor nodes and a number of static sensor nodes deployed randomly. The flying anchor nodes continuously transmit a beacon with location information to unknown nodes in a network to estimate its location. The mobile anchor nodes are moving along a path that is assumed to be predefined. Assuming the uniform distribution of unknown nodes, that is, $\mathbb{R} = (N, M)$,

$$N = \{N_i(x_i, y_i, z_i), \quad \forall N_i \in \mathbb{R}\}, \tag{1}$$

where x_i, y_i, z_i are the coordinates of the unknown nodes. Similarly, j number of mobile anchor nodes are deploying on a fixed path and transmitting beacons continuously while on a random walk. The received signal strength (RSSI) is measured at each anchor node. The RSSI helps to determine the distance of each node from the anchor nodes and store the distance value in a matrix. The trilateration or triangulation method is then used to form a triangle that helps to measure the localization error by using the centroid formula. We assume that the mobile anchor nodes have enough power in the entire computation of localization error. Furthermore, the communication is in the form of spherical measurement that transforms the 3D computation in 2D form and hence makes the system very easy in computation.

$$\mathbf{M} = \{M_j(x_i, y_i, z_i), \quad \forall M_j \in \mathbb{R}, \ M_j \longrightarrow \rightharpoonup\}, \tag{2}$$

where $\rightharpoonup$ denotes the path for mobile anchor node. All the mobile anchor nodes are transferring some beacons in the form of signals. This signal basically includes the localization mobile anchor node. As the computation is in the form of spherical coordinates, so the distance between two unknown nodes is computed by

$$D_{x_i, y_i, z_i} = N_i(x_i, y_i, z_i) - N_0. \tag{3}$$

Let a node have three beacons from mobile anchor node at a particular location. Then the distance between N_1 and N_0 is 1. Therefore, RSSI also might go down in several points so the exact distance is computed by adding the error factor η; we have

$$D_i = (x, y, z) | (r_i - \eta_i)^2 \leq (x - \hat{x})^2 + (y - \hat{y})^2 + (z - \hat{z})^2, \tag{4}$$

where $\hat{x}, \hat{y}, \hat{z}$ are the estimated position of the sensor node. The distance between mobile anchor node and unknown node is computed by the Euclidean distance formula through equation (5). This is further illustrated in Figure 3.

$$\sum_{i=1}^{N} d(M, N) = \sqrt{(M_i - N)_i^2}. \tag{5}$$

The computation of received signal strength is crucial part of the localization process. We use the RSSI data given

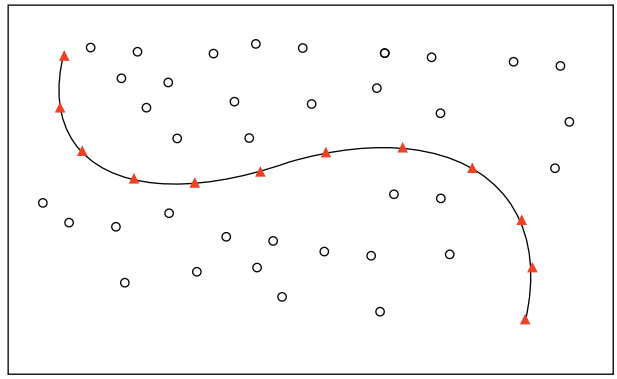


Figure 2: Sensor node deployment and network structure.

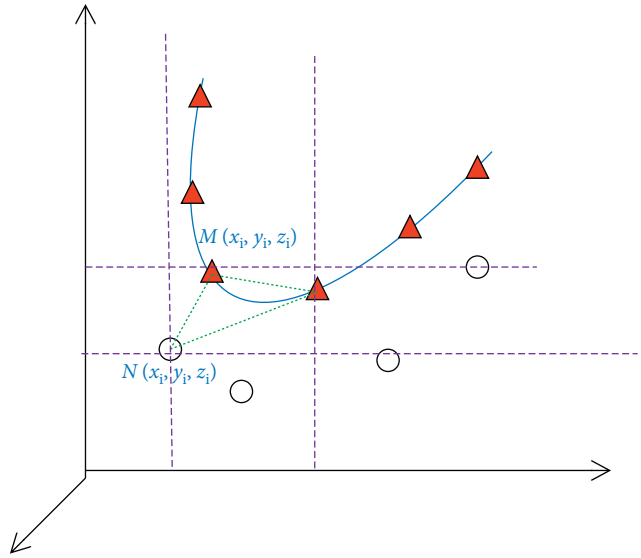


Figure 3: Distance measurement between mobile anchor and sensor node.

in GitHub repository [30]. The RSSI is calculated by the following formula [27]:

$$\text{RSSI}_{i,j} = P_T - P_{L_{i,j}} + \aleph_{i,j}, \tag{6}$$

where P_T dBm and P_L are a transmission power of device used in measurement phase and pathloss, respectively. Moreover, $\aleph$ represents the noise factor. Equation (3) shows that most power loss occurs at higher frequencies. This means antenna with specified gains; there will be the highest energy transfer in case of lower frequencies. Due to various signal path factors, the loss in the wireless communication path is different from equation (6). By merging the constants, adding losses, and using logarithmic power values, we have

$$d = d_0 \cdot 10^{(P_0 - P_L + E_\varphi)/10\eta}, \tag{7}$$

where d_0 is a reference distance corresponding to a reference transmission power P_0. The pathloss from the signal transmission is derived from [31]; we have

$$P_{L_{i,j}} = l_0 - 10n \log_{10}\left(\frac{d_{i,j}}{d_0}\right), \quad (8)$$

where l_0 (dB) is the reference pathloss value at $d_0 = 1$ (m). (1) (m). $\boldsymbol{N}$ is the pathloss exponent value for showing the environment characteristic. $d_{i,j}$ is the distance between nodes i and j in the 3D modeling system.

$$d_{i,j} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2}. \quad (9)$$

The LQI measures the error in the incoming modulation of successfully received packets. It is accessible via the Mgmt_{lqi} command, whose response gives the LQI values of all neighbors in table form [31]. Similar to RSSI, the LQI has the value range from {0...255} in which a higher value indicates higher link quality and vice versa. Also similar to RSSI, implementations can differ between manufacturers. Some manufacturers only make use of 7 bits, that is, {0...127}. Whereas RSSI provides raw signal strength without caring about link quality, LQI is essentially a measure of data throughput, and its values correlate very well with practical tests of data throughput [31], of which the received signal strength is but one factor. Without incorporating all factors that affect data throughput, for example, interference, noise, and radio characteristics, it is not possible to derive LQI theoretically. However, it is also reasonable to assume that in nonextreme environments, LQI should behave as a function of distance similar to RSSI. The following terms and definitions are used throughout the explanation of the algorithm.

