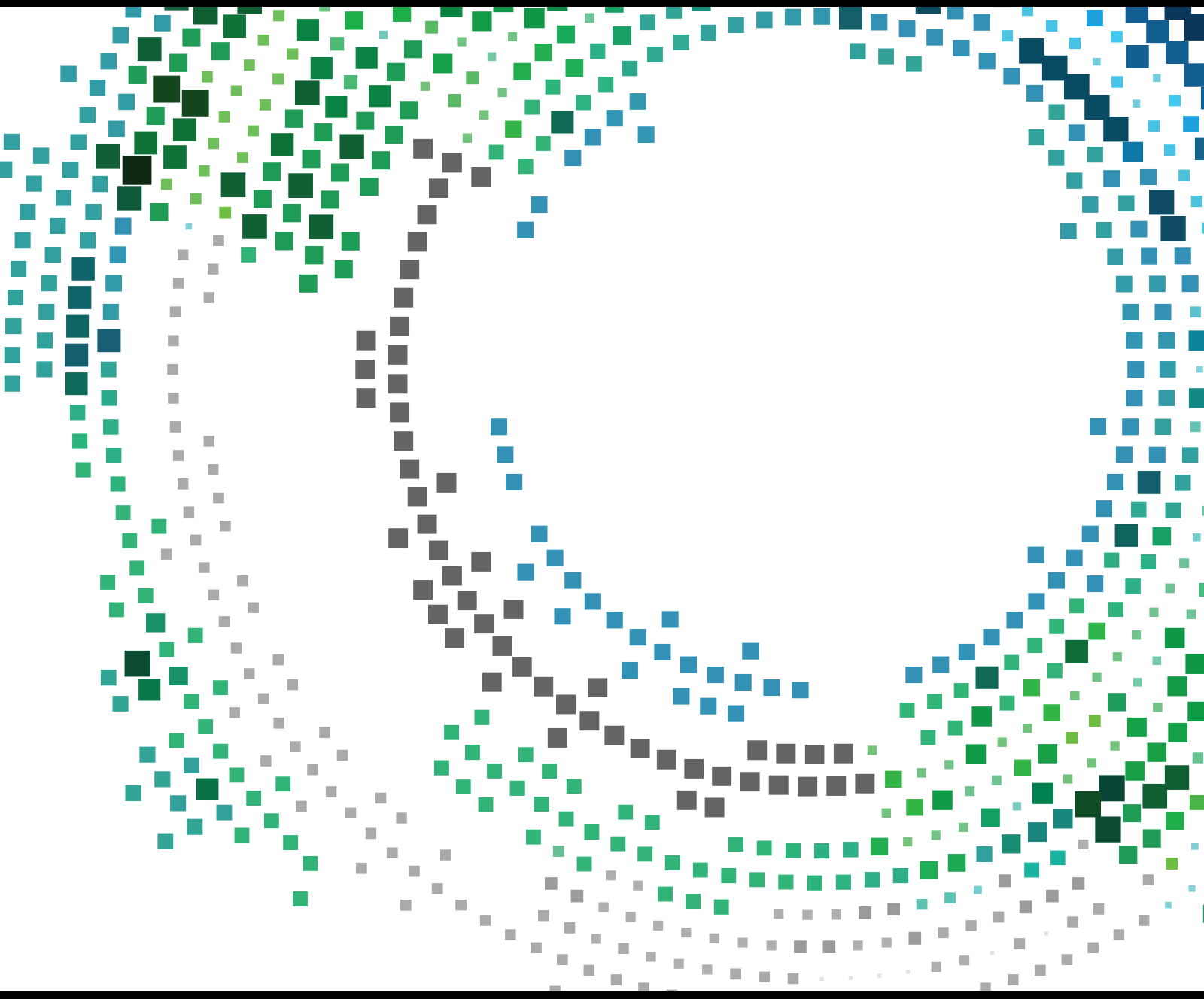
Mobile Information Systems

# Distributed Secure Computing for Smart Mobile IoT Networks

Lead Guest Editor: Vishal Sharma
Guest Editors: Daniel G. Reina, Zengpeng Li, Kathiravan Srinivasan, Navuday Sharma, and Vinod Karar



# Distributed Secure Computing for Smart Mobile IoT Networks

Mobile Information Systems

# Distributed Secure Computing for Smart Mobile IoT Networks

Lead Guest Editor: Vishal Sharma
Guest Editors: Daniel G. Reina, Zengpeng Li, Kathiravan Srinivasan, Navuday Sharma, and Vinod Karar

# Contents

**Distributed Secure Computing for Smart Mobile IoT Networks**
Vishal Sharma, Daniel G. Reina, Zengpeng Li, Kathiravan Srinivasan, Navuday Sharma, and Vinod Karar
Editorial (2 pages), Article ID 9864846, Volume 2022 (2022)

**Current Research Trends in IoT Security: A Systematic Mapping Study**
Jee Young Lee and Jungwoo Lee
Review Article (25 pages), Article ID 8847099, Volume 2021 (2021)

**A2 Chain: A Blockchain-Based Decentralized Authentication Scheme for 5G-Enabled IoT**
Xudong Jia, Ning Hu, Shi Yin, Yan Zhao, Chi Zhang, and Xinda Cheng
Research Article (19 pages), Article ID 8889192, Volume 2020 (2020)

**HAL-Based Resource Manipulation Monitoring on AOSP**
Thien-Phuc Doan, Jungsoo Park, and Souhwan Jung
Research Article (9 pages), Article ID 8863385, Volume 2020 (2020)

**Investment Priority Analysis of ICS Information Security Resources in Smart Mobile IoT Network Environment Using the Analytic Hierarchy Process**
Jiho Shin, Ilsun You, and Jung Taek Seo
Research Article (11 pages), Article ID 8878088, Volume 2020 (2020)

**Facilitating User Authorization from Imbalanced Data Logs of Credit Cards Using Artificial Intelligence**
Vinay Arora, Rohan Singh Leekha, Kyungroul Lee, and Aman Kataria
Research Article (13 pages), Article ID 8885269, Volume 2020 (2020)

**Rogue Device Mitigation in the Internet of Things: A Blockchain-Based Access Control Approach**
Uzair Javaid, Furqan Jameel, Umair Javaid, Muhammad Toaha Raza Khan, and Riku Jäntti
Research Article (13 pages), Article ID 8831976, Volume 2020 (2020)

**Achieving Message-Encapsulated Leveled FHE for IoT Privacy Protection**
Weiping Ouyang, Chunguang Ma, Guoyin Zhang, and Keming Diao
Research Article (10 pages), Article ID 8862920, Volume 2020 (2020)

**Self-Controllable Mobile App Protection Scheme Based on Binary Code Splitting**
Sungtae Kim, Taeyong Park, Geochang Jeon, and Jeong Hyun Yi
Research Article (11 pages), Article ID 8813243, Volume 2020 (2020)

**An Intrusion Detection Scheme Based on Repeated Game in Smart Home**
Rui Zhang, Hui Xia, Shu-shu Shao, Hang Ren, Shuai Xu, and Xiang-guo Cheng
Research Article (9 pages), Article ID 8844116, Volume 2020 (2020)

**Multiaccess Edge Computing Empowered Flying Ad Hoc Networks with Secure Deployment Using Identity-Based Generalized Signcryption**

Muhammad Asghar Khan, Insaf Ullah, Shibli Nisar, Fazal Noor, Ijaz Mansoor Qureshi, Fahimullah Khanzada, Hizbullah Khattak, and Muhammad Adnan Aziz

Research Article (15 pages), Article ID 8861947, Volume 2020 (2020)

Hindawi
Mobile Information Systems
Volume 2022, Article ID 9864846, 2 pages
https://doi.org/10.1155/2022/9864846

*Editorial*

# Distributed Secure Computing for Smart Mobile IoT Networks

**Vishal Sharma**,[1] **Daniel G. Reina**,[2] **Zengpeng Li**,[3] **Kathiravan Srinivasan**,[4] **Navuday Sharma**,[5] **and Vinod Karar**[6]

[1]*Queen's University Belfast, Belfast, NI BT7 1NN, UK*
[2]*University of Seville, Seville 41004, Spain*
[3]*School of Cyber Science and Technology, Shandong University, Qingdao 266237, China*
[4]*Vellore Institute of Technology, Vellore 632014, India*
[5]*Ericsson, Tallinn 11415, Estonia*
[6]*Central Scientific Instruments Organisation, Chandigarh 160030, India*

Correspondence should be addressed to Vishal Sharma; vishal_sharma2012@hotmail.com

Received 22 February 2022; Accepted 22 February 2022; Published 15 April 2022



With billions of devices operations as a part of the Internet of Things (IoT), the operational complexity of the networks increases to many folds. In terms of threat detection, it requires moving from a centralised detection model to decentralised and distributed formations. Distributed computing facilitates better services; however, it has multiple ownership issues, requiring better system management [1]. With distributed systems, security needs to be revisited to make them aloof from cyber threats with new solutions, like sound and data steganography for authentication [2], preventing stealthy adversaries [3], access control [4], or network anomaly detection [5]. New and advanced solutions are required to solve the computationally intensive problems with better offloading in distributed setup targeting smart mobile IoT. Another prominent issue in distributed IoT networks is the data-islands dilemma [6], which requires intelligent and secure mechanisms to handle data integrity and privacy. Different solutions can be adapted like the use of distributed ledger technologies, such as blockchain [7], to take authorisation and access control of many portable devices without letting the system fall short of decentralised attacks; Iota Tangle [8] can be used mainly for securing IoT environment, or gossip protocol-based Hashgraph [9] can be used for increased fairness and better security constraints without using block-based architecture. Understanding smart devices privacy, trust, and security with better authentication protocols is another side to explore [10]. Several key issues need to be addressed by covering the gap in the literature, which must help answer concerns related to achieving password-based authentication, keeping data privacy, outsourcing security, and intelligent security solutions using machine learning.

In this special issue (SI), a total of ten articles were selected following a rigorous review process where the articles were handled without any competing conflict of interest. The articles in this SI cover a wide range of security and privacy issues in distributed computing related to IoT, blockchain, resource manipulation, industrial control systems, credit cards, smart homes, and aerial networks. Some of the highlights include the following: In [11], the authors proposed an $A^2$ chain that uses an edge computing setup to decentralise the services. This article relies on the usage of sidechain technologies to securely share the identity verification of IoT devices. The authors used the proposed blockchain setup to authenticate the 5G-enabled IoT devices. Overall, this approach reduces the authentication time and communication cost whereby consuming less storage space. In [12], the authors focused their work on user authorisation, where the primary task was to detect credit card frauds from imbalanced data logs. The authors relied on the machine learning models and suggested that RUSBoost be a more appropriate model when imbalanced records need to be evaluated for fraud detection. The authors used datasets to show the efficacy of their proposed solution. Their results showed a possibility of high precision between 94.20 and 99.30 for three different credit card datasets.

In the direction of distributed security, rogue devices can be much harmful in any setup. These devices can be silent attackers that use the system's weak defence to launch attacks. The authors considered this area of research in [13], where they proposed a blockchain-based access control for mitigating rogue devices in IoT. The authors aimed at removing the centralised mode of detection by replacing architecture with the blockchain, which offers secure device registration using smart contracts. The access control mechanism prohibits unregistered devices, and the approach is evaluated using a case study and in-depth performance evaluations. Furthermore, in [14], the authors focused on secure deployment in flying ad hoc networks using identity-based generalised signcryption. Their proposed work used Mobile Edge Computing (MEC), where UAVs act as a MEC node with the role of offloading in the network. The proposed security scheme is based on a hyperelliptic curve. The authors formally verified their proposed security scheme using the AVISPA tool and compared it with five relevant security schemes against security functionalities.

Through its collection of diverse articles on distributed security, we believe that this special issue will benefit the research community.

## Conflicts of Interest

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

*Vishal Sharma*
*Daniel G. Reina*
*Zengpeng Li*
*Kathiravan Srinivasan*
*Navuday Sharma*
*Vinod Karar*

## Acknowledgments

The guest editors appreciate the high-quality submissions from the authors and the timely support of reviewers.

## References

[1] S. Jiang, T. Jiang, and L. Wang, "Secure and efficient cloud data deduplication with ownership management," *IEEE Transactions on Services Computing*, vol. 13, no. 6, p. 1, 2017.

[2] D. Datta, L. Garg, K. Srinivasan et al., "An efficient sound and data steganography based secure authentication system," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 723–751, 2021.

[3] G. Park, C. Lee, H. Shim, Y. Eun, and K. H. Johansson, "Stealthy adversaries against uncertain cyber-physical systems: threat of robust zero-dynamics attack," *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 4907–4919, 2019.

[4] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, "Access control for emerging distributed systems," *Computer*, vol. 51, no. 10, pp. 100–103, 2018.

[5] D. Patel, K. Srinivasan, C.-Y. Chang, T. Gupta, and A. Kataria, "Network anomaly detection inside consumer networks-A hybrid approach," *Electronics*, vol. 9, no. 6, p. 923, 2020.

[6] Z. Li, V. Sharma, and S. P. Mohant, "Preserving data privacy via federated learning: challenges and solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 3, pp. 8–16, 2020.

[7] Y. Xu, G. Wang, J. Yang, J. Ren, Y. Zhang, and C. Zhang, "Towards secure network computing services for lightweight clients using blockchain," *Wireless Communications and Mobile Computing*, vol. 201812 pages, 2018, https://doi.org/10.1155/2018/2051693, Article ID 2051693.

[8] W. F. Silvano and R. Marcelino, "Iota Tangle: a cryptocurrency to communicate Internet-of-Things data," *Future Generation Computer Systems*, vol. 112, pp. 307–319, 2020.

[9] A. A. Zahoor, M. M. Khan, and J. Arshad, "A comparative study of distributed ledger technologies," in *Blockchain for Cybersecurity and Privacy*, pp. 29–55, CRC Press, Boca Raton, FL, USA, 2020.

[10] V. Sharma, I. You, K. Andersson, F. Palmieri, M. H. Rehmani, and J. Lim, "Security, privacy and trust for smart mobile-internet of things (M-IoT): a survey," *IEEE Access*, vol. 8, pp. 167123–167163, 2020.

[11] X. Jia, N. Hu, S. Yin, Y. Zhao, C. Zhang, and X. Cheng, "$A^2$ chain: a blockchain-based decentralized authentication scheme for 5G-enabled IoT," *Mobile Information Systems*, vol. 2020, Article ID 8889192, 19 pages, 2020.

[12] V. Arora, R. S. Leekha, K. Lee, and A. Kataria, "Facilitating user authorization from imbalanced data logs of credit cards using artificial intelligence," *Mobile Information Systems*, vol. 2020, Article ID 8885269, 9 pages, 2020.

[13] U. Javaid, F. Jameel, U. Javaid, M. T. Raza Khan, and R. Jäntti, "Rogue device mitigation in the internet of things: a blockchain-based access control approach," *Mobile Information Systems*, vol. 2020, Article ID 8831976, 8 pages, 2020.

[14] 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. 202011 pages, 2020, https://doi.org/10.1155/2020/8861947, Article ID 8861947.

Hindawi
Mobile Information Systems
Volume 2021, Article ID 8847099, 25 pages
https://doi.org/10.1155/2021/8847099

*Review Article*

# Current Research Trends in IoT Security: A Systematic Mapping Study

**Jee Young Lee and Jungwoo Lee**

*Graduate School of Information, Yonsei University, Seoul 037222, Republic of Korea*

Correspondence should be addressed to Jee Young Lee; j.ann.lee@yonsei.ac.kr

Received 26 September 2020; Revised 29 January 2021; Accepted 27 February 2021; Published 13 March 2021

Academic Editor: Vishal Sharma



The smart mobile Internet-of-things (IoT) network lays the foundation of the fourth industrial revolution, the era of hyper-connectivity, hyperintelligence, and hyperconvergence. As this revolution gains momentum, the security of smart mobile IoT networks becomes an essential research topic. This study aimed to provide comprehensive insights on IoT security. To this end, we conducted a systematic mapping study of the literature to identify evolving trends in IoT security and determine research subjects. We reviewed the literature from January 2009 to August 2020 to identify influential researchers and trends of keywords. We additionally performed structural topic modeling to identify current research topics and the most promising ones via topic trend estimation. We synthesized and interpreted the results of the systematic mapping study to devise future research directions. The results obtained from this study are useful to understand current trends in IoT security and provide insights into research and development of IoT security.

## 1. Introduction

The era of hyper-connectivity, hyper-intelligence, and hyper-convergence established by the fourth industrial revolution is continuing in earnest as smart mobile Internet-of-things (M-IoT) environments are developing. The Internet of things (IoT) establishes a new networking paradigm in which various devices (e.g., network devices, sensors, and actuators) become essential elements for communication. Various objects can be considered as "smart" because they are equipped with microprocessors and network transceivers, enabling communication and the provision of autonomous services. IoT is a promising field of research related to building device networks connected to the Internet and promotes smart environments. IoT is associated with many research areas and new computing paradigms. The M-IoT cloud-computing domain, which lies at the intersection of the cloud, mobile, and IoT domains, provides new paradigms of fog computing, edge computing, mobile-edge computing (MEC), the semantic web of things, and mobile crowdsensing. Elazhary [1] summarized various related concepts. The Internet of mobile things (i.e., M-IoT) is a special case of IoT concerned with mobile IoT devices. Such devices include smartphones, vehicles, and wearable devices [2]. The IoT paradigm is also evolving into smart M-IoT devices, which in turn provide smart services and computing functions.

IoT-based smart systems and services are being developed in various fields, such as home automation, energy management, healthcare, and financial transaction management [3–6]. It is also branching into new domains, such as social IoT, in which smart objects are transformed into social objects; industrial IoT, which converges with different industries; smart-wearable IoT, which combines deep learning and wearable technologies; and medical IoT, which is integrated with medical applications [3–6].

Smart M-IoT provides smart convergence services to users of IoT environments. Accordingly, many researchers in various fields are now involved with IoT development. For the continued spread and development of smart M-IoT, it is necessary to consider security, as the devices and platforms of smart M-IoT mainly remain threatened [7]. The emphasis

on security will increase, and both consolidated and new researchers need understanding and insights on IoT security.

The remainder of this paper is organized as follows. Section 2 discusses related work about the study on IoT topics and trends. Section 3 describes the conducted systematic mapping study on IoT security. Section 4 discusses the main findings. Influential authors are identified in Section 4.1, and keyword-based clusters and keyword trends are presented in Section 4.2. Research topics related to IoT security are categorized in Section 4.3, and the trend of topics is discussed in Section 4.4. Section 4.5 provides future perspectives by synthesizing the keyword and topic trends. Finally, conclusions are drawn in Section 5.

## 2. Related Work

*2.1. Research Methodology.* One of the first challenges before conducting research in any field of study is identifying relevant previous studies and establishing the need for new research [8]. Secondary research analyzes existing studies (primary research) and seeks to provide relevant insights to researchers and guide the design of future research. Secondary research methodologies include the review, systematic literature review (SLR), and systematic mapping study.

In the review or survey, researchers select important literature according to their expertise. Then, they synthesize and organize the contents. The review provides new understanding and insights about the content through in-depth content comparison analyses. However, as the content should be analyzed closely, there is a limit to the number of documents that can be included in the study due to time and cost constraints [8, 9].

The SLR applies an explicit and systematic protocol for collecting, selecting, and analyzing research literature [10]. It provides quantitative and statistical insights on the subject by analyzing primary studies to answer research questions while providing aggregate result data [11]. Therefore, SLRs can be performed with studies that can quantitatively extract information meeting the aggregation criteria.

The relatively recently developed systematic mapping study is a more open form of SLR, which aims to organize a research area [9]. This method uses the same protocol as the SLR to find and select research literature. Unlike the SLR, the systematic mapping study classifies subfields of a research area [11, 12] and focuses on identifying and classifying themes by collecting as many studies as possible [13]. The categories used are generally based on publication information (e.g., author name, author affiliation, publication source, publication type, and publication date) and/or information about the adopted research method [13]. A systematic mapping study is sometimes conducted as a preliminary study before the SLR [14, 15]. It classifies subject areas and identifies those requiring detailed content comparisons. Research on text mining and visualization tools that can be used to efficiently perform this type of analysis is ongoing [14, 16, 17]. Petersen et al. [9, 15] noted that performing a systematic mapping study before an SLR provided valuable research design criteria. Kitchenham et al. [13, 18] stated that systematic mapping can provide input data for subsequent studies. In other words, systematic mapping reduces the preparation time for subsequent research. In addition, it provides an overview of research areas and identifies research gaps. Moreover, it helps in identifying research trends and educational materials.

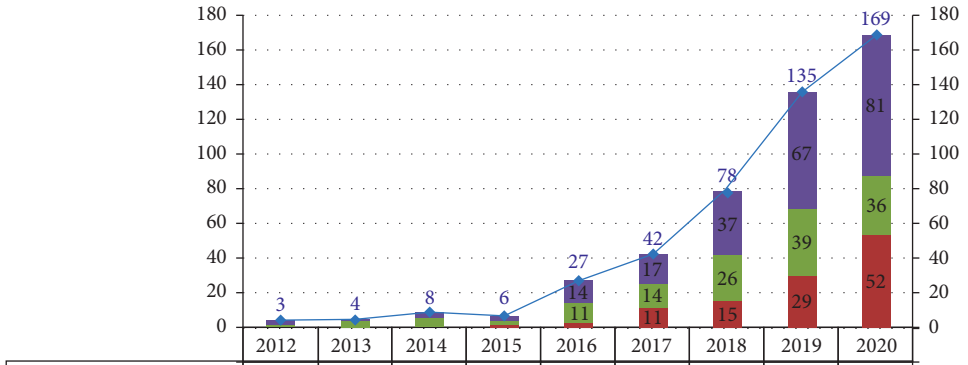
*2.2. Comparison with Related Reviews.* To better understand existing secondary research related to IoT, Scopus articles classified as "review" between January 2012 and October 2020 were collected, obtaining 472 review articles. These articles were then further categorized into labels "IoT security review," "IoT application review," or "IoT review," as shown in Figure 1.

Reviews related to IoT have been increasing rapidly since 2018. IoT applications including smart cities [19, 20], smart health [21, 22], smart agriculture [23, 24], and smart vehicles [25, 26] were the most frequently reviewed. In 2020, IoT security reviews were more numerous than IoT reviews. Note that we did not classify articles that have partially discussed security under label "IoT security review." Instead, we classified the articles that exclusively focus on security under this label. Table 1 compares recent reviews on IoT security from 2017 to 2020 in terms of methodology. Most of these reviews synthesized and organized contents using a review/survey method. From them, articles similar to our study are listed in Table 2.

Existing studies have some limitations. Alaba et al. [27] focused on the classification of security threats but did not cover the overall contents and did not discuss new technologies, such as machine learning (ML). Mendez Mena et al. [28] focused on IoT architectures but did not consider applications. Obaidat et al. [32] aimed to comprehensively cover IoT security but omitted related applications. In contrast, Hassija et al. [29] did not cover IoT as a whole, focusing only on applications. Hameed et al. [31] did not deal with trust as a security requirement. The major limitation of the abovementioned reviews is that they fail to provide research trends.

Sharma et al. [7] dealt with the most recent paradigm in depth, focusing on smart M-IoT, and provided a roadmap for related surveys. However, it was not a study focused on providing early insights to researchers entering from other fields. Macedo et al. [30] focused on providing insights and research trends using an SLR, but they omitted privacy. In addition, they only selected 131 articles for review. Most of the review studies not listed in Table 2 focused on specific areas of IoT security, such as layer protocols [33], intrusion detection [34], device security [35, 36], trust [37], and security of specific IoT applications [38]. Thus, a systematic mapping study is still required to determine research topics and trends in IoT security and gain insights on this field.

*2.3. Contributions of This Study.* For the transition to a secure, smart M-IoT, we should understand the available resources on IoT security. We aimed to provide researchers interested in IoT research with early insights on IoT security by conducting a systematic mapping study. To the best of our




| | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 |
|---|---|---|---|---|---|---|---|---|---|
| IoT application review | 2 | 1 | 3 | 2 | 14 | 17 | 37 | 67 | 81 |
| IoT review | 1 | 3 | 5 | 3 | 11 | 14 | 26 | 39 | 36 |
| IoT security review | 0 | 0 | 0 | 1 | 2 | 11 | 15 | 29 | 52 |
| Total | 3 | 4 | 8 | 6 | 27 | 42 | 78 | 135 | 169 |

Figure 1: Trends in IoT-related review articles.

Table 1: Comparison of methodology used in IoT security review articles from 2017 to 2020.

| Methodology | 2017 | 2018 | 2019 | 2020 |
|---|---|---|---|---|
| Review/Survey | 11 | 15 | 25 | 47 |
| SLR | 0 | 0 | 4 | 5 |
| Systematic mapping study | 0 | 0 | 0 | 0 |

knowledge, no such studies focused on IoT security are available. We applied big data mining tools to large volumes of literature for the systematic mapping study, which is thus unbiased and replicable. We classify research on IoT security based on keywords and topics. We also explain trends and provide new understanding about keyword evolution and promising research topics. The results from this study may be used by lecturers to teach the overview, main topics, and trends related to IoT security. In addition, a qualitative content analysis provides future research directions.

In this study, we also demonstrated the application of big data mining to a systematic mapping study. The methods and findings reported in this paper may provide research opportunities by improving the overall understanding of IoT security and its research trends. In addition, the results of this study can be useful to researchers in other fields who intend to investigate IoT convergence.

## 3. Methods

In this study, we conducted a systematic mapping study of current research related to IoT security by mixing quantitative and qualitative approaches. The quantitative approach involves collecting literature on IoT security and conducting a systematic mapping study to identify influential researchers and concurrent keywords. We then classify the topics using an ML-based structural topic model (STM). Next, we perform qualitative content analysis to devise future research directions by synthesizing and discussing the latest keyword and topic trends. Our research aims to answer the following research questions:

RQ1. Who are influential researchers in IoT security?

RQ2. What are the major keywords in IoT security?

RQ2-1. What is the keyword-based research area?

RQ2-2. How are keywords evolving?

RQ3. What are the topics in IoT security field?

RQ3-1. What is topic-based research classification?

RQ3-2. What is the trend of topics?

RQ4. What are the most influential keywords in IoT security?

RQ5. What are promising research topics in IoT security?

Figure 2 shows the research framework that we used to understand the current status and trends in IoT security.

We selected studies according to PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) [8]. We adopted a review protocol consisting of search terms, resources to be searched, study selection criteria, and study selection procedures, as listed in Table 3. We used Boolean operator AND to combine IoT and security-related terms (e.g., "secure," "security," "privacy," and "trust"). We filtered the data based on the document type (e.g., "article"), source (e.g., "journal"), and language (e.g., "English"). The main research question and review protocols are listed in Table 3. Our literature search was conducted using 1,365 studies published from January 2009 to August 2020. Unlike existing review studies, we analyzed a large volume of articles to obtain comprehensive insights. To process that large volume, we used big data mining tools.

*3.1. Bibliometric Mapping Study on IoT Security.* In recent years, bibliometric analyses, co-citation network analyses, and keyword co-occurrence network analyses have been widely

Table 2: Comparison with related review articles.

| Article | Adopted methodology | Main focus | Contribution/impact |
|---|---|---|---|
| Alaba et al. [27] | Review | IoT security threats and vulnerabilities | (i) Classification of security threats in the context of applications, architecture, communication, and data<br>(ii) Attack analysis for security scenarios |
| Mendez Mena et al. [28] | Review | Security from the perspective of IoT architecture | (i) IoT architecture technology and protocol review by layer<br>(ii) Review of privacy issues<br>(iii) Summarize ongoing security issues of IoT |
| Hassija et al. [29] | Review | Security of IoT application | (i) IoT application security related issues and threat sources review<br>(ii) Discussion of technology to increase trust in IoT applications<br>(iii) Discussion of the latest technology to increase the level of security |
| Macedo et al. [30] | SLR | IoT security overall | (i) Review of literature over the last 8 years to identify security issues and trends in terms of authentication, access control, data protection, and trust |
| Hameed et al. [31] | Review | Requirements of IoT security | (i) Review privacy, lightweight encryption framework, security routing, internal attack detection, and resilience management as security requirements<br>(ii) Explain the latest technology for resilience management and detection of internal attacks |
| Obaidat et al. [32] | Review | IoT security overall | (i) Comprehensive investigation of security, privacy, security frameworks, technologies, threats, vulnerabilities, and countermeasures.<br>(ii) Classification of the impact of attacks according to -NIST's FIPS 199 definitions |
| Sharma et al. [7] | Review | Security, privacy, and trust in smart M-IoT | (i) The first survey discussing the security of smart M-IoT<br>(ii) Describe the security framework of smart M-IoT and conduct an in-depth investigation in terms of security, privacy, and trust to provide research tasks, unresolved issues, and research directions |
| Our study | Systematic mapping study | IoT security overall | (i) Classify large-volume literature related to IoT security from 2009 to the present<br>(ii) Discussion of research trends through co-occurrence keyword mapping<br>(iii) Discussion of research trends through topic mapping<br>(iv) Provide future research direction |

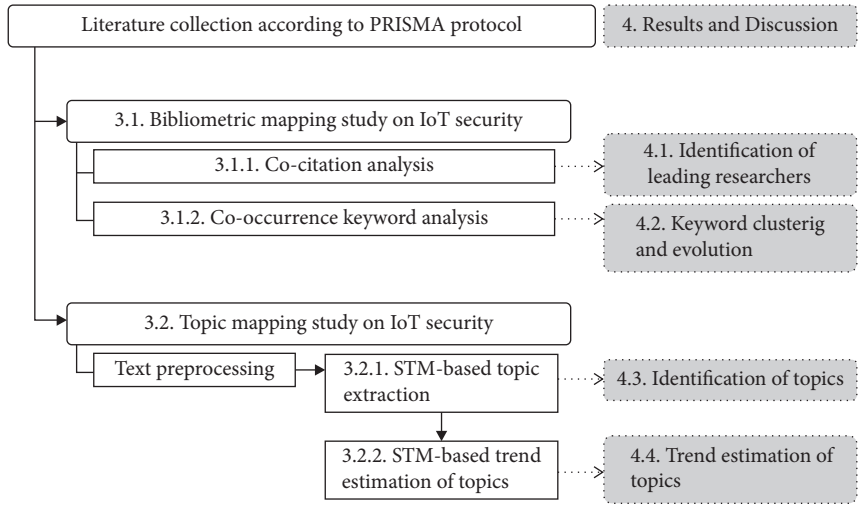



Figure 2: Research framework adopted in this study.

used to determine research trends [39–41]. Co-citation network analysis determines the structure of scientific communications by analyzing the associations among citations. Co-occurrence keyword network analysis allows to understand the knowledge structure underlying a technical field by analyzing links between keywords found in the literature.

Table 3: Research question and review protocol.

| Research goal | | What are the research trends in IoT security? |
|---|---|---|
| Review protocol | Search terms | ("IoT" OR "Internet of things") AND ("secure" OR "security" OR "privacy" OR "trust") in title |
| | Resources | Scopus |
| | Study selection criteria | Journal articles written in English |
| | Study selection procedures | Two researchers searched the databases and checked each other's work. |
| Study filtering | No. of studies satisfying criteria | 1,528 |
| | Duplication | −2 |
| | Unavailable abstract | −13 |
| | Unavailable author keywords | −148 |
| | No. of studies after filtering | 1,365 |

Radhakrishnan et al. [41] demonstrated the role of keyword co-occurrence networks in systematic reviews. In this current study, we conducted co-citation and co-occurrence keyword mapping studies to provide answers to RQ1 and RQ2.

*3.1.1. Co-Citation Network Analysis to Identify Authors of IoT Security Research.* By analyzing the co-citations of studies on IoT security, we can identify influential researchers and understand the research flow [42–44], and then we can answer RQ1. We performed author clustering by the relevance obtained from direct citation relationships. We used the quality function proposed by Traag et al. [45] and modified by Waltman and Van Eck [42] for clustering. The quality function is given by

$$Q(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \delta(x_i, x_j)\left(a_{ij} - \frac{\gamma}{2n}\right), \quad (1)$$

where $n$ is the number of studies, $a_{ij}$ measures the relation between studies $i$ and $j$, $\gamma$ is a resolution parameter, and $x_i$ denotes the cluster to which study $i$ is assigned. Function $\delta(x_i, x_j)$ is 1 if $x_i = x_j$ and 0 otherwise. The relation between studies $i$ and $j$ is measured as follows:

$$a_{ij} = \frac{c_{ij}}{\sum_{k=1}^{n} c_{ik}}. \quad (2)$$

In equation (2), if study $i$ cites study $j$ or vice versa, $c_{ij}$ is 1, whereas it is 0 otherwise. Hence, if there is no direct citation relation between studies $i$ and $j$, the relation measure, $c_{ij}$, is zero.

We used the CitNetExplorer tool for citation analysis [46] and set resolution parameter $\gamma$ to 1 and the number of parameter optimization iterations to 10.

*3.1.2. Co-Occurrence Keyword Network Analysis to Map Keyword Evolution on IoT Security.* Keyword co-occurrence analysis is commonly used to determine research trends, and it has been used to conduct a systematic literature review in [41]. We adopted the method proposed by Van Eck and Waltman [47] to construct and analyze a co-occurrence keyword network that answers RQ2 and RQ4.

We performed co-occurrence analysis on keywords collected from different studies. A keyword may appear in various forms (e.g., "blockchain," "blockchain," "blockchain," or "blockchains"). Therefore, after arranging a thesaurus, we applied it and grouped the keywords with the same meaning to then create a keyword co-occurrence matrix. Next, we generated a similarity matrix normalized according to the association strength of the keyword co-occurrence matrix [48]. Similarity $s_{ij}$ between items $i$ and $j$ according to the association strength is given by

$$s_{ij} = \frac{c_{ij}}{c_i c_j}, \quad (3)$$

where $c_{ij}$ represents the number of co-occurrences of items $i$ and $j$, and $c_i$ and $c_j$ represent the total number of occurrences of items $i$ and $j$, respectively.

Next, we visualized the similarities based on the similarity matrix by constructing a 2D map [49], where item 1, ..., $n$ is allocated such that the distance between any pair of items $i$ and $j$ reflects similarity $s_{ij}$ as accurately as possible. Items with high similarity were grouped closely, and those with low similarity remained distant. Specifically, we minimized the weighted sum of the squared Euclidean distances between all pairs. The higher the similarity between the two items, the higher the weight of the squared distance in the sum. The objective function for minimization is given by

$$V(x_1, \ldots, x_n) = \sum_{i<j} s_{ij} \left\| x_i - x_j^2 \right\|, \quad (4)$$

where vector $x_i = (x_{i1}, x_{i2})$ represents the position of item $i$ in the 2D map and $\| \cdot \|$ represents the Euclidean norm.

From bibliometric mapping, we obtained the nodes corresponding to the keywords in the co-occurrence network, link weight, total link strength, and occurrence weights. The link weight corresponds to the number of links per node, and the total link strength is the number of links from other nodes connected to a target node. In addition, the occurrence weight represents the frequency of keyword occurrence. We then performed clustering based on the mapping results according to the method proposed by Waltman et al. [49]. To improve clustering accuracy, we applied the smart local-moving algorithm developed by Waltman and Van Eck [50].

Finally, we used the VOSviewer tool to create and visualize the bibliometric map for keyword co-occurrence

network analysis [47]. We set the minimum number of occurrences of a keyword to 5 as a parameter in VOSviewer and set resolution $\gamma$ to 1 with a minimum cluster size of 5. We consulted two IoT experts to analyze the clusters regarding the similarities of the co-occurrence keyword network.

*3.2. Topic Mapping Study to Identify Topics in IoT Security.* Regarding RQ3 and RQ5, we conducted text mining to categorize research related to IoT security and identify its trends. Text mining, also known as knowledge discovery from text, relies on various text analyses and processes to extract meaningful information from unstructured text data using natural language processing [51, 52]. In this study, we conducted STM-based topic modeling.

*3.2.1. STM-Based Topic Extraction to Classify Topics in IoT Security.* Topic modeling is an unsupervised learning method to determine and classify topics underlying textual data. The STM proposed by Roberts et al. [53] is a modified and extended version of the latent Dirichlet allocation, the most widely used topic modeling method. The STM determines the distribution of words constituting a topic based on the frequency of words in a document along with metadata (e.g., author's gender and age, publication year). The STM estimates the correlation between topics using the covariance matrix of the corresponding logistic normal distribution [53]. Figure 3 illustrates the STM, which can be divided into three components: a topic prevalence model that controls how words are allocated to topics as a function of covariates; a topical content model that controls the frequency of the terms in each topic as a function of the covariates; and a core language model [54].