Definition 1. A mobile anchor node is a node that travels over a trajectory with a known coordinate. The coordinates are determined by GPS or some positioning device.

Definition 2. An unknown node is a node whose position is not known and needed to compute by using the anchor node mobility.

Definition 3. A trajectory is a path used for static path planning in the entire network.

The localization scheme requires some basic assumptions.

(1) GPS is attached to all the flying anchors so that the system can identify its real location all the time

(2) In a particular environment, it is also possible that the positioning device provides some inaccuracies

(3) Anchor node is battery operated however equipped with sufficient energy to accomplish the task

(4) Antenna type of anchor is always an isotropic radiator so that it cannot consume much energy

2.3. Proposed Localization Algorithm Based on C-CURVE Trajectory. Assume that the deployment region is in the form of a square and a node is deployed randomly over the sensing region. As the scheme is for range-free algorithms so a region is divided into several networks, that is, $N\omega = (N\omega_1, N\omega_2, \ldots, N\omega_n)$, in which the networks are not overlapped like a well-known scheme of a point in triangulation (APIT) [32] in which all the nodes are adjacent on overlapped triangles. The overlapped triangles can increase the computation cost as well as energy. The volume of localization is $(1000 \times 1000 \times 1000)$ m, that is, $Vm = (Vm_1, Vm_2, \ldots, Vm_n)$. The distance between the networks is not necessarily computed because the mobile anchor node is moving along the path which is free from the square division. The vertex of the interested region is in the form of minimum and maximum coordinates. The initial coordinates of mobile anchor node are computed by the following formula:

$$(X, Y, Z) = \left|\frac{\left(x_{\max} - x_{\min}\right),\ \left(y_{\max} - y_{\min}\right),\ \left(z_{\max} - z_{\min}\right)}{2}\right|. \quad (10)$$

The node calculating the initial coordinates of mobile anchor node will act as a reference anchor node. This node always initiates the process in the entire processing of the algorithm. It is also possible that a reference anchor node does not respond due to low battery or node failure. In this case, a control can transfer to the next mobile anchor and select the other three anchor nodes to form a triangulation. This is the beauty of our proposed scheme that the control is transferred to the next level by calibration.

Definition 4. A node initiating the process of localization is said to be a reference anchor node. It may be a static or mobile node.

The entire algorithm can be completed in the following steps.

Step 1.A mobile reference anchor node travels on a static C-CURVE path and broadcasts a beacon node to all other sensor nodes in a communication region. The beacon node contains the ID and coordinates of mobile anchor node.

$$\tau(t) = \sum RAN\ D\left(x_{ri}(t),\ y_{ri}(t),\ z_{ri}(t)\right) \in \mathbb{R}, \quad (11)$$

where r denotes the communication radius and t is a time factor that denotes the dynamic nature of mobile anchor node in a particular time. τ is a function that models the trajectory with a random location "RAND."

Step 2. A mobile anchor node then starts estimating distance by sampling RSSI versus distance. At the end of each RSSI measurement, multiple values of RSSI are received because an unknown node keeps sending the RSSI signal until a mobile anchor node is in its communication range. A function of RSSI is formed here

that is recorded in a matrix form. Here, we introduce a method of link quality induction that is similar to RSSI, but only the successive RSSI introduced by IEEE 802.15.4. The linked quality induction is the highest value of RSSI recorded in the matrix; we have

$$D_{\text{est}} = \varepsilon(\text{RSSI}) = \sum_{i=0}^{3} M_i, \tag{12}$$

where D_{est} and ε are an estimated distance and mapping function for interpolation of linked quality induction value. In a net phase of RSSI computation, the volume computation is the main problem. At the reference mobile node, we recorded the node coordinates as x_0, y_0, z_0 in a square form. Initially, a mobile anchor node is moving from initial coordinates to half of the C-CURVE. According to the nature of the curve, a mobile anchor is moving on a positive radius, that is, $(1/2R)$. Here, a mobile node is needed to make a move to the second half of the curve. This is a tricky part as a mobile node is needed to know the exact position of the trajectory. In the second half of the C-CURVE, the curve side is also changed and given the value of $(-\sqrt{3}/2)$. The radius is negative that shows the length of the trajectory in a 3D plane. The random walk length is then computed by

$$L = \frac{1}{2}R + \frac{-\sqrt{3}}{2}, \tag{13}$$

where L is a localization region. We noticed that the localization region has a problem of network coverage. To overcome this problem, it is possible to change the trajectory in a deep form. For this, we add some constant value on the curve depth that falls over the bounded region. Another solution is to deploy enough anchor nodes on the boundary of the entire network. This also reduces the coverage problem while localizing all the static nodes in a 3D space.

Step 3. A matrix of all lined quality induction and RSSI values is generated. Now, we need to overcome the interpolation error that is due to having some extra RSSI values from one anchor node towards the sensor node. The compact variations are again generated by mapping the LQI's values from the above matrix.

$$\varepsilon\text{: RSSI} \longrightarrow = \sum_{i=1}^{N} N_i \text{LQI}^i. \tag{14}$$

Step 4. In the next step, we find the maximum of LQI's values from the LQI matrix; we have

$$\left[\varepsilon\text{: LQI}_{\text{imax}}\right] = \max\left(\sum_{i=1}^{N} \text{LQI}\right), \quad i \in [1, N]. \tag{15}$$

To compute the sensor node position, we need all four beacon points. Once we have all four beacon points, two cross sections are constructed. The triangulation is being considered on the circular cross section with their corresponding LQI values.