According to Roberts et al. [53], given the number of topics (K), observed words and design matrices $\{w_{d,n}\}$, topic prevalence (X), topical content (Y), and K-dimensional hyperparameter vector ($\sigma$), data generation for document d can be modeled as

$$\gamma_k \sim \text{Normal}_p\left(0, \sigma_k^2 I_p\right), \quad \text{for } k = 1, \ldots, K-1, \tag{5}$$

$$\theta_d \sim \text{LogisticNormal}_{K-1}\left(\Gamma' x_d', \Sigma\right), \tag{6}$$

$$Z_{d,n} \sim \text{Multinominal}_K\left(\theta_d\right), \quad \text{for } n = 1, \ldots, N_d, \tag{7}$$

$$W_{d,n} \sim \text{Multinominal}_V\left(\beta_{Z_{d,n}}\right) \quad \text{for } n = 1, \ldots, N_d, \tag{8}$$

$$\beta_{d,k,v} = \frac{\exp\left(m_v + K_{k,v}^{(t)} + K_{y_d,v}^{(c)} + K_{y_d,k,v}^{(i)}\right)}{\sum_v \exp\left(m_v + K_{k,v}^{(t)} + K_{y_d,v}^{(c)} + K_{y_d,k,v}^{(i)}\right)}, \quad \text{for } v = 1, \ldots, V \text{ and } k = 1, \ldots, K, \tag{9}$$

where $\Gamma = [\gamma_1 | \ldots | \gamma_K]$ is a $P \times (K-1)$ matrix of coefficients for the topic prevalence model specified by equations (5) and (6), and $\left\{K_{\cdot,\cdot}^{(t)}, K_{\cdot,\cdot}^{(c)}, K_{\cdot,\cdot}^{(i)}\right\}$ is a collection of coefficients for the topical content model specified by equation (9). Equations (7) and (8) constitute the core language model.

In topic extraction, it is essential to determine the optimal number of topics ($K$) for the STM [55, 56]. To this end, the STM provides useful indicators, with the most widely used being the held-out likelihood and semantic coherence. From Figure 4, as the number of topics gradually increases from 5 to 20, we can determine the point where both the held-out likelihood and semantic coherence have high values [56], obtaining 12 as the optimal number of topics.

To interpret the topics derived according to their optimal quantity in the STM, main words representing each topic can be analyzed. We selected the main words of a topic according to four criteria: highest probability, frequency and exclusivity, lift weight, and score. Highest probability words are the upper words in the topic-word distribution. Frequency and exclusivity words are those derived using the weighted harmonic mean of the word rank, which reflects frequently used and exclusive words in a topic. Lift-weight words are derived by assigning high weights to less frequent words in other topics. The score is obtained by dividing the log frequency of a specific word in a specific topic by the log frequency of that word in other topics. To extract and analyze latent topics related to IoT security from the abstracts of the analyzed articles, we implemented the STM on the *R* software [55].

*3.2.2. STM-Based Trend Estimation of Topics in IoT Security.* We identified hot topics with uptrends and cold topics with downtrends in IoT security. The trend of a topic was estimated by setting the publication year as the covariate for that topic.

## 4. Results and Discussion

*4.1. Identification of Leading Researchers in IoT Security.* The results from the co-citation network analysis are shown in Figure 5. We analyzed and visualized the co-citation network using CitNetExplorer, obtaining 8 clusters of 52

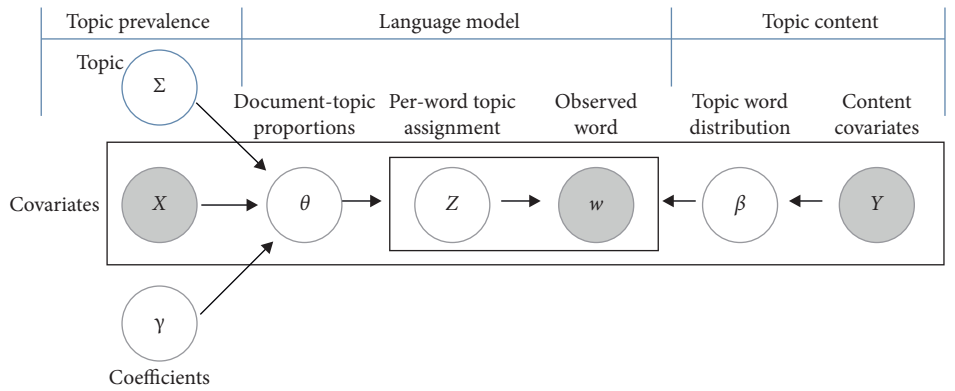



Figure 3: Diagram of STM concepts and processes.

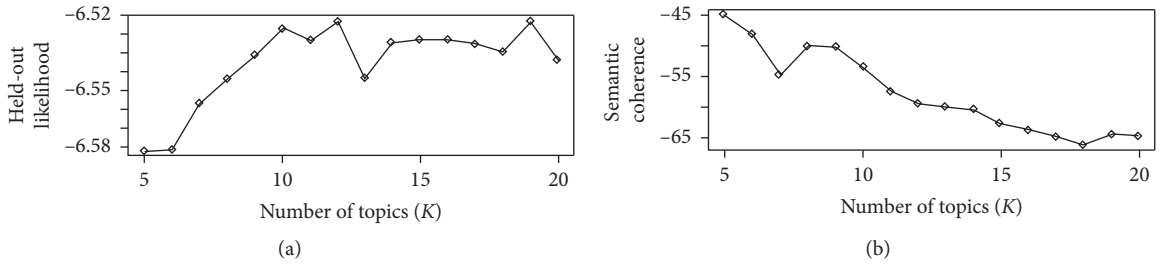



Figure 4: Diagnostic indicators to determine the optimal number of topics. (a) Held-out likelihood. (b) Semantic coherence.

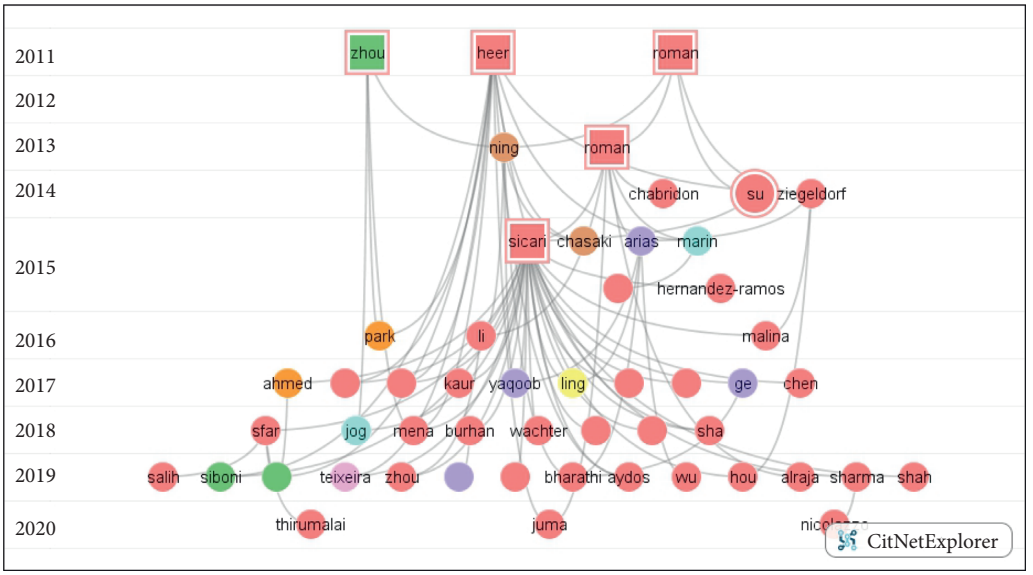



Figure 5: Co-citation network with the 52 most frequently cited publications grouped in 8 clusters (one color per cluster). The network was obtained using CitNetExplorer.

frequently cited publications. In the co-citation network, highly relevant clusters are located close together. Thus, the 8 clusters are closely related, as can be seen from the unseparated location of the nodes in the cluster. The articles on IoT security by Heer et al. [57] and Roman et al. [58] received high attention in the research community since 2011. The study with the highest citation score was authored by Sicari et al. [59] and published in 2015.

*4.2. Keyword Clustering and Evolution of Research on IoT Security.* From the 3,142 keywords in the 1,365 studies, 147 were derived by setting the minimum number of

occurrences of a keyword to 5, and the keyword co-occurrence network analysis was performed on 146 keywords, excluding IoT, which was present in all the studies given its use with Boolean operation AND during the search.

Figure 6 shows the obtained keyword co-occurrence network with 10 clusters, and Table 4 summarizes the network and cluster information. In Figure 6, the node size is proportional to the number of occurrences of the corresponding keyword, and the link thickness is proportional to the weight of the links connecting the nodes. The node color represents the cluster containing that node.

The main keywords of cluster 1, represented by red nodes, are "sdn," "machine learning," "trust," "attacks," "ddos," and "secure routing." This cluster was summarized as the study on the introduction of artificial intelligence (e.g., ML and deep learning) to improve IoT security performance. There is increasing interest in research to improve security by introducing ML or deep learning to detect DDoS (distributed denial-of-service) attacks, malicious code, abnormal behavior, and abnormal energy consumption for IoT devices [60–66]. There was also a study aimed to ensure secure content-sharing in an IoT environment by applying ML to explore the social trust of smart device users [67, 68].

Cluster 2, represented by green nodes, consists of main keywords "ecc," "encryption," "cryptography," "aes," "energy efficiency," and "lightweight cryptography." This cluster is associated with lightweight encryption for resource-constrained IoT devices, such as those with a small size, limited computing power, and low-power consumption. Research on lightweight encryption algorithms has been conducted in relation to data and personal information security in a resource-limited environment of smart devices. The advanced encryption standard (AES) and error-correcting codes (ECC) are mainly used as basic lightweight encryption elements. Various studies have been aimed to optimize lightweight encryption while balancing security and performance management [69–76].

In cluster 3, represented by blue nodes, "privacy preservation," "cloud computing," "fog computing," "edge computing," "data privacy," and "differential privacy" are the main keywords. This cluster can be summarized with the topic of privacy preservation in IoT devices. The crowd-sensing mode of smart M-IoT, a new paradigm of IoT, collects and delivers more privacy data. Thus, privacy preservation is becoming more important [77–79]. In addition, intelligent IoT applications enhanced with cloud, edge, and fog computing increasingly deal with personal information to provide intelligent services, and many studies on personal information protection and data protection are being conducted [80–83]. Among the personal information protection approaches, differential privacy is gaining attention as a mechanism to provide intelligent services by grasping user behavior patterns without infringing on personal information by adding noise to prevent the identification of personal information [81, 84–88].

Cluster 4, represented by yellow nodes, consists of main keywords, "wsn," "cps," "coap," "6lowpan," "smart object," and "sensor node." This cluster is related to studies on secure communication of smart objects in wireless sensor networks (WSNs). To transmit the information measured by sensor nodes in smart M-IoT, security is essential [89–91]. In this regard, studies on the use of IPSec/IPv6 and OpenSSL in virtual private networks have been performed to protect smart objects and provide end-to-end security [92]. The same is true for studies on end-to-end security framework development of the Constrained Application Protocol (CoAP) [93–95] and on frameworks in which smart-object users designate privacy preferences to protect personal information generated and consumed by smart objects [96]. Smart objects that have recently attracted attention are vehicles that are equipped with various sensor devices, actuators, GPS (global positioning system) receivers, and micro-embedded computers to collect, process, and transmit vast amounts of data [97, 98]. Vehicular sensor networks provide connected sensor devices that collect data and enable safer and more fluid road traffic [99]. The Internet-of-vehicles concept supports real-time vehicle-to-everything (V2X) wireless communication based on fog and edge computing [100–102]. Therefore, safe data transmission and privacy protection in vehicles, which are now smart objects, play an essential role in their development.

In cluster 5, represented by purple nodes, the main keywords are "key management," "signcryption," "elliptic curves," and "digital signature." This cluster is thus related to digital signcryption. Digital signature encryption has been investigated on algorithms, such as the elliptic curve digital-signature algorithm, digital-signature mobile applications, and digital-signature systems, to achieve document integrity and provide nonrepudiation security services in a distributed computing environment [103–107]. It is also important to satisfy reliability and confidentiality requirements of crowdsourced data [108, 109].

Cluster 6, represented by cyan nodes, comprises keywords "smart home," "raspberry pi," "arduino," and "face detection." This cluster can be described as building safe smart homes in an IoT environment. Wireless communications and sensor technologies, key components of IoT applications, are prerequisites for the security and confidentiality of smart homes [110, 111]. Before data transmission through the Session Initiation Protocol (SIP) in a home network, mutual safety verification should be conducted between devices to block advance devices that may cause risks. To this end, a secure trust relationship should be established between smart home devices, external smart devices, and other IoT devices [112–114]. A study has been conducted to design a secure IoT microcontroller module using the Raspberry Pi platform and various IoT sensors [115–117]. To achieve flexible device utilization, heterogeneous device interoperability, security enhancement of smart homes, and software-defined networks (SDN) have been applied [118, 119].

In cluster 7, represented by orange nodes, the main keywords are "privacy," "healthcare," "information security," "e-health," and "wban." This cluster can be related to IoT-based healthcare system security. As medical information systems manage patient data, data security and privacy protection are important. In IoT-based healthcare, studies on encryption and authentication protocols for user

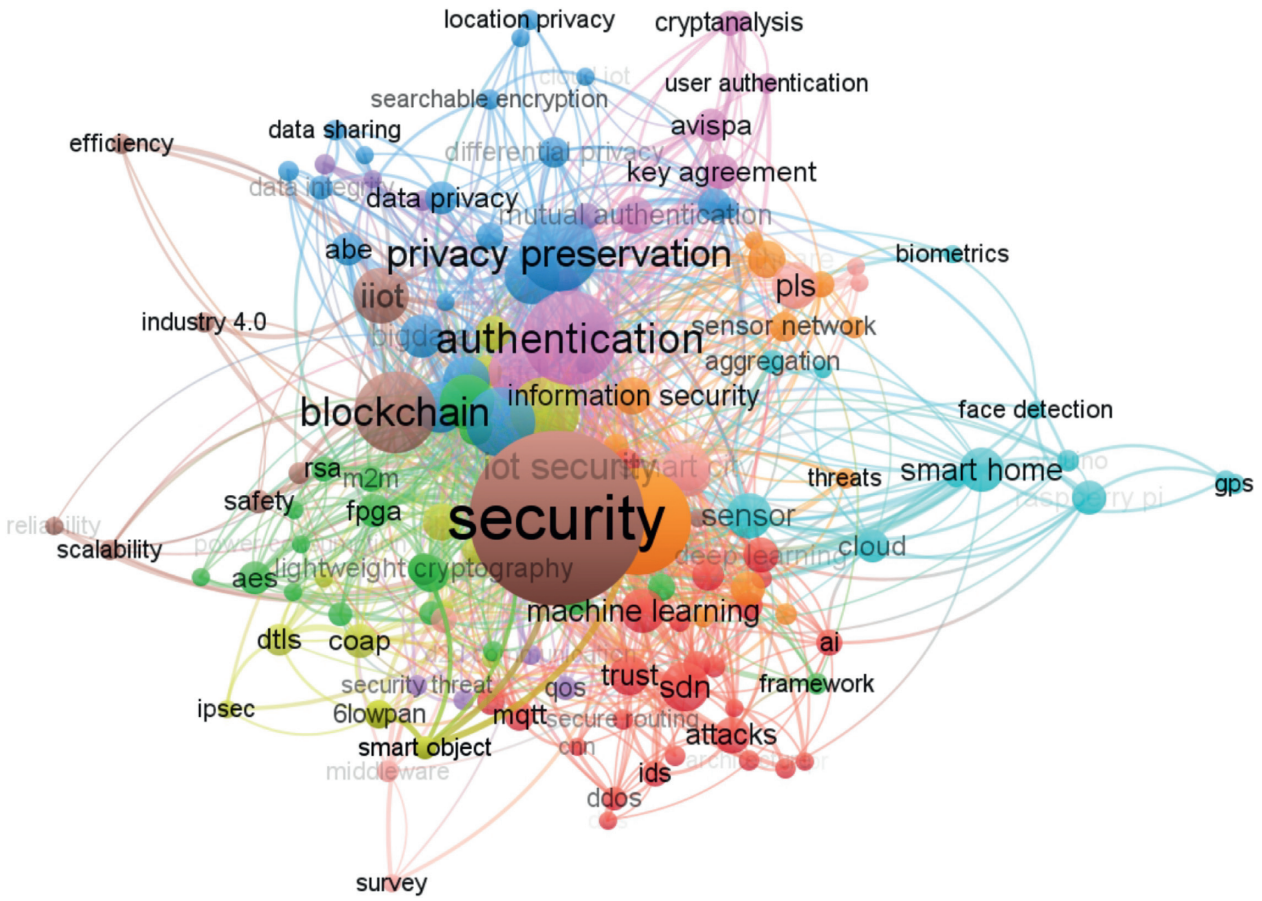



Figure 6: Keyword co-occurrence network obtained using VOSviewer.

Table 4: Specifications of keyword co-occurrence network.

| Cluster | Keywords | *X* | *Y* | Weight (occurrences) | Weight (links) | Weight (total link strength) |
|---|---|---|---|---|---|---|
| 1 | SDN | 0.292 | −0.598 | 32 | 35 | 30 |
| | Machine learning | 0.178 | −0.393 | 27 | 29 | 23 |
| | Deep learning | 0.498 | −0.244 | 17 | 25 | 16 |
| | Game theory | 0.322 | −0.537 | 11 | 17 | 9 |
| | Social IoT | 0.464 | −0.378 | 11 | 11 | 8 |
| 2 | ECC | −0.294 | 0.170 | 39 | 46 | 37 |
| | 5G | −0.057 | −0.207 | 16 | 25 | 15 |
| | Lightweight cryptography | −0.412 | −0.279 | 12 | 13 | 10 |
| | Lightweight encryption | −0.763 | −0.342 | 5 | 9 | 4 |
| 3 | Privacy preservation | −0.047 | 0.570 | 79 | 49 | 53 |
| | Cloud computing | −0.203 | 0.114 | 62 | 46 | 55 |
| | Fog computing | −0.120 | 0.506 | 39 | 39 | 35 |
| | Edge computing | −0.296 | 0.228 | 29 | 38 | 27 |
| 4 | WSN | −0.084 | 0.142 | 62 | 55 | 50 |
| | CPS (Cyber-physical systems) | −0.363 | −0.155 | 20 | 24 | 19 |
| | IoT device | −0.021 | −0.345 | 9 | 12 | 8 |
| | Smart object | −0.409 | −0.759 | 6 | 8 | 6 |
| 5 | Key management | 0.045 | 0.366 | 15 | 25 | 15 |
| | Authentication protocol | 0.029 | 0.668 | 10 | 10 | 9 |
| | Signcryption | −0.677 | 0.809 | 6 | 8 | 6 |
| | Digital signature | −0.549 | 0.769 | 5 | 10 | 4 |

Table 4: Continued.

| Cluster | Keywords | $X$ | $Y$ | Weight (occurrences) | Weight (links) | Weight (total link strength) |
|---|---|---|---|---|---|---|
| 6 | Sensor | 0.465 | −0.137 | 28 | 43 | 27 |
| | Smart home | 1.090 | −0.013 | 27 | 26 | 23 |
| | Raspberry Pi | 1.378 | −0.086 | 16 | 10 | 9 |
| | Arduino | 1.323 | 0.012 | 7 | 10 | 7 |
| 7 | Privacy | 0.163 | −0.150 | 138 | 82 | 126 |
| | Healthcare | 0.514 | 0.553 | 20 | 22 | 17 |
| | Information security | 0.150 | 0.185 | 20 | 21 | 14 |
| | E-health | 0.660 | 0.484 | 10 | 20 | 10 |
| 8 | Security | −0.052 | −0.142 | 360 | 119 | 306 |
| | Blockchain | −0.487 | 0.141 | 86 | 57 | 68 |
| | Industrial IoT | −0.525 | 0.456 | 41 | 37 | 37 |
| | Smart contract | −0.746 | −0.022 | 7 | 11 | 7 |
| 9 | Mutual authentication | 0.157 | 0.673 | 19 | 25 | 17 |
| | Key agreement | 0.391 | 0.789 | 17 | 21 | 17 |
| | BAN (Burrows–Abadi–Needham) logic | 0.441 | 1.195 | 6 | 11 | 5 |
| | User authentication | 0.512 | 1.028 | 6 | 9 | 6 |
| 10 | Smart city | 0.295 | −0.005 | 31 | 35 | 27 |
| | Cybersecurity | 0.306 | −0.028 | 23 | 32 | 20 |
| | Mobile edge computing | 0.761 | 0.491 | 5 | 7 | 5 |
| | Secure energy efficiency | 0.750 | 0.536 | 5 | 5 | 5 |

Note. Column keywords contain the four most representative words (from most to least important) for each cluster. Columns $X$ and Y indicate the coordinates in the corresponding axes of the keyword node on the network shown in Figure 6.

authentication [120–123] and data encryption for patient privacy protection [124–127] are relevant. Safe and efficient medical data retrieval is important for remote medical monitoring. Given the difficulty to collect medical data safely and efficiently owing to the resource limitations of IoT devices, various studies on providing medical services by combining IoT and edge clouds have been conducted [128, 129]. In addition, to collect data, aggregate them safely and efficiently, and transmit them to a server, a study has been conducted on a system leveraging fog computing [130, 131]. There is also a growing interest in introducing unmanned aerial vehicles (UAVs) as smart objects for collecting health data. In fact, UAVs can collect health data, encrypt them, and transmit them to authenticated body sensor hives using low-power secure communications [132].

In cluster 8, represented by brown nodes, the main keywords are "blockchain," "iiot," "safety," "smart contract," and "industry 4.0." This cluster can be described as a blockchain applied to IoT applications. It is essential to ensure the integrity of data generated in IoT environments. In this regard, research on blockchain-based encryption has been conducted [133–136]. Trust relationships must be established between disparate entities in the IoT ecosystem [137]. An analysis on the combination of blockchain and trust evaluation technologies has been conducted accordingly [138, 139]. Regarding Industry 4.0, the interest in industrial IoT (IIoT) is increasing. In particular, blockchain-based smart contracts have been studied. In addition, blockchains that provide transaction transparency, immutability, auditability, and high security for IoT-based international trade have been proposed [140, 141]. In recent years, the interest in decentralized security mechanisms based on blockchain has increased regarding the storage of important data generated by IoT systems [142, 143].

Cluster 9, represented by pink nodes, consists of main keywords "authentication," "rfid," "mutual authentication," "key agreement," and "user authentication." This cluster is thus related to multiple forms of authentication. Smart M-IoT environments establish networks that provide smart services based on user information. Therefore, the privacy of users and the confidentiality of sensitive data must be guaranteed. Device authentication, radio-frequency identification (RFID), and user authentication are security functions that must be provided in any IoT environment [144–151].

Cluster 10, represented by coral-pink nodes, has main keywords "smart city," "pls," "cybersecurity," "middleware," and "mobile-edge computing." This cluster can be summarized by security related to IoT-based smart cities. A smart city is an IoT application that manages a city with minimal or without human intervention and provides smart services. Beyond the smart home, it connects all sensors and smart objects at the city level to provide real-time smart services. Therefore, research on the protection of citizens' personal information [152–154], management of IoT devices in heterogeneous device network environments [155, 156], and integrated security solutions considering the entire security stack [157, 158] has been conducted.

We also conducted a co-occurrence keyword network considering the year of publication to find answer RQ2-2. Figure 7 shows the obtained network with temporal information (publication year) encoded as a color map. Until 2017, there were many keywords related to networks, such as "6lowpan," "dtls," "m2m communications," "ips," "rfid," "sensor networks," and "middleware." During the first half of 2018, many studies included keywords related to the security of data delivered over IoT applications, such as "privacy preservation," "authentication," and "data

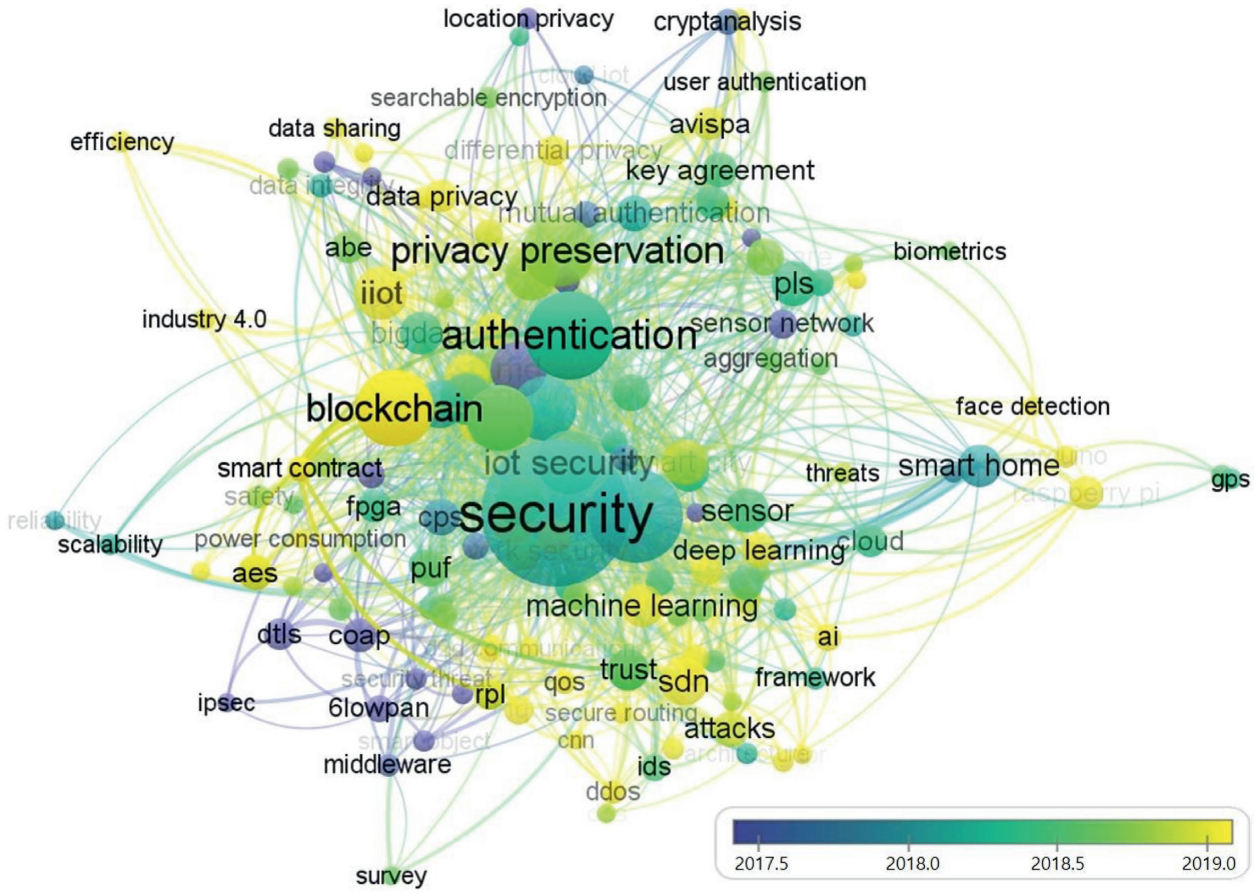



Figure 7: Keyword co-occurrence network reflecting temporal evolution. The network was obtained using VOSviewer.

integrity." During the second half of 2018, keywords, such as "trust," "fog computing," "healthcare," and "smart city," were prevalent. Since 2019, keywords related to the convergence of new technologies in the Industry 4.0 and other fields, such as "blockchain," "software-defined networking," "iiot," "machine learning," "deep learning," and "social iot," have become predominant.

*4.3. Identification of Topics in IoT Security.* Information about the identified topics is summarized in Table 5. For each topic, 10 top words were considered under four criteria: highest probability, frequency and exclusivity, lift weight, and score. The three most meaningful keywords per criterion are included in Table 5. We also created a label explaining each topic by analyzing the five studies with the highest proportion of contents related to that topic and containing its top words. We discussed with two IoT experts the selection of the top words and topic labels.

Topic 1 is related to understanding the characteristics of IoT across a variety of aspects and the analysis and discussion of security issues and solutions for the layers of IoT networks [159–169].

Topic 2 is related to encryption and authentication for securely sharing data in an IoT-based healthcare environment considering detailed access control. With the spread of IoT applications, smart health is becoming an attractive paradigm. As it deals with user information and sensitive medical information, the security and mutual authentication of medical sensor devices for personal information protection, encryption, and real-time monitoring are key elements [125, 170–181].

Topic 3 is related to secure and lightweight encryption designs tailored for IoT applications. Lightweight encryption with low processing time and low power consumption is required to protect and secure data transmissions of resource-constrained IoT devices. Block encryption, such as AES and S-box, Galois Counter Mode, and physical unclonable functions, are being utilized, evaluated, and proposed [70, 72, 73, 182–188].

Topic 4 is related to security using ML. Considering the heterogeneity of IoT networks and devices, it has become more common for SDN technologies to be integrated into IoT applications to form flexible and manageable architecture. When a network attack occurs in an SDN, ML can be introduced as a detection technology to dynamically control and route the communication flow. Recently, studies using ML to detect and automatically respond to DDoS attacks, abnormal patterns, and data leaks against IoT networks and devices have increased [60, 189–199].

Topic 5 is related to risk assessment and prioritization of IoT security threats. For a secure IoT environment, various

Table 5: STM-based topic extraction results and top words per topic according to four criteria.

| Topic (proportions) | Top words | | | | Topic label |
|---|---|---|---|---|---|
| | Highest probability | Frequency and exclusivity | Lift weight | Score | |
| 1 (15%) | Security<br>Issue<br>Challenge | Discuss<br>Issue<br>Challenge | Attitude<br>Society<br>Taxonomy | Layer<br>Security<br>WSN | IoT security issues |
| 2 (9%) | Data<br>Access<br>Encrypt | Patient<br>Medical<br>Healthcare | Biometric<br>Ciphertext-policy<br>CP-ABE (Ciphertext-policy attribute-based encryption | Patient<br>Medical<br>Signature | Secure data sharing for healthcare |
| 3 (6.5%) | Algorithm<br>Encrypt<br>Power | PUF (Physical unclonable function<br>FPGA (field programmable gate array)<br>S-box | Scalar<br>Simeck<br>AES-GCM (advanced encryption standard-Galois counter mode) | PUF<br>S-box (substitution-box)<br>FPGA | Lightweight encryption |
| 4 (7.6%) | Device<br>Attack<br>Detect | SDN<br>Learning<br>Intrusion | OpenFlow<br>SDN-IoT<br>Cyber-attack | Detect<br>Attack<br>SDN | Security with ML |
| 5 (9%) | Model<br>Develop<br>Risk | Risk<br>Assess<br>Measure | ANP (analytic network process)<br>Casual<br>Diagram | Workforce<br>Risk<br>Assess | Risk assessment |
| 6 (8.5%) | Authentication<br>Protocol<br>Attack | Authentication<br>Mutual<br>Protocol | BAN<br>PMIPv6 (Proxy mobile IPv6)<br>AVISPA (automated validation of Internet security protocols) | Authentication<br>Protocol<br>Mutual | Mutual authentication protocol |
| 7 (7.4%) | Cloud<br>Edge<br>Fog | Edge<br>Eavesdropping<br>Fog | Colluding<br>SSR (secrecy sum rate)<br>Tensor-based | Fog<br>Eavesdropping<br>Offload | MEC security |
| 8 (8%) | Node<br>Energy | Rout<br>RPL (routing protocol for low-power and lossy networks) | Acyclic<br>Leach | Rout<br>Energy | Energy-efficient routing protocol |
| 9 (6.5%) | Rout<br>Sensor<br>Control<br>Home | Cluster<br>Camera<br>Arduino<br>Raspberry | RPL<br>Burglar<br>Caution<br>Diabetes | Cluster<br>Arduino<br>Camera<br>Gadget | Secure home automation system |
| 10 (6%) | Smart<br>Blockchain<br>Home | City<br>Blockchain<br>Smart | Commerce<br>Campus<br>Cart | Blockchain<br>Smart<br>City | Integration of blockchain and IoT |
| 11 (8.5%) | Privacy<br>User<br>Collect | Privacy<br>Preserving<br>User | Cyber-physics<br>Mile<br>Participant | Privacy<br>Preserving<br>Data | Privacy preservation |
| 12 (8%) | Device<br>Protocol<br>Key | DTLS (datagram transport layer security)<br>CoAP<br>End-end | EDHOC (ephemeral Diffie-Hellman over common open software environment)<br>Rekey<br>AEAD (authenticated encryption with associated data) | DTLS<br>TLS<br>CoAP | End-to-end security |

studies have prioritized security threats by applying approaches such as product-development life cycle, decision-making trial-and-evaluation laboratory, analytic network processing, and graph theory to develop risk assessment and management frameworks [200–207].

Topic 6 corresponds to research on the development of user mutual authentication protocols for social IoT, IoT-based Long-Term Evolution (LTE), LTE-advanced networks, WSNs, and NFC (near-field communication) payment systems [144, 208–218]. In addition, the verification of authentication protocols using software tools, such as BAN and AVISPA, has gained popularity [213, 214, 217, 219–221]. Recently, the target of authentication has gained attention for mobile smart objects, such as drones and vehicles [219, 221, 222].

Topic 7 is related to MEC security. MEC integrated with IoT applications offload computationally intensive tasks at the network edge. As the edges are susceptible to cyber

threats, there is a growing interest in their security. The main related studies include areas such as personal information protection and secure data collection, and transmission for MEC-supported IoT applications [223–241].

Topic 8 is related to the development of energy-efficient routing protocols that minimize the transmission power for routing between nodes in IoT networks. For instance, a routing protocol for low-power and lossy networks (RPL), a protocol for low-power and low-loss networks, and corresponding security methods have been developed [242–253].

Topic 9 is related to secure home automation systems toward automation, safety, and security through the control of home appliances and sensors. Research on this subject has two main subtopics. The first subtopic is related to security against cyberattacks in the home network [112, 254–259], and the second one is related to home automation providing safety against external physical intrusion [260–266].

Topic 10 is related to the adoption of blockchain in smart-IoT applications, such as smart contracts, smart inventory management, smart e-commerce, and smart shopping systems [140, 155, 267–279].

Topic 11 concerns privacy decisions and privacy preservation in the value chain of IoT data in environments where IoT devices collect personal data and forward them to third parties. Research on this subject has two main subtopics. The first subtopic is related to personal information security [280–283]. The second subtopic is related to the data value chain, including information related to the owner's perception of privacy protection and the right to make decisions about personal information protection [96, 284–287].

Topic 12 includes studies on transport protocols for end-to-end security [288–290]. To achieve end-to-end secure communication between an IoT back end and resource-limited smart things, various studies on communication protocols such as DTLS and CoAP [291, 292] and key setting protocols such as EDHOC have been conducted [293, 294].

*4.4. Trend Estimation of Topics in IoT Security.* To answer RQ5, we estimated the trends over time for each topic by setting the year as a covariate, obtaining the results shown in Figure 8. Topics with an upward trend (increasing influence) are topics 4 (security through ML), 7 (MEC security), 8 (energy-efficient routing protocols), and 10 (blockchain and IoT integration). On the other hand, topics 1 (IoT security issues), 5 (risk assessment), 6 (mutual authentication protocol), and 12 (end-to-end security)show a decreasing trend.

*4.5. Challenges and Future Perspectives.* We identify the evolution of keywords in Section 4.2. Figure 9 shows the part of Figure 6 containing the keywords (colored nodes) of clusters closely related to "blockchain," which is the core of keyword evolution, as identified in Figure 7.

In Figure 9, "blockchain" is connected to "machine learning," "deep learning," "ai," and "sdn" at the bottom-right area. Thus, there is a relation to topic 4. Node "edge computing" shown above "blockchain" can be linked to topic 7. In addition, "efficiency," which is connected to the upper-left area of "blockchain," and "rpl," which is connected at the bottom of the center area, can be related to topic 8. These results indicate that the trends obtained from keywords and topics suitably agree. Based on the analyzed studies and discussions, we summarize below challenges and future perspectives related to secure distributed smart M-IoT applications.

*4.5.1. Secure Distributed Framework for Smart M-IoT Applications.* Various studies on the integration of SDN, fog and edge computing, and blockchain have been conducted aiming to improve the security of IoT applications [270, 275, 276, 278, 295–302].

Medhane et al. [295] proposed a blockchain-enabled distributed security framework for next-generation IoT applications by implementing an edge cloud security framework using an SDN. The proposed framework consists of an IoT device layer, an edge cloud layer, and a blockchain-enabled SDN. Gateway nodes in the edge cloud layer act as access points for the distributed SDN and quickly detect attacks by analyzing real-time data received from IoT devices. All roaming IoT devices and SDN servers share data through blockchain technology. The proposed security framework shows improved results in terms of packet delivery rate, throughput, and delay compared with frameworks without blockchain, edge cloud, and SDN. The framework is also effective for data confidentiality, integrity, and availability. However, energy consumption has increased.

The blockchain-based decentralized security architecture proposed by Rathore et al. [298] is a layered model consisting of sensing, edge computing, fog computing, and cloud layers. The sensing layer comprises many smart devices and widely distributed sensing nodes that monitor various environments and activities in public infrastructure. The edge computing layer consists of low-power high-performance SDN switches at the edge of the network. Each SDN switch at the edge computing layer connects to multiple sensors, and the switch processes and analyzes the data traffic of sensors. The fog computing layer with several SDN controllers is connected to the SDN switch cluster at the edge computing layer and analyzes the processed data. The SDN controller of a fog computing node consists of four components: traffic flow analyzer, traffic flow classifier, blockchain-based attack detection module, and attack mitigation module. Learning attack detection in the fog computing layer can be distributed to reduce the computational overhead and provide a fast response through simultaneous computations. Moreover, the fog computing layer transmits the traffic analysis results to the cloud layer. This decentralized architecture improves the attack detection performance by dynamically updating the attack detection model of each fog computing node using blockchain technology. It also prevents single points of failure inherent to centralized architecture. However, there is an overhead for blockchain operations.