Step 5. The last step is to implement the centroid-based formula on the matrix of LQI to compute the estimated position of the form

$$\left[\varepsilon\text{: LQI}_{\text{imax}}\right] \longrightarrow E = \sum_{i=1}^{N} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2}, \tag{16}$$

where E is a localization error. The pseudocode of our proposed technique is given in Figure 4.

3. Results and Discussion

3.1. Simulation Setup. In this section, we can provide a detailed overview of the proposed algorithm. The simulation runs in Matlab and considers an area of $(1000 \times 1000 \times 1000)$ m in 3D space. The RSSI dataset is being taken from pspachos GitHub [32]. The GitHub data is being emulated in the NEST simulator first to rectify the RSSI signals. After refining the data, we have used these for simulation purposes. The number of mobile anchor nodes was fixed to 5 over 150 unknown sensor nodes. The simulation was run 2000 times to record an initial value of the localization error. The 2000 iterations produce 2000×5 numbers of anchor nodes. The simulation runs randomly, and the ratio of the localization is determined by the ratio of the number of localizable nodes to the total number of unknown sensor nodes. The localizable nodes are those nodes which are in the sensing range of the mobile anchor nodes. The unlocalized nodes will fall outside the area of network coverage. This can highlight the coverage degree of the path. Hence, the overall degree of a localized node is computed by

$$E_{\text{Ration}} = \frac{\partial(\longrightarrow N)}{N_i}, \tag{17}$$

where ∂ is a number of localized nodes from N sensor nodes and N_i is a total number of unknown nodes. The simulation is for 3D area that approximately covers the volume in a cubic meter. Furthermore, the accuracy will be determined by the ratio of localization error and communication radius, that is, $\lambda = E/r$. The accuracy of any localization algorithm can be measured using the standard deviation of unknown nodes. If the standard deviation of the localization error is smaller than the mean error values, which means that the data elements in the form of coordinates are scattered over a 3D space, this shows that the deployment of sensor node is very crucial in localization scenarios. Furthermore, in case of a mobile-based network, the design of trajectory is very important. Hence, the trajectories are always in a way that covers the entire localization network. The mean localization error is computed by the average and the standard deviation is computed by

Algorithm 1 proposed algorithm

Data: Beacons transmitted by mobile anchor nodes in $\Re$ communication radius
Result: Localization coordinates of unkown nodes (x_i, y_i, z_i), $i \in [1, N]$ initializatiom
While mobile anchor node random walk on C Curve shape, $M \rightarrow i = 1$ to N do
Step1 Keep searching for beacons
While mobile beacons $\neq 4$ do
Step2 Keep sensing radio frequencies in passive mode
$M \rightarrow$ ID, $(x_i, y_i, z_i) \in \Re$,
$\tau \rightarrow$: f(RAND) $\in M(x_i, y_i, z_i)$ d = RSSI $\rightarrow \sum M_i$
Step3 Adding the noise factor on each distance.
$L = 0.5(radius) + (-0.9)$ after manipulation
$\Im$: RSSI $\rightarrow = sumN_i, LQI$ from $i = 1 : N$
Step4 Compute the maximum of LQI values
Choose four beacon points to maintain a cross section area
Apply quadlateration on 4 non-collinear points
Form a triangulation using four beacon points.
Store all LQI values along with noise factor in a matrix. (M,N) $\rightarrow 1 : N$
Step5 while LQI matrix $\rightarrow 1 : N$ do le $= \Sigma \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2}(M, N) = 1 : N$
Calculate arithmetic mean of estimated coordinates

Figure 4: C-CURVE-based proposed algorithm.

$$\mu E = \frac{1}{\partial} \sum_{i=1}^{\partial} \lambda, \tag{18}$$

$$\sigma E = \frac{1}{\partial} \sum_{i=1}^{\partial} (\lambda - \mu)^2. \tag{19}$$

In our proposed algorithm of C-CURVE, the localizable points are determined on the basis of the number of successive LQI values. The successive LQIs are the countermeasure upon all the RSSI values. Therefore, the number of successive LQIs is the best approach to find the number of localized nodes. The path in our proposed scheme is in the form of a dual curve that almost covers the entire region of the network. However, some nodes at the edge of the graph might not listen to the mobile anchor call, which may produce some errors in the localization process. We found that the nodes closed to the paths are giving very nominal errors and due to having a curvy shape, most of the nodes are in the neighbor of the mobile path. This reflects the higher accuracy of the proposed system. Nodes within the sensing region also send multiple RSSI values until a mobile anchor is in the domain of the sensing region. However, most of the RSSI fail due to low battery and multipath effect. This is another reason for obtaining an LQI value instead of RSSI. As mentioned earlier, this work is based on simulation only, so we use the GitHub database for RSSI and consider the higher RSSI as an LQI from a node to mobile anchor. The following parameters are used for the initial simulation shown in Table 3.

3.2. Localization Error. In a 3D environment, each node has a localization cost and error; hence, in case of 3D system, we are interested in recording the average localization error. The mean localization error is recorded as 1.4 m, which is far better than SCAN, DOUBLE SCAN, and HILBERT

Table 3: Simulation parameters.

Parameter	Range
Area	$(1000 \times 1000 \times 1000)$ m
Mobile anchor speed	8 m/sec
Number of mobile anchor nodes	5
Number of unknown nodes	150
Communication radius	(100–150) m
Path loss exponent	2–5

algorithm. The mobile anchor node, true location of sensor, and estimated position are shown in Figure 5.

Considering that a node N_1 is close to the mobile anchor node at a certain period of time, the following RSSI were recorded. The LQI is then measured using the IEEE Mgmt request command as shown in Table 4.

The average localization error is computed along the entire axis with different communication radius. The number of mobile anchor nodes was fixed to check the authenticity of the algorithm for different radius. It is to be observed that the number of localization points is not decreasing but the weaker RSSI can affect the error for higher communication radius as shown in Figure 6 and the graph plot is shown in Figure 7. RSSI and LQI values represent the power level gain after loss in cable and at antenna. The higher RSSI values basically represent the stronger signal which is always required to form a triangulation between several sensor nodes. If the signal is weaker, the node cannot communicate with other nodes efficiently. Hence, it can be through a value that is beyond the boundary of the considered triangle. All the studies related to RSSI-based localization have difficulties to establish a link between signal strength and distance. In real deployment, it is challenging to map RSSI to the distance, and obtaining a fading function reflects that RSSI is affected by pathloss, shadowing, and fading, which also affect communication radius.

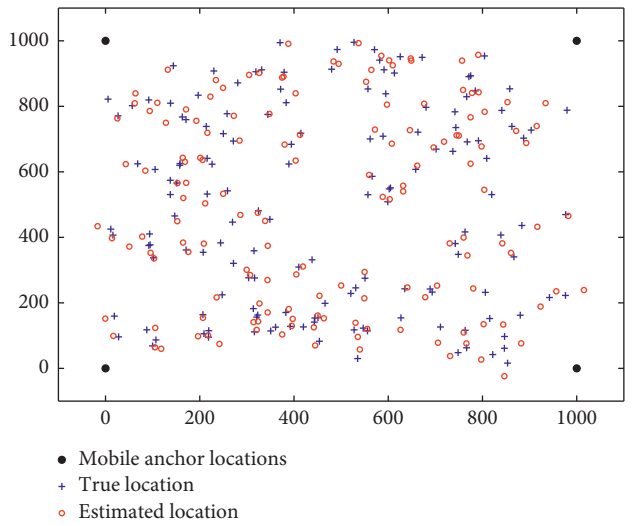


Figure 5: Position of mobile anchor, sensor node, and estimated node position.

Table 4: Sample LQI measurement.

Node	RSSI in upward direction	RSSI in downward direction	Successive LQI
N1	0.967319090984647	0.253562174262323	Mgmt LQI request
N2	−0.958798243408558	0.284087888577925	Mgmt LQI request
N3	0.524669344883845	−0.851306101551702	Mgmt LQI request
N4	−0.478708185913592	−0.877974072931153	Mgmt LQI request

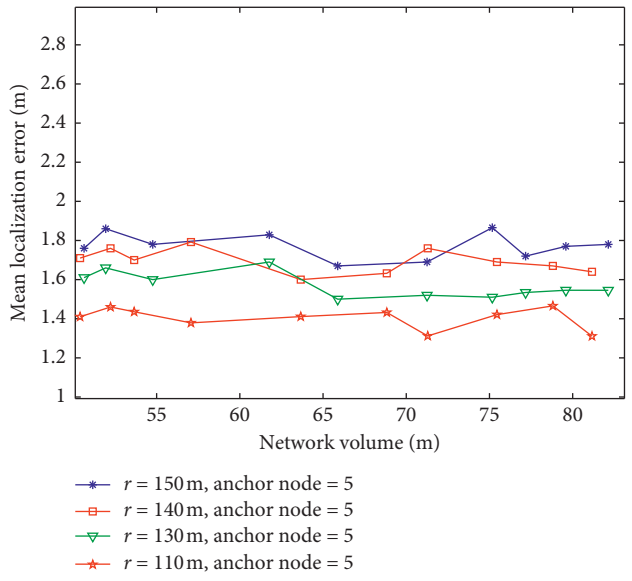


Figure 6: Average localization error in different region of the network with 5 anchor nodes.

3.3. Computation of Node Mobility and Network Coverage. The node mobility is always measured by the initial position and the relative change in the location of mobile anchor

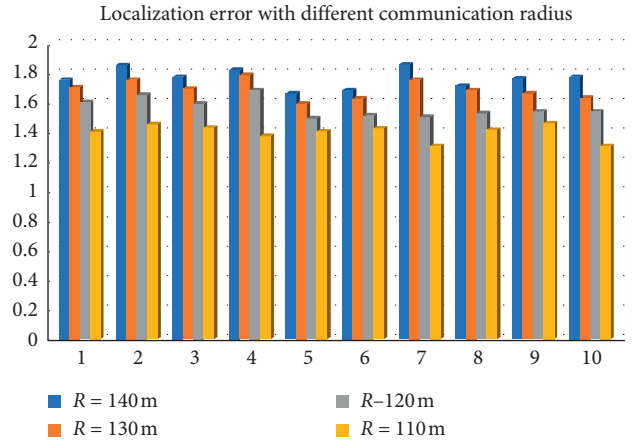


Figure 7: Localization error estimation with different network coverage.

nodes. With the concept of speed and velocity, the mobility of the mobile anchor node is measured along with the specific time. The addition of time variation subtracting with the average distance to the initial point is known as mobile node mobility; we have

$$M_i = \frac{1}{T - \Delta}. \tag{20}$$

The total sum of distance upon time is multiplied to initial mobility factor to compute the final movement of mobile anchor node.

$$M_f = \prod_{i=1}^{M} \sum_{t=0}^{T-\Delta t} \left| D_i(t + \Delta t) - D_i(t) \right|, \tag{21}$$

$$D_i(t) = \left(\frac{1}{n-1}\right) \prod_{i=1}^{M} \sum_{t=0}^{T-\Delta t} \left| D_i(t + \Delta t) - D_i(t) \right| - \sum D(N, M), \tag{22}$$

where T is a periodic motion for each mobile anchor and Δt is a change in time for each mobile node so that there should not be any collision between the nodes. Hence, the final mobility is measured by equations (5) and (22). We have

$$D_i(t) = \left(\frac{1}{n-1}\right) \prod_{i=1}^{M} \sum M_i. \tag{23}$$

The probability density function (PDF) can be used to define the accuracy of the C-CURVE method. To compute the PDF, the Rayleigh fading is considered to measure the localization error. The form of the Rayleigh is shown in Figure 8. In presence of Rayleigh fading with obstructions and reflection, the error might be affected because of multipath fading [33]. Another factor of time constant is associated with variations along with the mobile node movement in seconds or even in minutes. The effect of the carrier also plays an important role with $fc = 900$ MHz or approximately 1.9 GHz for the mobile node. The effect of localization error is further shown in Figure 9.