It remains necessary to develop a secure distributed IoT framework that integrates fog and edge computing, ML-

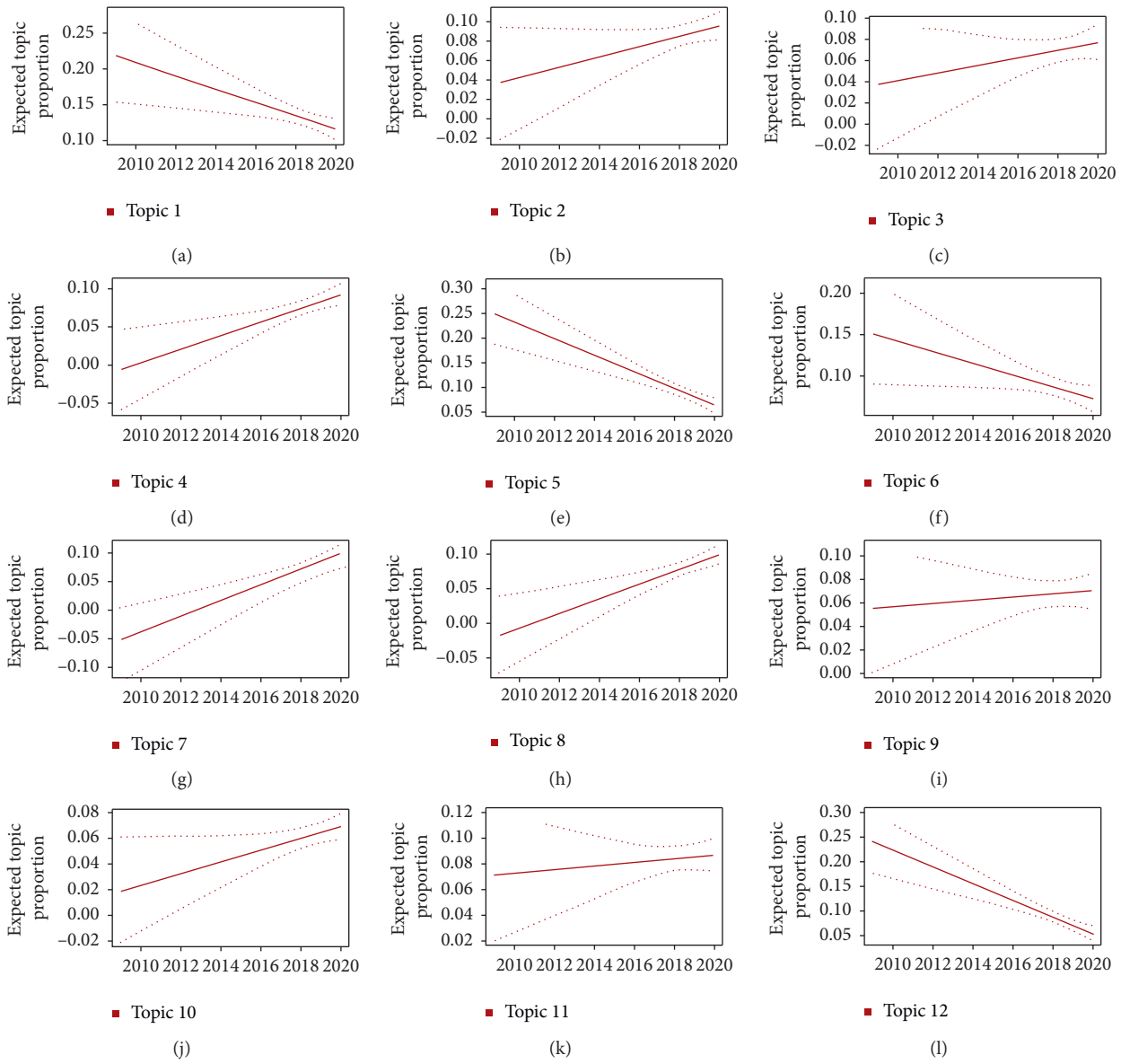



Figure 8: Topic trend estimation over time. We set the covariate to year and estimated the trends based on the change in the proportion of studies on each topic over time.

based SDN, and blockchain technology. Using fog and edge computing, the fog computing layer must analyze malicious traffic flows using ML algorithms to construct an intelligent attack detection model and dynamically update and manage traffic rules at edge computing nodes. This way, an ML-based SDN controller can enable fast attack detection. In addition, data privacy at the fog node level must be considered. The decentralized nature of blockchain supports secure distributed computing through the distributed trust concept. IoT devices and SDN servers can safely share data using blockchain [270, 295–298]. Therefore, a secure and energy-efficient blockchain-enabled architecture of ML-based SDN controllers for IoT networks is still required [303]. As new devices and applications are connected to IoT applications over time, unknown attacks can be developed. ML-based security is important to detect unknown attacks and respond properly in real time. In addition, in a secure distributed framework, IoT devices with limited resources can support routing protocols with high throughput, low latency, and low energy consumption. Thus, it remains necessary to develop a blockchain-based lightweight security protocol [281, 303].

*4.5.2. Smart Objects in Smart M-IoT Applications.* IoT devices can detect valuable data to build many intelligent applications. In addition, they can make important decisions to control their surroundings. Several IoT applications rely on end-to-end security between IoT devices and the cloud. However, realizing end-to-end security in IoT applications is difficult due to the wide variety of devices. In addition, most IoT devices have limited resources and cannot support heavy

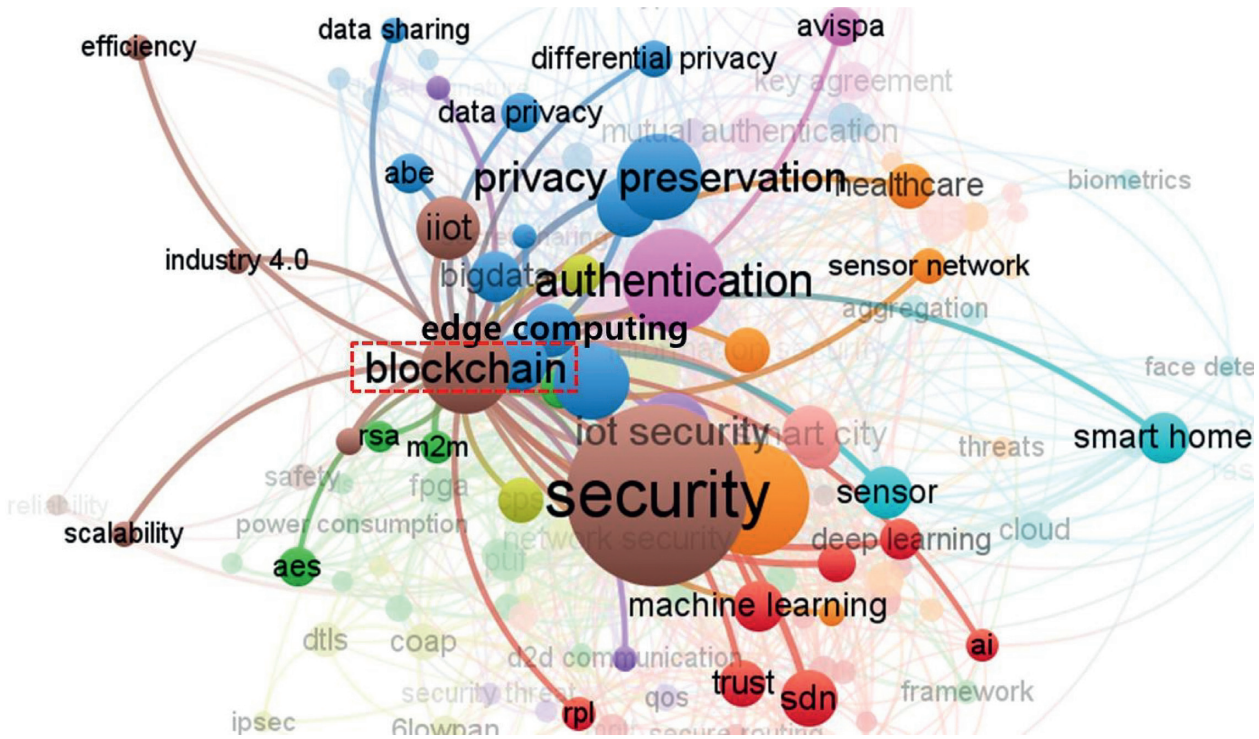



Figure 9: Keywords closely related to the keyword "blockchain" in Figure 6.

security applications such as firewalls. In [1], the introduction of edge computing into IoT device security for various applications is analyzed. Firewalls, intrusion detection systems, distributed traffic monitoring, attribute-based access control, and authentication protocols are analyzed at the edge computing layer for resource-limited IoT devices. To integrate edge computing, an algorithm and a lightweight secure communication protocol to establish trust between IoT devices and the edge should be first developed.

Talavera et al. [2] investigated security issues between the sensing layer and IoT devices and those at the IoT application layer, which involves smart homes, smart meters, smart cities, smart grids, and other solutions that directly handle end users and provide services. Therefore, unique security issues occur at this layer, such as data theft and privacy issues. Thus, a method to quantify and manage risk levels through rigorous penetration testing of IoT devices is required. Whenever IoT devices interact, a seamless authentication process must be implemented. To protect the user and environment data from being captured, mechanisms based on cryptographic techniques such as RSA, SHA256, or hash chain are needed. In addition, to increase the security level, Talavera et al. [2] recommend further development of recent technologies such as blockchain, fog and edge computing, and ML-based solutions.

Shin and Byun [3] proposed a privacy protection method for IoT devices in a smart city by applying edge computing. By processing data in near real time at the edge, they solve the heterogeneity problem of IoT devices and improve the overall performance, resulting in faster response times. Therefore, their method provides better quality of service for IoT applications.

To achieve smart applications, numerous IoT devices deployed around the world should generate large amounts of user and environment data. Consequently, much personal information can be leaked, posing a threat to individuals and the society as a whole. Therefore, IoT applications and their smart objects must be stable, secure, and robust. Smart objects that have attracted increasing interest in recent years include autonomous vehicles and UAVs. They have been combined with IoT to establish V2X communication and the Internet of drones. However, security concerns such as personal information protection, data encryption, and authentication remain to be addressed. Fog and edge computing, blockchain-based and SDN-enabled V2X communication, and Internet of drones can complete the available range of smart M-IoT services that include smart health, smart homes, smart cities, smart factories, smart agriculture, and smart transportation. As a result, more diverse smart services should be proposed, and the convergence of various fields will be promoted [101, 102, 132, 221, 302].

## 5. Conclusions

For the successful introduction and spread of smart M-IoT applications, security is an essential requirement. Many review studies have been conducted to understand IoT security. However, many of them have focused on specific areas of IoT security. In addition, existing studies have primarily provided in-depth professional content analysis. In contrast, we provide comprehensive initial insights in a different approach

than previous studies. Our study provides IoT security keyword clusters, keyword trends, topic classification, and topic trends to interested researchers. Then, we synthesize and explain keyword evolution and topics with increasing influence. We recommend pursuing research on the development of a secure decentralized framework integrating edge computing, ML-based SDN, and blockchain, as well as research on vehicles and UAVs as smart M-IoT objects.

Our research has various limitations. For instance, when collecting articles to be analyzed, a keyword search was performed on the article titles. Therefore, articles implicitly related to IoT security may be omitted from this study. Nevertheless, our study provides new researchers with comprehensive initial insights on the security required for smart M-IoT. In addition, this study has demonstrated the application of a method to perform a systematic mapping study using big data mining to process many documents. This method can be applied to systematic reviews in other fields.

## Data Availability

The list of the 1,365 research articles used in this study is available upon request to the corresponding author, at j.ann.lee@yonsei.ac.kr.

## Conflicts of Interest

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

## References


[1] H. Elazhary, "Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: disambiguation and research directions," *Journal of Network and Computer Applications*, vol. 128, pp. 105–140, 2019.

[2] L. E. Talavera, M. Endler, and I. Vasconcelos, "The mobile hub concept: enabling applications for the internet of mobile things," in *Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 123–128, St. Louis, MO, USA, 2015.

[3] T. Shin and J. Byun, "Design and implementation of a vehicle social enabler based on social Internet of things," *Mobile Information Systems*, vol. 2016, Article ID 4102163, 11 pages, 2016.

[4] A. R. Dargazany, P. Stegagno, and K. Mankodiya, "WearableDL: wearable internet-of-things and deep learning for big data analytics—concept, literature, and future," *Mobile Information Systems*, vol. 2018, Article ID 8125126, 20 pages, 2018.

[5] H.-K. Ra, H. J. Yoon, S. H. Son et al., "HealthNode: software framework for efficiently designing and developing cloud-based healthcare applications," *Mobile Information Systems*, vol. 2018, Article ID 6071580, 12 pages, 2018.

[6] H.-S. Kim, S. Yun, H. Kim et al., "An efficient SDN multicast architecture for dynamic industrial IoT environments," *Mobile Information Systems*, vol. 2018, Article ID 8482467, 11 pages, 2018.

[7] V. Sharma, I. You, K. Andersson, F. Palmieri, M. H. Rehmani, and J. Lim, "Security, privacy and trust for smart mobile-internet of things (M-IoT): a survey," *IEEE Access*, vol. 8, pp. 167123–167163, 2020.

[8] S. Marcos-Pablos, A. García-Holgado, and F. J. García-Peñalvo, *Guidelines for Performing Systematic Research Projects Reviews*, 2020.

[9] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[10] S. Keele, "Guidelines for performing systematic literature reviews in software engineering," Technical Report, EBSE, Goyang-si, South Korea, 2007.

[11] D. Gough, J. Thomas, and S. Oliver, "Clarifying differences between review designs and methods," *Systematic Reviews*, vol. 1, no. 1, p. 28, 2012.

[12] B. Kitchenham, R. Pretorius, D. Budgen et al., "Systematic literature reviews in software engineering - a tertiary study," *Information and Software Technology*, vol. 52, no. 8, pp. 792–805, 2010.

[13] B. A. Kitchenham, D. Budgen, and O. Pearl Brereton, "Using mapping studies as the basis for further research - a participant-observer case study," *Information and Software Technology*, vol. 53, no. 6, pp. 638–651, 2011.

[14] C. Marshall and P. Brereton, "Tools to support systematic literature reviews in software engineering: a mapping study," in *Proceedings of the 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 296–299, IEEE, Baltimore, MD, USA, 2013.

[15] K. Petersen, R. Feldt, S. Mujtaba et al., "Systematic mapping studies in software engineering," in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, vol. 12, pp. 1–10, Trondheim, Norway, 2008.

[16] E. Zavala, X. Franch, and J. Marco, "Adaptive monitoring: a systematic mapping," *Information and Software Technology*, vol. 105, pp. 161–189, 2019.

[17] K. R. Felizardo, N. Salleh, R. M. Martins et al., "Using visual text mining to support the study selection activity in systematic literature reviews," in *Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement*, pp. 77–86, Alberta, Canada, 2011.

[18] B. A. Kitchenham, D. Budgen, and O. P. Brereton, "The value of mapping studies–A participant-observer case study," in *Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering (Ease)*, pp. 1–9, Ciudad Real, Spain, 2010.

[19] A. Wang, P. Wang, X. Miao et al., "A review on non-terrestrial wireless technologies for Smart City Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 16, no. 6, Article ID 1550147720936824, 2020.

[20] S. L. Ullo and G. Sinha, "Advances in smart environment monitoring systems using IoT and sensors," *Sensors*, vol. 20, no. 11, p. 3113, 2020.

[21] K. T. Kadhim, A. M. Alsahlany, S. M. Wadi et al., "An overview of patient's health status monitoring system based on internet of things (IoT)," *Wireless Personal Communications*, vol. 114, pp. 1–28, 2020.

[22] Y. A. Qadri, A. Nauman, Y. B. Zikria, A. V. Vasilakos, and S. W. Kim, "The future of healthcare internet of things: a survey of emerging technologies," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1121–1167, 2020.

[23] K. L. Raju and V. Vijayaraghavan, "IoT technologies in agricultural environment: a survey," *Wireless Personal Communications*, vol. 113, 2020.

[24] H. Farooq, H. U. Rehman, A. Javed, M. Shoukat, and S. Dudely, "A review on smart IoT based farming," *Annals of Emerging Technologies in Computing*, vol. 4, no. 3, pp. 17–28, 2020.

[25] K. Kiela, V. Barzdenas, M. Jurgo et al., "Review of V2X-IoT standards and frameworks for ITS applications," *Applied Sciences*, vol. 10, no. 12, p. 4314, 2020.

[26] M. A. Rahim, M. A. Rahman, M. Rahman et al., "Evolution of IoT-enabled connectivity and applications in automotive industry: a review," *Vehicular Communications*, vol. 27, Article ID 100285, 2020.

[27] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: a survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.

[28] D. Mendez Mena, I. Papapanagiotou, and B. Yang, "Internet of things: survey on security," *Information Security Journal: A Global Perspective*, vol. 27, no. 3, pp. 162–182, 2018.

[29] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[30] E. L. C. Macedo, E. A. R. de Oliveira, F. H. Silva et al., "On the security aspects of Internet of Things: a systematic literature review," *Journal of Communications and Networks*, vol. 21, no. 5, pp. 444–457, 2019.

[31] S. Hameed, F. I. Khan, and B. Hameed, "Understanding security requirements and challenges in Internet of Things (IoT): a review," *Journal of Computer Networks and Communications*, vol. 2019, Article ID 9629381, 14 pages, 2019.

[32] M. A. Obaidat, S. Obeidat, J. Holst et al., "A comprehensive and systematic survey on the internet of things: security and privacy challenges, security frameworks, enabling technologies, threats, vulnerabilities and countermeasures," *Computers*, vol. 9, no. 2, 2020.

[33] 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.

[34] J. Arshad, M. A. Azad, R. Amad, K. Salah, M. Alazab, and R. Iqbal, "A review of performance, energy and privacy of intrusion detection systems for IoT," *Electronics*, vol. 9, no. 4, p. 629, 2020.

[35] X. Yao, F. Farha, R. Li et al., "Security and privacy issues of physical objects in the IoT: challenges and opportunities," *Digital Communications and Networks*, 2020, in Press.

[36] B. Liao, Y. Ali, S. Nazir, L. He, and H. U. Khan, "Security analysis of IoT devices by using mobile computing: a systematic literature review," *IEEE Access*, vol. 8, pp. 120331–120350, 2020.

[37] A. Sharma, E. S. Pilli, A. P. Mazumdar et al., "Towards trustworthy Internet of Things: a survey on Trust Management applications and schemes," *Computer Communications*, vol. 160, 2020.

[38] M. A. Ferrag, L. Shu, X. Yang, A. Derhab, and L. Maglaras, "Security and privacy for green IoT-based agriculture: review, blockchain solutions, and challenges," *IEEE Access*, vol. 8, pp. 32031–32053, 2020.

[39] R. R. Braam, H. F. Moed, and A. F. J. Van Raan, "Mapping of science by combined co-citation and word analysis. I. Structural aspects," *Journal of the American Society for Information Science*, vol. 42, no. 4, pp. 233–251, 1991.