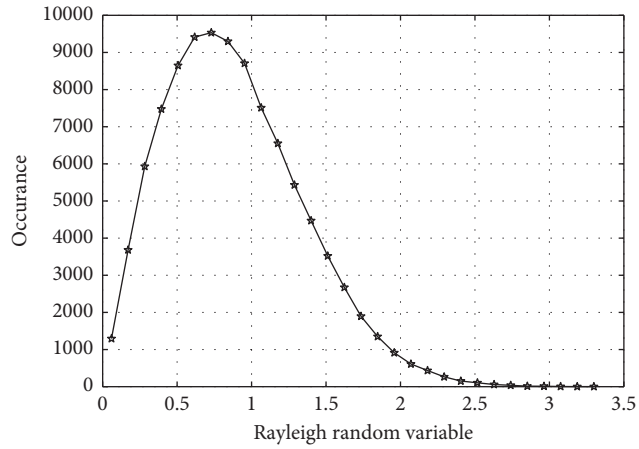


Figure 8: Rayleigh fading.

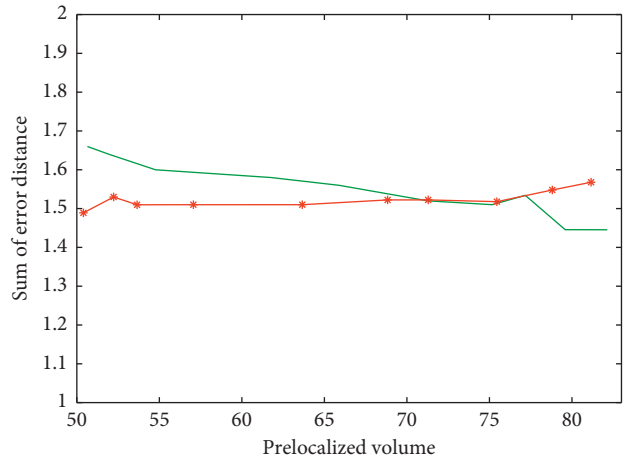


Figure 9: Localization error in presence of noise.

From Figure 9, we observed that the localization error is gradually increased in the presence of fading. The amount of data elements spreading along the region is very nominal. And it is possible to retain the correct error by using the framework like extended Kalman filtering and fuzzy logic [34]. The network lifetime and coverage in our proposed system are well described.

3.4. Comparison with Existing Techniques. To authenticate our proposed solution, we have compared our localization algorithm with the state-of-the-art localization techniques based on mobile anchor-based localization. The well-known methods SCAN, DOUBLE SCAN, and HILBERT are being compared with our C-CURVE localization algorithm. SCAN trajectories are very simple to compute with the path on some straight lines with parallel distribution along y-axis. However, in a round movement, the mobile anchor node lost enough energy in a given radius R. The distance in a SCAN is computed by the following formula:

$$D = \left(\frac{L}{R} + 1\right) \times L + \left(\frac{L}{R}\right) \times R = \left(\frac{L}{R} + 2\right), \tag{24}$$

whereas for the DOUBLE SCAN and HILBERT algorithm, the distance is computed by the following relationships:

$$D = 2\left[\left(\frac{L-R}{2R} + 1\right)\right] \times L + \left(\frac{L-R}{2R}\right) \times 2R, \tag{25}$$

$$D = 4^n \times R = \left(\frac{L}{R}\right)^2 \times R = \frac{L^2}{R}. \tag{26}$$

The SCAN, DOUBLE SCAN, and HILBERT are simulated on the area of (450×450) m, whereas we simulate our idea for $(1000 \times 1000 \times 1000)$ m 3d space. Hence, the suitability of our proposed system is far better than other techniques. We observed that the localization error is still more than 1.2 m for a small region. SCAN and DOUBLE SCAN have almost the same localization error with a difference of 0.02%, whereas in HILBERT, this percentage is increased to 3.5%. The only reason for this variation is a

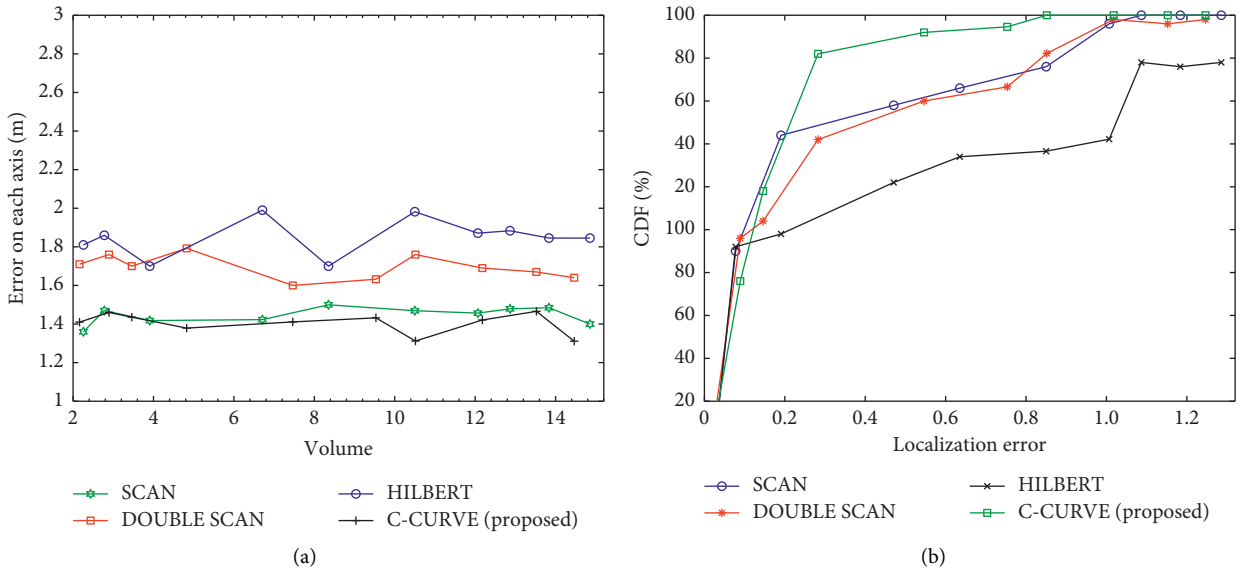


Figure 10: (a) Comparison of SCAN, DOUBLE SCAN, and HILBERT algorithm with our proposed solution (20 m with 20% anchor density). (b) CDF of localization error.

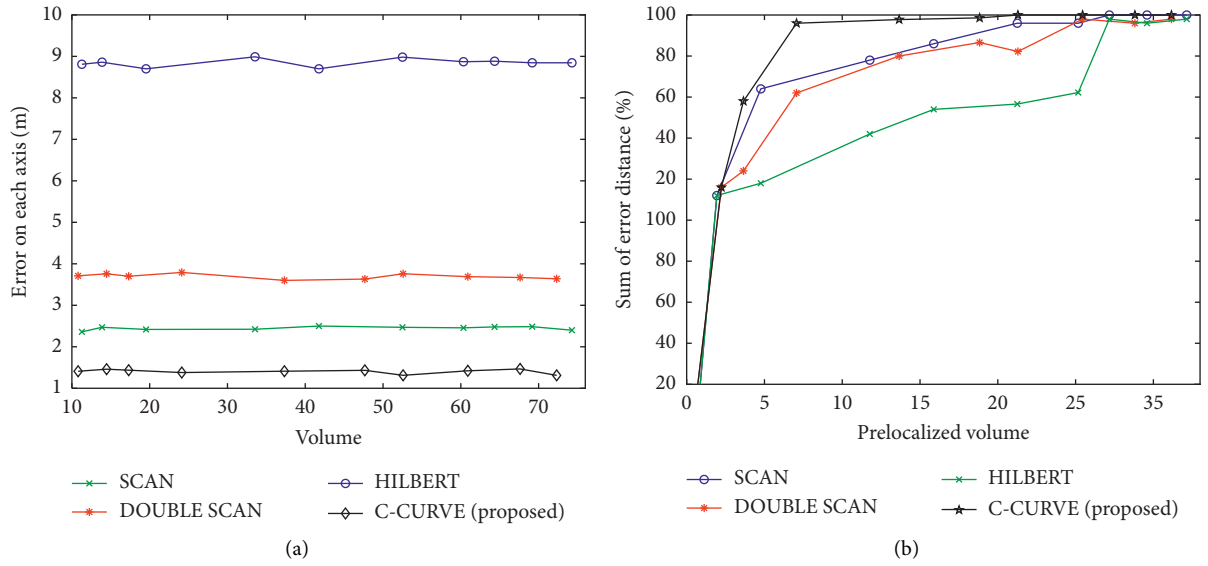


Figure 11: (a) Comparison of SCAN, DOUBLE SCAN, and HILBERT algorithm with our proposed solution (70 m with 50% anchor density). (b) CDF of localization error.

small variation in anchor node density and a very small region of 30 m. Furthermore, there are a lot of turns in the HILBERT algorithm that increase the complexity, whereas in our proposed system, there is a very slight movement of mobile anchor along its trajectory, which helps to save enough energy and hence computation cost. The comparison analysis is shown in Figure 10(a) along with the CDF for small distance as shown in Figure 10(b).

We observe that the CDF for DOUBLE SCAN was better than that for SCAN and HILBERT algorithm. However, after 5 m, it starts gaining worse localization error due to not having enough beacon on the trajectory part. It was observed that the localization error of SCAN, DOUBLE SCAN, and HILBERT becomes too worse in case of high resolution. Even just for a 60 m distance, the error becomes (2.4, 3.9, 8.7) m, respectively. The C-CURVE algorithm is not affected because of having enough network coverage path on the network. That is why our system gives a highly precise localization solution for mobile anchor-based localization system. By increasing such distance only up to ~70 m, the error goes high as shown in Figure 11 along with CDF for the particular distance.

The proposed algorithm has been compared with some other techniques based on mobile anchor flying over a random path. The mobile beacon assisted localization scheme presented in [35], used a movement strategy of

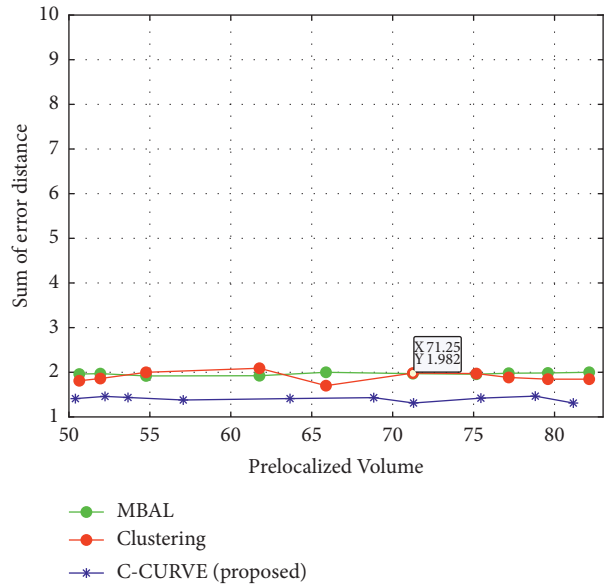


Figure 12: Comparison of MBAL, clustering technique, and proposed algorithm.