[40] Q. He, "Knowledge discovery through co-word analysis," *Library Trends*, vol. 48, no. 1, pp. 133–159, 1999.

[41] S. Radhakrishnan, S. Erbis, J. A. Isaacs et al., "Novel keyword co-occurrence network-based methods to foster systematic reviews of scientific literature," *PloS One*, vol. 12, no. 3, Article ID e0172778, 2017.

[42] L. Waltman and N. J. Van Eck, "A new methodology for constructing a publication-level classification system of science," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 12, pp. 2378–2392, 2012.

[43] N. J. Van Eck and L. Waltman, "Citation-based clustering of publications using CitNetExplorer and VOSviewer," *Scientometrics*, vol. 111, no. 2, pp. 1053–1070, 2017.

[44] R. Klavans and K. W. Boyack, "Which type of citation analysis generates the most accurate taxonomy of scientific and technical knowledge?" *Journal of the Association for Information Science and Technology*, vol. 68, no. 4, pp. 984–998, 2017.

[45] V. A. Traag, P. Van Dooren, and Y. Nesterov, "Narrow scope for resolution-limit-free community detection," *Physical Review E*, vol. 84, no. 1, Article ID 016114, 2011.

[46] N. J. Van Eck and L. Waltman, "CitNetExplorer: a new software tool for analyzing and visualizing citation networks," *Journal of Informetrics*, vol. 8, no. 4, pp. 802–823, 2014.

[47] N. J. Van Eck and L. Waltman, "Software survey: VOSviewer, a computer program for bibliometric mapping," *Scientometrics*, vol. 84, no. 2, pp. 523–538, 2010.

[48] N. J. Eck and L. Waltman, "How to normalize cooccurrence data? An analysis of some well-known similarity measures," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 8, pp. 1635–1651, 2009.

[49] L. Waltman, N. J. Van Eck, and E. C. M. Noyons, "A unified approach to mapping and clustering of bibliometric networks," *Journal of Informetrics*, vol. 4, no. 4, pp. 629–635, 2010.

[50] L. Waltman and N. J. Van Eck, "A smart local moving algorithm for large-scale modularity-based community detection," *The European Physical Journal B*, vol. 86, no. 11, p. 471, 2013.

[51] J. Y. Lee, "Deep learning research trend analysis using text mining," *International Journal of Advanced Culture Technology*, vol. 7, no. 4, pp. 295–301, 2019.

[52] J. Lee, "A study on research trend analysis and topic class prediction of digital transformation using text mining," *International Journal of Advanced Smart Convergence*, vol. 8, no. 2, pp. 183–190, 2019.

[53] M. E. Roberts, B. M. Stewart, D. Tingley et al., "The structural topic model and applied social science," in *Advances in Neural Information Processing Systems Workshop on Topic Models: Computation, Application, and Evaluation*Harrahs and Harveys, Lake Tahoe, NV, USA, 2013.

[54] M. E. Roberts, B. M. Stewart, and E. M. Airoldi, "A model of text for experimentation in the social sciences," *Journal of the American Statistical Association*, vol. 111, no. 515, pp. 988–1003, 2016.

[55] M. E. Roberts, B. M. Stewart, and D. Tingley, "stm: R package for structural topic models," *Journal of Statistical Software*, vol. 10, no. 2, pp. 1–40, 2014.

[56] H. M. Wallach, I. Murray, R. Salakhutdinov et al., "Evaluation methods for topic models," in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1105–1112, Montreal, Canada, 2009.

[57] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, and K. Wehrle, "Security challenges in the IP-based internet of things," *Wireless Personal Communications*, vol. 61, no. 3, pp. 527–542, 2011.

[58] R. Roman, P. Najera, and J. Lopez, "Securing the internet of things," *Computer*, vol. 44, no. 9, pp. 51–58, 2011.

[59] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: the road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[60] M. V. O. de Assis, L. F. Carvalho, J. J. P. C. Rodrigues, J. Lloret, and M. L. Proença Jr, "Near real-time security system applied to SDN environments in IoT networks using convolutional neural network," *Computers & Electrical Engineering*, vol. 86, Article ID 106738, 2020.

[61] H. W. Kim and E. H. Song, "Behavior-based malware detection using deep learning for improve security of iot infrastructure," *International Journal of Advanced Science and Technology*, vol. 28, no. 5, pp. 128–134, 2019.

[62] F. Li, A. Shinde, Y. Shi, J. Ye, X.-Y. Li, and W. Song, "System statistics learning-based IoT security: feasibility and suitability," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6396–6403, 2019.

[63] F. Li, Y. Shi, A. Shinde, J. Ye, and W. Song, "Enhanced cyber-physical security in internet of things through energy auditing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5224–5231, 2019.

[64] B. Chatterjee, D. Das, S. Maity et al., "RF-PUF: enhancing IoT security through authentication of wireless nodes using in-situ machine learning," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 388–398, 2019.

[65] I. Kotenko, I. Saenko, and A. Branitskiy, "Framework for mobile internet of things security monitoring based on big data processing and machine learning," *IEEE Access*, vol. 6, pp. 72714–72723, 2018.

[66] U. Sairam and M. V. Bhanu Prakash, "Dl and ml approaches along with blockchain towards iot security," *International Journal of Advanced Science and Technology*, vol. 29, no. 4, pp. 826–832, 2020.

[67] B. Wang, Y. Sun, T. Q. Duong, L. D. Nguyen, and N. Zhao, "Security enhanced content sharing in social IoT: a directed hypergraph-based learning scheme," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4412–4425, 2020.

[68] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2020.

[69] A. Singh, N. Chawla, J. H. Ko et al., "Energy efficient and side-channel secure cryptographic hardware for IoT-edge nodes," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 421–434, 2018.

[70] 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.

[71] L. E. Kane, J. J. Chen, R. Thomas, V. Liu, and M. Mckague, "Security and performance in IoT: a balancing act," *IEEE Access*, vol. 8, pp. 121969–121986, 2020.

[72] A. Alamer, B. Soh, and D. E. Brumbaugh, "Mickey 2.0. 85: a secure and lighter MICKEY 2.0 cipher variant with improved power consumption for smaller devices in the IoT," *Symmetry*, vol. 12, no. 1, 2020.

[73] A. Prathiba and V. S. Kanchana Bhaaskaran, "Hardware footprints of S-box in lightweight symmetric block ciphers for IoT and CPS information security systems," *Integration*, vol. 69, pp. 266–278, 2019.

[74] D. Fang, Y. Qian, and R. Q. Hu, "A flexible and efficient authentication and secure data transmission scheme for IoT applications," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3474–3484, 2020.

[75] Z. Mishra and B. Acharya, "High throughput and low area architectures of secure IoT algorithm for medical image encryption," *Journal of Information Security and Applications*, vol. 53, 2020.

[76] M. Sri Lakshmi and V. Srikanth, "A study on light weight cryptography algorithms for data security in IOT," *International Journal of Engineering & Technology*, vol. 7, no. 2.7, pp. 887–890, 2018.

[77] Q. Xu, Z. Su, M. Dai et al., "APIS: privacy-preserving incentive for sensing task allocation in cloud and edge-cooperation mobile Internet of Things with SDN," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5892–5905, 2019.

[78] K. Janjua, M. A. Shah, A. Almogren et al., "Proactive forensics in IoT: privacy-aware log-preservation architecture in fog-enabled-cloud using holochain and containerization technologies," *Electronics (Switzerland)*, vol. 9, no. 7, pp. 1–39, 2020.

[79] Z. Xu, R. Gu, T. Huang et al., "An IoT-oriented offloading method with privacy preservation for cloudlet-enabled wireless metropolitan area networks," *Sensors (Switzerland)*, vol. 18, no. 9, 2018.

[80] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[81] S. Li, Z. Liu, Z. Huang, H. Lyu, Z. Li, and W. Liu, "DynaPro: dynamic wireless sensor network data protection algorithm in IoT via differential privacy," *IEEE Access*, vol. 7, pp. 167754–167765, 2019.

[82] S. Patil and S. Joshi, "Demystifying user data privacy in the world of IOT," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 4412–4418, 2019.

[83] Y. S. Zhao and H. C. Chao, "A green and secure iot framework for intelligent buildings based on fog computing," *Journal of Internet Technology*, vol. 19, no. 3, pp. 837–843, 2018.

[84] K. Gai, Y. Wu, L. Zhu et al., "Differential privacy-based blockchain for industrial internet-of-things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4156–4165, 2019.

[85] C. Yin, J. Xi, R. Sun et al., "Location privacy protection based on differential privacy strategy for big data in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3628–3636, 2017.

[86] H. Cao, S. Liu, L. Wu et al., "SCRAPPOR: an efficient privacy-preserving algorithm base on sparse coding for information-centric IoT," *IEEE Access*, vol. 6, pp. 63143–63154, 2018.

[87] H. Cao, S. Liu, R. Zhao et al., "IFed: a novel federated learning framework for local differential privacy in Power Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 16, no. 5, 2020.

[88] M. Sun and W. P. Tay, "On the relationship between inference and data privacy in decentralized IoT networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 852–866, 2020.

[89] Y. Ju and H. J. Mun, "The research on security technology for low-performance iot sensor node," *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 3, pp. 594–597, 2018.

[90] A. Yadav, A. Tripathi, N. Rakesh, and S. Pandey, "Protecting composite IoT server by secure secret key exchange for XEN intra virtual machines," *International Journal of Information and Computer Security*, vol. 12, no. 1, pp. 53–69, 2020.

[91] K. Haseeb, A. Almogren, I. U. Din et al., "SASC: secure and authentication-based sensor cloud architecture for intelligent internet of things," *Sensors (Switzerland)*, vol. 20, no. 9, 2020.

[92] M. Juma, A. A. Monem, and K. Shaalan, "Hybrid end-to-end VPN security approach for smart IoT objects," *Journal of Network and Computer Applications*, vol. 158, Article ID 102598, 2020.

[93] J. Choi, Y. In, C. In, S. Seok, H. Seo, and H. Kim, "Secure IoT framework and 2D architecture for End-To-End security," *The Journal of Supercomputing*, vol. 74, no. 8, pp. 3521–3535, 2018.

[94] R. H. Randhawa, A. Hameed, and A. N. Mian, "Energy efficient cross-layer approach for object security of CoAP for IoT devices," *Ad Hoc Networks*, vol. 92, 2019.

[95] J. D. De Hoz Diego, J. Saldana, J. Fernandez-Navajas, and J. Ruiz-Mas, "Decoupling security from applications in CoAP-based IoT devices," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 467–476, 2020.

[96] G. Sagirlar, B. Carminati, and E. Ferrari, "Decentralizing privacy enforcement for Internet of Things smart objects," *Computer Networks*, vol. 143, pp. 112–125, 2018.

[97] M. U. Aftab, Y. Munir, A. Oluwasanmi et al., "A hybrid access control model with dynamic COI for secure localization of satellite and IoT-based vehicles," *IEEE Access*, vol. 8, pp. 24196–24208, 2020.

[98] A. Patwari, P. S. S. Bhavya, and R. K. Maheswari, "NodeMCU and IoT-based safety and security ecosystem for heavy vehicles," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 5, pp. 1482–1490, 2020.

[99] F. Al-Turjman and J. P. Lemayian, "Intelligence, security, and vehicular sensor networks in internet of things (IoT)-enabled smart-cities: an overview," *Computers & Electrical Engineering*, vol. 87, p. 106776, 2020.

[100] N. A. Hussein and M. I. Shujaa, "Secure vehicle to vehicle voice chat based MQTT and coap internet of things protocol," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 1, pp. 526–534, 2020.

[101] S. Kumar and J. Singh, "Internet of vehicles over vanets: smart and secure communication using IoT," *Scalable Computing: Practice and Experience*, vol. 21, no. 3, pp. 425–440, 2020.

[102] V. Sharma, J. Kim, Y. Ko et al., "An optimal security management framework for backhaul-aware 5G-vehicle to everything (V2X)," *Journal of Internet Technology*, vol. 21, no. 1, pp. 245–260, 2020.

[103] S. Belguith, N. Kaaniche, M. Hammoudeh, and T. Dargahi, "Proud: verifiable privacy-preserving outsourced attribute based signcryption supporting access policy update for cloud assisted iot applications," *Future Generation Computer Systems*, vol. 111, pp. 899–918, 2020.

[104] A. Shahzad, K. Zhang, and A. Gherbi, "Intuitive development to examine collaborative iot supply chain system underlying privacy and security levels and perspective powering through proactive blockchain," *Sensors*, vol. 20, no. 13, p. 3760, 2020.

[105] S. A. El-Rahman, D. Aldawsari, M. Aldosari, O. Alrashed, and G. Alsubaie, "A secure cloud based digital signature application for IoT," *International Journal of E-Services and Mobile Applications*, vol. 10, no. 3, pp. 42–60, 2018.

[106] M. A. Mughal, X. Luo, A. Ullah, S. Ullah, and Z. Mahmood, "A lightweight digital signature based security scheme for human-centered Internet of Things," *IEEE Access*, vol. 6, pp. 31630–31643, 2018.

[107] A. Karati, C.-I. Fan, and R.-H. Hsu, "Provably secure and generalized signcryption with public verifiability for secure data transmission between resource-constrained IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10431–10440, 2019.

[108] A. Karati, S. H. Islam, G. Biswas et al., "Provably secure identity-based signcryption scheme for crowdsourced industrial Internet of Things environments," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2904–2914, 2017.

[109] W. Liu, X. Wang, and W. Peng, "Secure remote multi-factor authentication scheme based on chaotic map zero-knowledge proof for crowdsourcing internet of things," *IEEE Access*, vol. 8, pp. 8754–8767, 2020.

[110] A. Yang, C. Zhang, Y. Chen et al., "Security and privacy of smart home systems based on the Internet of Things and stereo matching algorithms," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2521–2530, 2019.

[111] N. M. Sundaram, S. Arunkumar, and S. Kaliappan, "Smart home security monitoring system using IOT," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 2, pp. 256–258, 2018.

[112] J. Ahn, I.-G. Lee, and M. Kim, "Design and implementation of hardware-based remote attestation for a secure internet of things," *Wireless Personal Communications*, vol. 144, pp. 1–33, 2020.

[113] M. Park, H. Oh, and K. Lee, "Security risk measurement for information leakage in IoT-based smart homes from a situational awareness perspective," *Sensors*, vol. 19, no. 9, p. 2148, 2019.

[114] T. Adiono, B. Tandiawan, and S. Fuada, "Device protocol design for security on internet of things based smart home," *International Journal of Online Engineering (iJOE)*, vol. 14, no. 07, pp. 161–170, 2018.

[115] K. Timur, Y. Kim, H. Cho et al., "Conception of smart home perimeter security system based on solar powered IoT solutions," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 2056–2058, 2019.

[116] S. Snigdha and K. Haribabu, "IoT based security system using raspberry PI and mail server," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 11, pp. 1702–1704, 2019.

[117] A. Khanum and R. Shivakumar, "An enhanced security alert system for smart home using IOT," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 1, pp. 27–34, 2019.

[118] P. K. Sharma, J. H. Park, Y.-S. Jeong, and J. H. Park, "Shsec: sdn based secure smart home network architecture for internet of things," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 913–924, 2019.

[119] M. Boussard, D. T. Bui, R. Douville et al., "Future spaces: reinventing the home network for better security and automation in the IoT era," *Sensors (Switzerland)*, vol. 18, no. 9, 2018.

[120] D. Noori, H. Shakeri, and M. N. Torshiz, "Scalable, efficient, and secure RFID with elliptic curve cryptosystem for Internet of Things in healthcare environment," *EURASIP*

*Journal on Information Security*, vol. 2020, no. 1, 11 pages, 2020.

[121] B. A. Alzahrani, A. Irshad, K. Alsubhi et al., "A secure and efficient remote patient-monitoring authentication protocol for cloud-IoT," *International Journal of Communication Systems*, vol. 33, no. 11, 2020.

[122] S. Arunkumar, M. Vetriselvi, and S. Thanalakshmi, "Cryptography based security solutions to IoT enabled health care monitoring system," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. 7, pp. 265–272, 2020.

[123] P. Huang, L. Guo, M. Li, and Y. Fang, "Practical privacy-preserving ECG-based authentication for IoT-based healthcare," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9200–9210, 2019.

[124] R. Hamza, Z. Yan, K. Muhammad, P. Bellavista, and F. Titouna, "A privacy-preserving cryptosystem for IoT E-healthcare," *Information Sciences*, vol. 527, pp. 493–510, 2020.

[125] X. Guo, H. Lin, Y. Wu, and M. Peng, "A new data clustering strategy for enhancing mutual privacy in healthcare IoT systems," *Future Generation Computer Systems*, vol. 113, pp. 407–417, 2020.

[126] A. A. Abd El-Latif, B. Abd-El-Atty, E. M. Abou-Nassar et al., "Controlled alternate quantum walks based privacy preserving healthcare images in Internet of Things," *Optics and Laser Technology*, vol. 124, 2020.

[127] R. Boussada, B. Hamdane, M. E. Elhdhili et al., "Privacy-preserving aware data transmission for IoT-based e-health," *Computer Networks*, vol. 162, 2019.

[128] X. Wang and S. Cai, "Secure healthcare monitoring framework integrating ndn-based IoT with edge cloud," *Future Generation Computer Systems*, vol. 112, 2020.

[129] R. Ding, H. Zhong, J. Ma, X. Liu, and J. Ning, "Lightweight privacy-preserving identity-based verifiable IoT-based health storage system," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8393–8405, 2019.

[130] A. Ullah, G. Said, M. Sher, and H. Ning, "Fog-assisted secure healthcare data aggregation scheme in IoT-enabled WSN," *Peer-to-Peer Networking and Applications*, vol. 13, no. 1, pp. 163–174, 2020.

[131] R. Saha, G. Kumar, M. K. Rai, R. Thomas, and S.-J. Lim, "Privacy Ensured ${e}$ -Healthcare for Fog-Enhanced IoT Based Applications," *IEEE Access*, vol. 7, pp. 44536–44543, 2019.

[132] A. Islam and S. Young Shin, "A blockchain-based secure healthcare scheme with the assistance of unmanned aerial vehicle in Internet of Things," *Computers & Electrical Engineering*, vol. 84, p. 106627, 2020.

[133] Y. Liu and S. Zhang, "Information security and storage of Internet of Things based on block chains," *Future Generation Computer Systems*, vol. 106, pp. 296–303, 2020.

[134] Q. Zhao, S. Chen, Z. Liu, T. Baker, and Y. Zhang, "Blockchain-based privacy-preserving remote data integrity checking scheme for IoT information systems," *Information Processing & Management*, vol. 57, no. 6, Article ID 102355, 2020.

[135] M. Shen, H. Liu, L. Zhu et al., "Blockchain-Assisted secure device authentication for cross-domain industrial IoT," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 942–954, 2020.