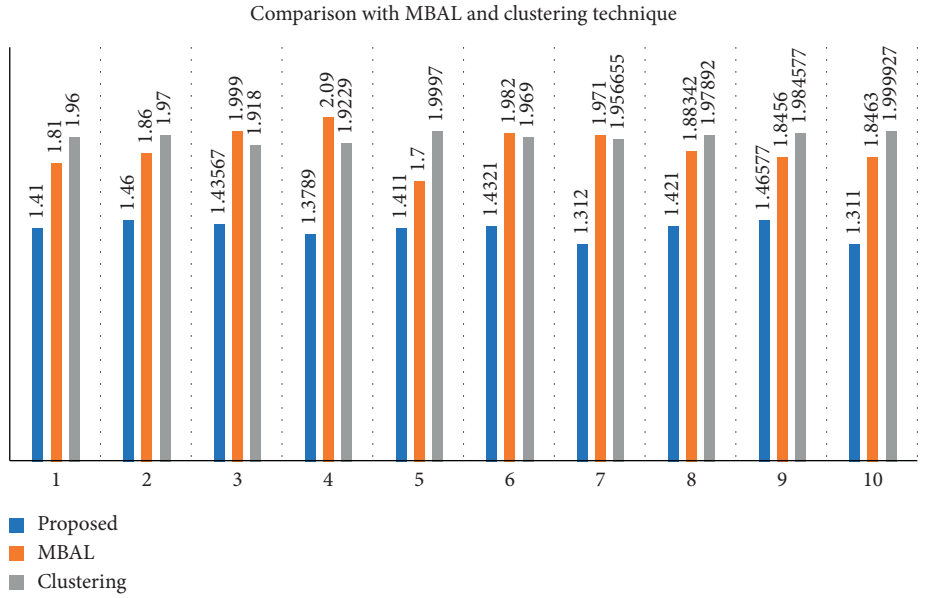


Figure 13: Comparison of MBAL, clustering technique, and proposed algorithm with 50% anchor nodes density.

mobile beacons, and used a fixed trajectory rather than a random walk over the network region. The number of messages broadcast by a beacon node and the path length is used as the main parameter in this scheme. So, if we increased the number of anchor nodes, the number of transmitted messages also increased. For 25 anchor nodes, a movement length of ~602 m has been covered. In our proposed algorithm, the path is a C-shaped random curve which minimizes the length of random walk for an anchor node which helps to minimize the computation cost and node battery lasts for a longer period. In case of increase in number of anchor nodes in MBAL scheme shown very good performance and decreases 93.3% of transmitted becaons messages and 95.3% of movement length (trajectory). However, the network coverage is a limitation of this scheme. With a 50% of anchor node density and a fixed distance of 70 m, we observed that, over the selected region, the error of MBAL is still very high as compared to our proposed algorithm as shown in Figure 12. The only reason behind this is that the MBAL only focused on a number of broadcast messages in the presence of fewer flying anchors. The proposed scheme has also been compared with the recent techniques proposed in [36], which form a clustering head as compared to triangles in a region. The bounded clusters are always equipped with anchor nodes over the deployed region. In case the region is very huge, the outer

anchor nodes failed to localize the inner most nodes. Therefore, our technique is far better than the clustering mechanism as shown in Figure 13.

4. Conclusions and Future Work

In this work, our focus was to localize the sensor nodes with the help of mobile anchor nodes. To do so, we have proposed a mobile anchor-based localization algorithm that follows the C-shaped trajectories in a 3D-based network and computed the localization error. The mobile anchor node keeps sending the beacons and obtains RSSI values against each node. The successive LQI value of a node can help to identify the average signal strength, which helps to accurately measure the distance. The distance matrix is then formed for triangulation computation, which accurately measures the node position. The simulation result shows that the C-CURVE algorithm shows much efficiency even in case of multipath fading. Our proposed algorithm also provides high accuracy even in presence of noise, due to the use of successive LQI values.

For future work, we are still working on localization error computation in presence of noise. In the second phase of the study, we are computing the complexity of the algorithm and measuring the noise factor in presence of Gaussian and intelligent noise. The noise factor is then eliminated by using extended Kalman filtering. We also compute the lower bound error to sum up this study.

Data Availability

The RSSI dataset is being taken from pspachos GitHub.

Conflicts of Interest

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

Acknowledgments

This work was fully supported by the National Nature Science Foundation of China under projects 266343 61673079, and 61703068 and the Natural Science Foundation of Chongqing under project cstc2018jcyjAX0160.

References

[1] T. Ahmad, X. Li, and B. C. Seet, "Parametric loop division for 3d localization in wireless sensor networks," *Sensors*, vol. 17, no. 7, p. 1697, 2017.

[2] F. Xu, H. Ye, F. Yang, and C. Zhao, "Software defined mission-critical wireless sensor network: architecture and edge offloading strategy," *IEEE Access*, vol. 7, pp. 10383–10391, 2019.

[3] R. E. Mohamed, A. I. Saleh, M. Abdelrazzak, and A. S. Samra, "Survey on wireless sensor network applications and energy efficient routing protocols," *Wireless Personal Communications*, vol. 101, no. 2, pp. 1019–1055, 2018.

[4] H. Wu, Z. Ding, and J. Cao, "GROLO: realistic range-based localization for mobile IoTs through global rigidity," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5048–5057, 2019.

[5] T. Ahmad, X. J. Li, and B. C. Seet, "A self-calibrated centroid localization algorithm for indoor ZigBee WSNs," in *Proceedings of the 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, pp. 455–461, Beijing, China, 2016 June.

[6] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. K. Karagiannidis, "A survey on mobile anchor node assisted localization in wireless sensor networks," *IEEE Communications Survey*, vol. 8, no. 13, pp. 2220–2243.

[7] A. Kumar and R. K. Paul, "Progressive localization using mobile anchor in wireless sensor network," *International Journal of Engineering and Computer Science*, vol. 6, 4 pages, 2019.

[8] T. Alhmiedat, A. Abu Taleb, and M. Bsoul, "A study on threads detection and tracking systems for military applications using WSNs," *International Journal of Computer Applications*, vol. 40, no. 15, pp. 12–18, 2012.

[9] M. A. Hussain and K. kyung Sup, "WSN research activities for military application," in *Proceedings of the 2009 11th International Conference on Advanced Communication Technology*, pp. 271–274, IEEE, Phoenix Park,South Korea, 2009, February.