[136] A. Gupta, B. Gupta, and K. K. Gola, "Blockchain technology for security and privacy issues in internet of things," *International Journal of Scientific and Technology Research*, vol. 9, no. 3, pp. 377–383, 2020.

[137] M. Zhaofeng, W. Lingyun, W. Xiaochang et al., "Blockchain-enabled decentralized trust management and secure usage control of IoT big data," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4000–4015, 2019.

[138] B. Shala, U. Trick, A. Lehmann, B. Ghita, and S. Shiaeles, "Blockchain and trust for secure, end-user-based and decentralized IoT service provision," *IEEE Access*, vol. 8, pp. 119961–119979, 2020.

[139] J. Chen, "Hybrid blockchain and pseudonymous authentication for secure and trusted IoT networks," *ACM SIGBED Review*, vol. 15, no. 5, pp. 22–28, 2018.

[140] M. Li, D. Hu, C. Lal et al., "Blockchain-enabled secure energy trading with verifiable fairness in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, 2020.

[141] R. M. Mathew, R. Suguna, and M. Shyamala Devi, "Exploration of blockchain for edifying safety and security in IoT based diamond international trade," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 8, pp. 3224–3228, 2019.

[142] C. Ge, Z. Liu, and L. Fang, "A blockchain based decentralized data security mechanism for the internet of things," *Journal of Parallel and Distributed Computing*, vol. 141, 2020.

[143] H. Rui, L. Huan, H. Yang, and Z. YunHao, "Research on secure transmission and storage of energy IoT information based on Blockchain," *Peer-to-Peer Networking and Applications*, vol. 13, no. 4, pp. 1225–1235, 2020.

[144] M. Hussain and U. Jain, "Simple and secure device authentication mechanism for smart environments using Internet of things devices," *International Journal of Communication Systems*, vol. 33, no. 16, Article ID e4570, 2020.

[145] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2018.

[146] K. Fan, Q. Luo, K. Zhang, and Y. Yang, "Cloud-based lightweight secure RFID mutual authentication protocol in IoT," *Information Sciences*, vol. 527, pp. 329–340, 2020.

[147] 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, p. 1962, 2020.

[148] J. Lee, S. Yu, K. Park et al., "Secure three-factor authentication protocol for multi-gateway IoT environments," *Sensors (Switzerland)*, vol. 19, no. 10, 2019.

[149] S. Anandhi, R. Anitha, and V. Sureshkumar, "IoT enabled RFID authentication and secure object tracking system for smart logistics," *Wireless Personal Communications*, vol. 104, no. 2, pp. 543–560, 2019.

[150] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, J. J. P. C. Rodrigues, and Y. Park, "Physically secure lightweight Anonymous user authentication protocol for internet of things using physically unclonable functions," *IEEE Access*, vol. 7, pp. 85627–85644, 2019.

[151] K. M. Renuka, S. Kumari, D. Zhao, and L. Li, "Design of a secure password-based authentication scheme for M2M networks in IoT enabled cyber-physical systems," *IEEE Access*, vol. 7, pp. 51014–51027, 2019.

[152] M. Gheisari, Q.-V. Pham, M. Alazab, X. Zhang, C. Fernandez-Campusano, and G. Srivastava, "ECA: an edge computing architecture for privacy-preserving in IoT-based smart city," *IEEE Access*, vol. 7, pp. 155779–155786, 2019.

[153] M. Gheisari, G. Wang, W. Z. Khan, and C. Fernández-Campusano, "A context-aware privacy-preserving method for IoT-based smart city using software defined networking," *Computers & Security*, vol. 87, p. 101470, 2019.

[154] T. Sasaki, Y. Morita, and T. Kobayashi, "Security requirements and technologies for smart city IoT," *NEC Technical Journal*, vol. 13, no. 1, pp. 54–57, 2018.

[155] S. Gong, E. Tcydenova, J. Jo, Y. Lee, and J. H. Park, "Blockchain-based secure device management framework for an internet of things network in a smart city," *Sustainability*, vol. 11, no. 14, p. 3889, 2019.

[156] C. Toma, A. Alexandru, M. Popa et al., "IoT solution for smart cities' pollution monitoring and the security challenges," *Sensors (Switzerland)*, vol. 19, no. 15, 2019.

[157] C. Badii, P. Bellini, A. Difino, and P. Nesi, "Smart city IoT platform respecting GDPR privacy and security aspects," *IEEE Access*, vol. 8, pp. 23601–23623, 2020.

[158] S. K. Singh, Y. S. Jeong, and J. H. Park, "A deep learning-based IoT-oriented infrastructure for secure smart city," *Sustainable Cities and Society*, vol. 60, 2020.

[159] J. Maruthi Nagendra Prasad, C. V. Lakshmi Narayana, and B. Pandurangaraju, "An extensive study on the applications and security issues of rfid technology in iot," *International Journal of Advanced Science and Technology*, vol. 29, no. 4, pp. 694–707, 2020.

[160] D. Singh, Pushparaj, M. K. Mishra et al., "Security issues in different layers of iot and their possible mitigation," *International Journal of Scientific and Technology Research*, vol. 9, no. 4, pp. 2762–2771, 2020.

[161] S. Kamalakkannan and N. Sivasankari, "Survey on issues in authentication based iot security," *International Journal of Scientific and Technology Research*, vol. 9, no. 2, pp. 1258–1260, 2020.

[162] M. U. Hassan, M. H. Rehmani, and J. Chen, "Privacy preservation in blockchain based IoT systems: integration issues, prospects, challenges, and future research directions," *Future Generation Computer Systems*, vol. 97, pp. 512–529, 2019.

[163] S. P. Maniraj, R. Pranay Sharma, M. Venkata Siva Kumar et al., "Vulnerabilities and security issues in cps and IOT for wire less communication," *International Journal of Recent Technology and Engineering*, vol. 7, no. 5, pp. 164–167, 2019.

[164] D. Kerana Hanirex, K. P. Thooyamani, and A. Muthu Kumaravel, "A study on emerging technology internet of things (IOT): an overview of architecture and security issues," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 6, pp. 1715–1719, 2019.

[165] H. A. Khattak, M. A. Shah, S. Khan, I. Ali, and M. Imran, "Perception layer security in internet of things," *Future Generation Computer Systems*, vol. 100, pp. 144–164, 2019.

[166] N. Wang, P. Wang, A. Alipour-Fanid, L. Jiao, and K. Zeng, "Physical-layer security of 5G wireless networks for IoT: challenges and opportunities," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8169–8181, 2019.

[167] S. Sicari, A. Rizzardi, and A. Coen-Porisini, "5G in the internet of things era: an overview on security and privacy challenges," *Computer Networks*, vol. 179, 2020.

[168] L. Tawalbeh, F. Muheidat, M. Tawalbeh et al., "IoT privacy and security: challenges and solutions," *Applied Sciences (Switzerland)*, vol. 10, no. 12, 2020.

[169] A. Kore and S. Patil, "Internet of things (Iot) enabled wireless sensor networks security challenges and current solutions," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 282–290, 2019.

[170] M. Amoon, T. Altameem, and A. Altameem, "Internet of things sensor assisted security and quality analysis for health care data sets using artificial intelligent based heuristic health management system," *Measurement*, vol. 161, Article ID 107861, 2020.

[171] J. Sun, H. Xiong, X. Liu et al., "Lightweight and privacy-aware fine-grained access control for IoT-oriented smart health," *IEEE Internet of Things Journal*, vol. 7, 2020.

[172] A. Tewari and B. B. Gupta, "An internet-of-things-based security scheme for healthcare environment for robust location privacy," *International Journal of Computational Science and Engineering*, vol. 21, no. 2, pp. 298–303, 2020.

[173] K. U. K. Reddy, S. Shabbiha, and M. R. Kumar, "Design of high security smart health care monitoring system using IoT," *International Journal*, vol. 8, no. 6, 2020.

[174] P. Vijayakumar, M. S. Obaidat, M. Azees et al., "Efficient and secure anonymous authentication with location privacy for IoT-based WBANs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2603–2611, 2019.

[175] J. Mathew and R. Jemima Priyadarsini, "Enhancing security in IoT healthcare services using fog computing," *International Journal of Advanced Science and Technology*, vol. 28, no. 17, pp. 444–450, 2019.

[176] L. Jiang, L. Chen, T. Giannetsos, B. Luo, K. Liang, and J. Han, "Toward practical privacy-preserving processing over encrypted data in IoT: an assistive healthcare use case," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10177–10190, 2019.

[177] J. John, M. S. Varkey, and M. Selvi, "Security attacks in s-wbans on iot based healthcare applications," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 2088–2097, 2019.

[178] W. Tang, J. Ren, K. Deng, and Y. Zhang, "Secure data aggregation of lightweight E-healthcare IoT devices with fair incentives," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8714–8726, 2019.

[179] X. C. Yin, Z. G. Liu, B. Ndibanje et al., "An iot-based anonymous function for security and privacy in healthcare sensor networks," *Sensors (Switzerland)*, vol. 19, no. 14, 2019.

[180] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system," *Information Sciences*, vol. 479, pp. 567–592, 2019.

[181] A. M. Elmisery, S. Rho, and M. Aborizka, "A new computing environment for collective privacy protection from constrained healthcare devices to IoT cloud services," *Cluster Computing*, vol. 22, no. S1, pp. 1611–1638, 2019.

[182] A. Prathiba and V. Bhaaskaran, "Lightweight S-box Architecture for secure internet of things," *Information*, vol. 9, no. 1, p. 13, 2018.

[183] M. Qasaimeh, R. S. Al-Qassas, and S. Tedmori, "Software randomness analysis and evaluation of lightweight ciphers: the prospective for IoT security," *Multimedia Tools and Applications*, vol. 77, no. 14, pp. 18415–18449, 2018.

[184] M. A. F. Al-Husainy and B. Al-Shargabi, "Secure and lightweight encryption model for IoT surveillance camera," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 2, pp. 1840–1847, 2020.

[185] S. S. Dhanda, B. Singh, and P. Jindal, "Lightweight cryptography: a solution to secure IoT," *Wireless Personal Communications*, vol. 112, no. 3, pp. 1947–1980, 2020.

[186] B. Seok, J. C. S. Sicato, T. Erzhena et al., "Secure D2D communication for 5G IoT network based on lightweight

cryptography," *Applied Sciences (Switzerland)*, vol. 10, no. 1, 2020.

[187] A. Ostad-Sharif, H. Arshad, M. Nikooghadam, and D. Abbasinezhad-Mood, "Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme," *Future Generation Computer Systems*, vol. 100, pp. 882–892, 2019.

[188] S. Rajesh, V. Paul, V. G. Menon et al., "A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices," *Symmetry*, vol. 11, no. 2, 2019.

[189] J. Roldán, J. Boubeta-Puig, J. Luis Martínez, and G. Ortiz, "Integrating complex event processing and machine learning: an intelligent architecture for detecting IoT security attacks," *Expert Systems with Applications*, vol. 149, Article ID 113251, 2020.

[190] M. Bagaa, T. Taleb, J. B. Bernabe et al., "A machine learning security framework for iot systems," *IEEE Access*, 2020.

[191] A. Sivanathan, H. Habibi Gharakheili, and V. Sivaraman, "Managing IoT cyber-security using programmable telemetry and machine learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 60–74, 2020.

[192] X. Guo, H. Lin, Z. Li et al., "Deep Reinforcement learning based QoS-aware secure routing for SDN-IoT," *IEEE Internet of things journal*, vol. 7, no. 7, pp. 6242–6251, 2019.

[193] N. K. Kadale and J. R. Prasad, "Overview for security of internet of things using machine learning," *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 349–355, 2020.

[194] X. Zhang, X. Chen, J. K. Liu et al., "DeepPAR and DeepDPA: privacy preserving and asynchronous deep learning for industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2081–2090, 2020.

[195] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[196] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.

[197] A. Ferdowsi and W. Saad, "Deep learning for signal authentication and security in massive internet-of-things systems," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1371–1387, 2019.

[198] F. Liang, W. G. Hatcher, W. Liao, W. Gao, and W. Yu, "Machine learning for security and the internet of things: the good, the bad, and the ugly," *IEEE Access*, vol. 7, pp. 158126–158147, 2019.

[199] F. Ullah, H. Naeem, S. Jabbar et al., "Cyber security threats detection in internet of things using deep learning approach," *IEEE Access*, vol. 7, pp. 124379–124389, 2019.

[200] K. C. Park and D.-H. Shin, "Security assessment framework for IoT service," *Telecommunication Systems*, vol. 64, no. 1, pp. 193–209, 2017.

[201] G. George and S. M. Thampi, "A graph-based security framework for securing industrial IoT networks from vulnerability exploitations," *IEEE Access*, vol. 6, pp. 43586–43601, 2018.

[202] F. I. Salih, N. A. A. Bakar, N. H. Hassan et al., "IOT security risk management model for healthcare industry," *Malaysian Journal of Computer Science*, pp. 131–144, 2019.

[203] M. Aydos, Y. Vural, and A. Tekerek, "Assessing risks and threats with layered approach to Internet of Things security," *Measurement and Control (United Kingdom)*, vol. 52, no. 5-6, pp. 338–353, 2019.

[204] J. R. C. Nurse, S. Creese, and D. De Roure, "Security risk assessment in internet of things systems," *IT Professional*, vol. 19, no. 5, pp. 20–26, 2017.

[205] M. Sohail, R. Ali, M. Kashif et al., "Trustwalker: an efficient trust assessment in vehicular internet of things (viot) with security consideration," *Sensors (Switzerland)*, vol. 20, no. 14, pp. 1–22, 2020.

[206] W. Abbass, Z. Bakraouy, Z. Bakraouy, A. Baina, and M. Bella, "Assessing the internet of things security risks," *Journal of Communications*, vol. 14, no. 10, pp. 958–964, 2019.

[207] H. Yi, ""Systolic inversion algorithms for building cryptographic systems based on security measurement in IoT-based advanced manufacturing," *Journal of the International Measurement Confederation*, vol. 161, 2020.

[208] A. Ghani, K. Mansoor, S. Mehmood, S. A. Chaudhry, A. U. Rahman, and M. Najmus Saqib, "Security and key management in IoT-based wireless sensor networks: an authentication protocol using symmetric key," *International Journal of Communication Systems*, vol. 32, no. 16, Article ID e4139, 2019.

[209] S. Rostampour, M. Safkhani, Y. Bendavid et al., "ECCbAP: a secure ECC-based authentication protocol for IoT edge devices," *Pervasive and Mobile Computing*, vol. 67, Article ID 101194, 2020.

[210] J. Sanchez-Gomez, D. Garcia-Carrillo, R. Marin-Perez et al., "Secure authentication and credential establishment in narrowband IoT and 5G," *Sensors*, vol. 20, no. 3, p. 882, 2020.

[211] H. S. Trivedi and S. J. Patel, "Design of secure authentication protocol for dynamic user addition in distributed Internet-of-Things," *Computer Networks*, vol. 178, 2020.

[212] H. L. Wu, C. C. Chang, and L. S. Chen, "Secure and anonymous authentication scheme for the internet of things with pairing," *Pervasive and Mobile Computing*, vol. 67, 2020.

[213] W. I. Bae and J. Kwak, "Smart card-based secure authentication protocol in multi-server IoT environment," *Multimedia Tools and Applications*, vol. 79, no. 23-24, pp. 15793–15811, 2020.

[214] S. Garg, K. Kaur, G. Kaddoum et al., "Toward secure and provable authentication for internet of things: realizing industry 4.0," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4598–4606, 2020.

[215] P. K. Panda and S. Chattopadhyay, "A secure mutual authentication protocol for IoT environment," *Journal of Reliable Intelligent Environments*, vol. 6, no. 2, pp. 79–94, 2020.

[216] D. Sethia, D. Gupta, and H. Saran, "NFC secure element-based mutual authentication and attestation for IoT access," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 4, pp. 470–479, 2018.

[217] A. A. Alamr, F. Kausar, J. Kim, and C. Seo, "A secure ECC-based RFID mutual authentication protocol for internet of things," *The Journal of Supercomputing*, vol. 74, no. 9, pp. 4281–4294, 2018.

[218] B. L. Parne, S. Gupta, and N. S. Chaudhari, "PSE-AKA: performance and security enhanced authentication key agreement protocol for IoT enabled LTE/LTE-A networks," *Peer-to-Peer Networking and Applications*, vol. 12, no. 5, pp. 1156–1177, 2019.

[219] K. Park, Y. Park, A. K. Das, S. Yu, J. Lee, and Y. Park, "A dynamic privacy-preserving key management protocol for V2G in social internet of things," *IEEE Access*, vol. 7, pp. 76812–76832, 2019.

[220] M. Alshahrani and I. Traore, "Secure mutual authentication and automated access control for IoT smart home using cumulative keyed-hash chain," *Journal of Information Security and Applications*, vol. 45, pp. 156–175, 2019.

[221] B. Bera, D. Chattaraj, and A. K. Das, "Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment," *Computer Communications*, vol. 153, pp. 229–249, 2020.

[222] W.-J. Tsaur and L.-Y. Yeh, "DANS: a secure and efficient driver-abnormal notification scheme with IoT devices over IoV," *IEEE Systems Journal*, vol. 13, no. 2, pp. 1628–1639, 2019.

[223] X. Li, S. Liu, F. Wu et al., "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4755–4763, 2018.

[224] A. Islam and S. Y. Shin, "BUAV: a blockchain based secure UAV-assisted data acquisition scheme in Internet of Things," *Journal of Communications and Networks*, vol. 21, no. 5, pp. 491–502, 2019.

[225] M. I. A. Zahed, I. Ahmad, D. Habibi, and Q. V. Phung, "Green and secure computation offloading for cache-enabled IoT networks," *IEEE Access*, vol. 8, pp. 63840–63855, 2020.

[226] B. Li, T. Chen, and G. B. Giannakis, "Secure mobile edge computing in IoT via collaborative online learning," *IEEE Transactions on Signal Processing*, vol. 67, no. 23, pp. 5922–5935, 2019.

[227] A. Nawaz, J. P. Queralta, J. Guan et al., "Edge computing to secure iot data ownership and trade with the ethereum blockchain," *Sensors (Switzerland)*, vol. 20, no. 14, pp. 1–17, 2020.

[228] W. Wang, P. Xu, D. Liu, L. T. Yang, and Z. Yan, "Lightweighted secure searching over public-key ciphertexts for edge-cloud-assisted industrial IoT devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4221–4230, 2020.

[229] J. Xia, G. Cheng, S. Gu, and D. Guo, "Secure and trust-oriented edge storage for internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4049–4060, 2020.

[230] T. Wang, M. Z. A. Bhuiyan, G. Wang, L. Qi, J. Wu, and T. Hayajneh, "Preserving balance between privacy and data integrity in edge-assisted internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2679–2689, 2020.

[231] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled IoT," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2622–2629, 2020.