[10] J. Fan, T. Liang, T. Wang, and J. Liu, "Identification and localization of the jammer in wireless sensor networks," *The Computer Journal*, vol. 62, no. 10, pp. 1515–1527, 2019.

[11] K. Chintalapudi, T. Fu, J. Paek et al., "Monitoring civil structures with a wireless sensor network," *IEEE Internet Computing*, vol. 10, no. 2, pp. 26–34, 2006.

[12] J. Masri, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson, "Analysis of wireless sensor networks for habitat monitoring," in *Proceedings of the Wireless sensor networks*, pp. 399–423, Springer, Boston, MA, USA, January 2004.

[13] K. Sha, W. Shi, and O. Watkins, "Using wireless sensor networks for fire rescue applications: requirements and challenges," in *Proceedings of the 2006 IEEE International Conference on Electro/Information Technology*, pp. 239–244, IEEE, East Lansing, MI, USA, 2006 May.

[14] Y. Li, Z. Wang, and Y. Song, "Wireless sensor network design for wildfire monitoring," in *Proceedings of the 2006 6th World Congress on Intelligent Control and Automation*, vol. 1, pp. 109–113, IEEE, Dalian, China, 2006 June.

[15] H. Yan, Y. Xu, and M. Gidlund, "Experimental e-health applications in wireless sensor networks," in *Proceedings of the 2009 WRI International Conference on Communications and Mobile Computing*, vol. 1, pp. 563–567, IEEE, Yunnan, China, 2009 January.

[16] S. Pirbhulal, H. Zhang, M. E Alahi et al., "A novel secure IoT-based smart home automation system using a wireless sensor network," *Sensors*, vol. 17, no. 1, p. 69, 2017.

[17] S. Elango and P. Gupta, "RSSI based indoor position monitoring using WSN in a home automation application," *Acta Electrotechnica et Informatica*, vol. 11, no. 4, p. 14, 2011.

[18] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Vehicular ad hoc networks: a new challenge for localization-based systems," *Computer Communications*, vol. 31, no. 12, pp. 2838–2849, 2008.

[19] T. Ahmad, X. J. Li, and B. C. Seet, "3D localization using social network analysis for wireless sensor networks," in *Proceedings of the 2018 IEEE 3rd International Conference on Communication and Information Systems (ICCIS)*, pp. 88–92, Singapore, Singapore, 2018, December.

[20] T. Ahmad, X. J. Li, and B.-C. Seet, "Noise reduction scheme for parametric loop division 3D wireless localization

algorithm based on extended kalman filtering," *Journal of Sensor and Actuator Networks*, vol. 8, no. 2, p. 24, 2019.

[21] Y. Wang, X. Ma, and G. Leus, "Robust time-based localization for asynchronous networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4397–4410, 2011.

[22] Y. S. Lee, J. W. Park, and L. Barolli, "A localization algorithm based on AOA for ad-hoc sensor networks," *Mobile Information Systems*, vol. 8, no. 1, pp. 61–72, 2012.

[23] P. Singh and S. Agrawal, "TDOA based node localization in WSN using neural networks," in *Proceedings of the 2013 International Conference on Communication Systems and Network Technologies*, pp. 400–404, IEEE, Gwalior, India, 2013 April.

[24] C. Alippi and G. Vanini, "A RSSI-based and calibrated centralized localization technique for wireless sensor networks," in *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, p. 5, Pisa, Italy, 2006 March.

[25] D. Koutsonikolas, S. M. Das, and Y. C. Hu, "Path planning of mobile landmarks for localization in wireless sensor networks," *Computer Communications*, vol. 30, no. 13, pp. 2577–2592, 2007.

[26] Z. Hu, D. Gu, Z. Song, and H. Li, "Localization in wireless sensor networks using a mobile anchor node," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '08)*, pp. 602–607, Xian, China, July 2008.

[27] H. Cui, Y. Wang, and J. Lv, "Path planning of mobile anchor in three-dimensional wireless sensor networks for localization," *Journal of Information and Computational Science*, vol. 9, no. 8, pp. 2203–2210, 2012.

[28] H. Li, J. Wang, X. Li, and H. Ma, "Real-time path planning of mobile anchor node in localization for wireless sensor networks," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA '08)*, pp. 384–389, Changsha, China, June 2008.

[29] H. Haddadi, M. Rio, G. Iannaccone, A. Moore, and R. Mortier, "Network topologies: inference, modeling, and generation," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 2, pp. 48–69, 2008.

[30] S. Sadowski and P. Spachos, "Rssi-based indoor localization with the internet of things," *IEEE Access*, vol. 6, pp. 30149–30161, 2018.

[31] ZigBee Standards Organisation, *ZigBee Specification*, ZigBee Alliance, San Ramon, CA, USA, 2008.

[32] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 81–95, ACM, Los Angeles, CA, USA, September 2003.

[33] T. Ahmad and J. Xue, B.-C. Seet, "3D localization based on parametric loop division and subdivision surfaces for wireless sensor networks," in *Proceedings of the 2016 25th Wireless and Optical Communication Conference (WOCC)*, pp. 1–6, IEEE, New york, NJ, USA, May 2016.

[34] T. Ahmad and J. Xue, W. Jiang and A. Ghaffar, "Frugal sensing: a novel approach of mobile sensor network localization based on fuzzy-logic," in *Proceedings of the ACM MobiArch 2020 The 15th Workshop on Mobility in the Evolving Internet Architecture*, pp. 8–15, Los Angeles, CA, USA, September 2020.

[35] K. Kim and W. Lee, "MBAL: a mobile beacon-assisted localization scheme for wireless sensor networks," in *Proceedings of the 2007 16th International Conference on Computer Communications and Networks*, pp. 57–62, IEEE, Honolulu, HI, USA, August 2007.

[36] S. Sahana and K. Singh, "Cluster based localization scheme with backup node in underwater wireless sensor network," *Wireless Personal Communications*, vol. 110, no. 4, pp. 1693–1706, 2020.