[232] P. Zhang, M. Durresi, and A. Durresi, "Multi-access edge computing aided mobility for privacy protection in Internet of Things," *Computing*, vol. 101, no. 7, pp. 729–742, 2019.

[233] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, "A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4831–4843, 2019.

[234] M. Durresi, A. Subashi, A. Durresi, L. Barolli, and K. Uchida, "Secure communication architecture for internet of things using smartphones and multi-access edge computing in environment monitoring," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 4, pp. 1631–1640, 2019.

[235] V. Sharma, I. You, D. N. K. Jayakody, and M. Atiquzzaman, "Cooperative trust relaying and privacy preservation via edge-crowdsourcing in social Internet of Things," *Future Generation Computer Systems*, vol. 92, pp. 758–776, 2019.

[236] D. Wang, B. Bai, K. Lei, W. Zhao, Y. Yang, and Z. Han, "Enhancing information security via physical layer approaches in heterogeneous IoT with multiple access mobile edge computing in smart city," *IEEE Access*, vol. 7, pp. 54508–54521, 2019.

[237] D. E. D. Abou-Tair, S. Büchsenstein, and A. Khalifeh, "A fog computing-based framework for privacy preserving IoT environments," *The International Arab Journal of Information Technology*, vol. 17, no. 3, pp. 306–315, 2020.

[238] S. K. Sood, "Mobile fog based secure cloud-IoT framework for enterprise multimedia security," *Multimedia Tools and Applications*, vol. 79, no. 15-16, pp. 10717–10732, 2020.

[239] K. Fan, H. Xu, L. Gao, H. Li, and Y. Yang, "Efficient and privacy preserving access control scheme for fog-enabled IoT," *Future Generation Computer Systems*, vol. 99, pp. 134–142, 2019.

[240] L. Ferretti, M. Marchetti, and M. Colajanni, "Fog-based secure communications for low-power IoT devices," *ACM Transactions on Internet Technology*, vol. 19, no. 2, 2019.

[241] Y. Yao, Z. Wang, and P. Zhou, "Privacy-preserving and energy efficient task offloading for collaborative mobile computing in IoT: an ADMM approach," *Computers and Security*, vol. 96, 2020.

[242] V. Kiran, S. Rani, and P. Singh, "Towards a light weight routing security in IoT using non-cooperative game models and dempster-shaffer theory," *Wireless Personal Communications*, vol. 110, no. 4, pp. 1729–1749, 2020.

[243] N. Djedjig, D. Tandjaoui, F. Medjek, and I. Romdhani, "Trust-aware and cooperative routing protocol for IoT security," *Journal of Information Security and Applications*, vol. 52, Article ID 102467, 2020.

[244] D. Airehrour, J. A. Gutierrez, S. K. Ray, and "SecTrust-RPL, "SecTrust-RPL: a secure trust-aware RPL routing protocol for Internet of Things," *Future Generation Computer Systems*, vol. 93, pp. 860–876, 2019.

[245] H. Hellaoui, M. Koudil, and A. Bouabdallah, "Energy-efficiency in security of 5G-based IoT: an end-to-end adaptive approach," *IEEE Internet of Things Journal*, vol. 7, 2020.

[246] A. Tandon and P. Srivastava, "Location based secure energy efficient cross layer routing protocols for IOT enabling technologies," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 7, pp. 368–374, 2019.

[247] B. K. Dhaliwal and R. K. Datta, "Secure and energy efficient trust aware routing protocol in IoT using the optimized artificial neural network: SEETA-IoT," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 4341–4353, 2019.

[248] P. Reddy, R. Babu, and R. Babu, "An evolutionary secure energy efficient routing protocol in Internet of Things," *International Journal of Intelligent Engineering and Systems*, vol. 10, no. 3, pp. 337–346, 2017.

[249] A. Anand, M. Conti, P. Kaliyar et al., "TARE: topology Adaptive Re-kEying scheme for secure group communication in IoT networks," *Wireless Networks*, vol. 26, no. 4, pp. 2449–2463, 2020.

[250] A. Arena, P. Perazzo, C. Vallati, G. Dini, and G. Anastasi, "Evaluating and improving the scalability of RPL security in the Internet of Things," *Computer Communications*, vol. 151, pp. 119–132, 2020.

[251] X. Fang, M. Yang, and W. Wu, "Security cost aware data communication in low-power IoT sensors with energy harvesting," *Sensors (Switzerland)*, vol. 18, no. 12, 2018.

[252] J. M. McGinthy and A. J. Michaels, "Secure industrial internet of things critical infrastructure node design," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8021–8037, 2019.

[253] I. Batra, S. Verma, A. Malik et al., "Hybrid logical security framework for privacy preservation in the green internet of things," *Sustainability (Switzerland)*, vol. 12, no. no. 14, 2020.

[254] M. Meenakshi, R. Naresh, and S. Pradeep, "Smart home: security and acuteness in automation of IOT sensors," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 3271–3274, 2019.

[255] H. Lee, "Home IoT resistance: extended privacy and vulnerability perspective," *Telematics and Informatics*, vol. 49, 2020.

[256] S. Bulusu, M. Krosuri, R. Koripella, and N. Sampath, "Smart and secure home automation using internet of things enabling technologies," *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 1, pp. 390–395, 2020.

[257] P. Kumar and L. Chouhan, "A secure authentication scheme for IoT application in smart home," *Peer-To-Peer Networking And Applications*, vol. 14, 2020.

[258] M. Park, H. Oh, and K. Lee, "Security risk measurement for information leakage in IoT-based smart homes from a situational awareness perspective," *Sensors (Switzerland)*, vol. 19, no. 9, 2019.

[259] B.-C. Chifor, I. Bica, V.-V. Patriciu, and F. Pop, "A security authorization scheme for smart home Internet of Things devices," *Future Generation Computer Systems*, vol. 86, pp. 740–749, 2018.

[260] H. H. Qasim, A. E. Hamza, H. H. Ibrahim, H. A. Saeed, and M. I. Hamzah, "Design and implementation home security system and monitoring by using wireless sensor networks WSN/internet of things IOT," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, p. 2617, 2020.

[261] G. Krishna, P. Kumar, K. Ravi et al., "Smart home authentication and security with IoT using face recognition," *International Journal of Recent Technology and Engineering*, vol. 7, pp. 705–709, 2019.

[262] S. Alani, S. N. Mahmood, S. Z. Attaallah et al., "IoT based implemented comparison analysis of two well-known network platforms for smart home automation," *International Journal of Electrical & Computer Engineering*, vol. 111 page, 2011.

[263] A. R. Syafeeza, M. K. Mohd Fitri Alif, Y. Nursyifaa Athirah et al., "IoT based facial recognition door access control home security system using raspberry pi," *International Journal of Power Electronics and Drive Systems*, vol. 11, no. 1, pp. 417–424, 2020.

[264] S. Ravikumar and D. Kavitha, "IoT based home monitoring system with secure data storage by Keccak-Chaotic sequence in cloud server," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[265] P. Gupta and M. Rajoriya, "Face recognition based home security system using IoT," *Journal of Critical Reviews*, vol. 7, no. 10, pp. 1001–1006, 2020.

[266] J. S. P. Peter, S. Selvakumar, H. Pandit et al., "Home automation and home security using arduino and ESP8266(IOT)," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 7, pp. 39–42, 2019.

[267] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7702–7712, 2019.

[268] S. Singh, I.-H. Ra, W. Meng et al., "SH-BlockCC: a secure and efficient Internet of things smart home architecture based on cloud computing and blockchain technology," *International Journal of Distributed Sensor Networks*, vol. 15, no. 4, Article ID 1550147719844159, 2019.

[269] A. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for IoT," *Sensors*, vol. 19, no. 2, p. 326, 2019.

[270] A. S. M. S. Hosen, S. Singh, P. K. Sharma et al., "Blockchain-based transaction validation protocol for a secure distributed IoT network," *IEEE Access*, vol. 8, pp. 117266–117277, 2020.

[271] J. Chi, Y. Li, J. Huang et al., "A secure and efficient data sharing scheme based on blockchain in industrial Internet of Things," *Journal of Network and Computer Applications*, vol. 167, 2020.

[272] A. Shahzad, K. Zhang, and A. Gherbi, "Intuitive development to examine collaborative iot supply chain system underlying privacy and security levels and perspective powering through proactive blockchain," *Sensors (Switzerland)*, vol. 20, no. 13, pp. 1–27, 2020.

[273] M. Sigwart, M. Borkowski, M. Peise et al., "A secure and extensible blockchain-based data provenance framework for the internet of things," *Personal and Ubiquitous Computing*, 2020.

[274] T. A. Alghamdi, I. Ali, N. Javaid et al., "Secure service provisioning scheme for lightweight IoT devices with a fair payment system and an incentive mechanism based on blockchain," *IEEE Access*, vol. 8, 2020.

[275] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, Q. Zhang, and K.-K. R. Choo, "An energy-efficient SDN controller architecture for IoT networks with blockchain-based security," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 625–638, 2020.

[276] B. W. Nyamtiga, J. C. S. Sicato, S. Rathore et al., "Blockchain-based secure storage management with edge computing for IoT," *Electronics (Switzerland)*, vol. 8, no. 8, 2019.

[277] P. Ghadekar, N. Doke, S. Kaneri et al., "Secure access control to IoT devices using blockchain," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 3064–3070, 2019.

[278] A. Muthanna, A. A. Ateya, A. Khakimov et al., "Secure and reliable IoT networks using fog computing with software-defined networking and blockchain," *Journal of Sensor and Actuator Networks*, vol. 8, no. 1, 2019.

[279] J. Ali, T. Ali, S. Musa et al., "Towards secure IoT communication with smart contracts in a Blockchain infrastructure," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, pp. 578–585, 2018.

[280] B. A. Alzahrani, S. A. Chaudhry, A. Barnawi, A. Al-Barakati, and M. H. Alsharif, "A privacy preserving authentication scheme for roaming in IoT-based wireless mobile networks," *Symmetry*, vol. 12, no. 2, p. 287, 2020.

[281] X. Wang, M. Umehira, B. Han, H. Zhou, P. Li, and C. Wu, "An efficient privacy preserving spectrum sharing framework for internet of things," *IEEE Access*, vol. 8, pp. 34675–34685, 2020.

[282] A. R. Sfar, Y. Challal, P. Moyal et al., "A game theoretic approach for privacy preserving model in IoT-based transportation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4405–4414, 2019.

[283] R. S. Apare and S. N. Gujar, "Implementing adaptive dragonfly optimization for privacy preservation in IoT,"

*Journal of High Speed Networks*, vol. 25, no. 4, pp. 331–348, 2019.

[284] Q. Sun, M. C. Willemsen, and B. P. Knijnenburg, "Unpacking the intention-behavior gap in privacy decision making for the internet of things (IoT) using aspect listing," *Computers & Security*, vol. 97, p. 101924, 2020.

[285] P. Emami Naeini, M. Degeling, L. Bauer et al., "The influence of friends and experts on privacy decision making in iot scenarios," *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, pp. 1–26, 2018.

[286] H. Oh, S. Park, G. M. Lee, J. K. Choi, and S. Noh, "Competitive data trading model with privacy valuation for multiple stakeholders in IoT data markets," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3623–3639, 2020.

[287] P. Menard and G. J. Bott, "Analyzing IoT users' mobile device privacy concerns: extracting privacy permissions using a disclosure experiment," *Computers & Security*, vol. 95, Article ID 101856, 2020.

[288] A. A. A. El-Latif, B. Abd-El-Atty, S. E. Venegas-Andraca et al., "Providing end-to-end security using quantum walks in IoT networks," *IEEE Access*, vol. 8, pp. 92687–92696, 2020.

[289] B. Mukherjee, S. Wang, W. Lu et al., "Flexible IoT security middleware for end-to-end cloud-fog communication," *Future Generation Computer Systems*, vol. 87, pp. 688–703, 2018.

[290] C. M. Latha and K. L. S. Soujanya, "Enhancing end-to-end device security of internet of things using dynamic cryptographic algorithm," *International Journal of Civil Engineering and Technology*, vol. 9, no. 9, pp. 408–415, 2018.

[291] S. Raza, T. Helgason, P. Papadimitratos, and T. Voigt, "SecureSense: end-to-end secure communication architecture for the cloud-connected Internet of Things," *Future Generation Computer Systems*, vol. 77, pp. 40–51, 2017.

[292] C.-S. Park and W.-S. Park, "A group-oriented DTLS handshake for secure IoT applications," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1920–1929, 2018.

[293] S. Pérez, J. L. Hernández-Ramos, S. Raza et al., "Application layer key establishment for end-to-end security in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2117–2128, 2019.

[294] S. Pérez, D. Garcia-Carrillo, R. Marín-López, J. L. Hernández-Ramos, R. Marín-Pérez, and A. F. Skarmeta, "Architecture of security association establishment based on bootstrapping technologies for enabling secure IoT infrastructures," *Future Generation Computer Systems*, vol. 95, pp. 570–585, 2019.

[295] D. V. Medhane, A. K. Sangaiah, M. S. Hossain, G. Muhammad, and J. Wang, "Blockchain-enabled distributed security framework for next-generation IoT: an edge cloud and software-defined network-integrated approach," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6143–6149, 2020.

[296] A. Jindal, G. S. Aujla, and N. Kumar, "SURVIVOR: a blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment," *Computer Networks*, vol. 153, pp. 36–48, 2019.

[297] A. H. Sodhro, S. Pirbhulal, L. Zongwei, K. Muhammad, and N. Zahid, "Towards blockchain-enabled security technique for industrial Internet of Things based decentralized applications," *Journal of Grid Computing*, vol. 18, pp. 615–628, 2020.

[298] S. Rathore, B. Wook Kwon, and J. H. Park, "BlockSecIoTNet: blockchain-based decentralized security architecture for IoT network," *Journal of Network and Computer Applications*, vol. 143, pp. 167–177, 2019.

[299] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "A trustworthy privacy preserving framework for machine learning in industrial IoT systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6092–6102, 2020.

[300] S. Rathore, Y. Pan, J. H. Park, and " BlockDeepNet, "A blockchain-based secure deep learning for IoT network," *Sustainability (Switzerland)*, vol. 1114 pages, 2019.

[301] S. S. S. Sugi and S. R. Ratna, "A novel distributed training on fog node in IoT backbone networks for security," *Soft Computing*, vol. 24, no. 24, pp. 18399–18410, 2020.

[302] L. Xie, Y. Ding, H. Yang, and X. Wang, "Blockchain-based secure and trustworthy internet of things in SDN-enabled 5G-VANETs," *IEEE Access*, vol. 7, pp. 56656–56666, 2019.

[303] P. Sudhakaran and M. C, "Energy efficient distributed lightweight authentication and encryption technique for IoT security," *International Journal of Communication Systems*, p. e4198, 2019.

Hindawi
Mobile Information Systems
Volume 2020, Article ID 8889192, 19 pages
https://doi.org/10.1155/2020/8889192

*Research Article*

# $A^2$ Chain: A Blockchain-Based Decentralized Authentication Scheme for 5G-Enabled IoT

**Xudong Jia,[1] Ning Hu,[1,2] Shi Yin,[1] Yan Zhao,[1] Chi Zhang,[1] and Xinda Cheng[1]**

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China*
[2]*Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518000, China*

Correspondence should be addressed to Ning Hu; huning@gzhu.edu.cn

Received 13 August 2020; Revised 7 October 2020; Accepted 25 October 2020; Published 21 December 2020

Academic Editor: Vishal Sharma



The fifth-generation mobile communication technology (5G) provides high-bandwidth and low-latency data channels for massive IoT terminals to access the core business network. At the same time, it also brings higher security threats and challenges. Terminal identity authentication is an important security mechanism to ensure the core business network; however, most of the existing solutions adopt a centralized authentication model. Once the number of authentication requests exceeds the processing capacity of the authentication center service, it will cause authentication request congestion or deadlock. The decentralized authentication model can effectively solve the above problems. This article proposes a decentralized IoT authentication scheme called $A^2$ Chain. First, $A^2$ Chain uses edge computing to decentralize the processing of authentication requests and eliminate the burden on authentication services and the network. Second, to implement cross-domain identity verification of IoT devices, $A^2$ Chain uses blockchain, and sidechain technologies are used to securely share the identity verification information of IoT devices. Additionally, $A^2$ Chain replaces public key infrastructure (PKI) algorithm with identity-based cryptography (IBC) algorithm to eliminate the management overhead caused by centralized authentication model.

## 1. Introduction

With the rapid development of 5G networks, their fast speed, low latency, and high access will provide a broader platform for the development of IoT technology [1–3]. As defined by 3GPP, 5G supports access to at least $10^6$ devices per square kilometre [4]. As 5G powers IoT, it also brings huge challenges to IoT security [5]. The authentication of IoT devices is an important step in ensuring IoT security.

Traditional authentication schemes are usually centralized, which has high latency and untimely response problems in the 5G mass IoT device access scenario [6]. On the one hand, the authentication server or network node will have serious network congestion when massive IoT devices ask for authentication in this era where IoT devices are ubiquitous, and this will seriously affect the service quality of IoT applications [7–11]. On the other hand, centralized authentication usually requires the authentication center to respond to the authentication request of the IoT device. However, due to the long link distance, it cannot satisfy the delay-sensitive applications (for example, the internet of vehicles and unmanned aerial drones) [12]. Also, traditional centralized authentication uses a public key infrastructure-based authentication structure, which carries high computational and communication costs for IoT devices that have limited resources in terms of power, memory, and processing power [13, 14]. Secondly, in traditional public key infrastructure- (PKI-) based authentication models, there is a single point of failure and third-party trustworthiness issues [15].

Decentralized IoT authentication can meet the authentication needs of a large number of IoT devices, and authentication latency issues can be solved by authenticating the device identity through edge nodes. In decentralized authentication scheme of IoT, the decentralized security mechanism is necessary to protect network resources or data, especially to ensure the consistency of authentication data of edge authentication services. In recent years, blockchain

technology has gained widespread attention in authentication and access control research due to its decentralized and cryptographic properties. There is a natural fit between blockchain technology as a distributed ledger and the decentralized edge computing model, and researchers have already adopted edge computing to support services in blockchain networks [16–18]. Besides, blockchain's non-falsifiability and fault tolerance make it a good solution to authentication problems [19]. For example, in [20], the feasibility of using blockchain technology for IoT device authentication in edge computing systems is discussed, and blockchain-based smart contracts are introduced to handle the operation of authentication-related certificates; Jia et al. [21] proposed a blockchain-based cross-domain authentication system applied to the authentication process for data access to different IoT application domains; The study [22] was based on blockchain and elliptic curve cryptosystem cross-data center authentication and key exchange programs.

However, in existing blockchain-based authentication schemes, the authentication information of a large number of IoT devices is stored in a single blockchain, which poses a huge storage burden and scalability problem for blockchain nodes. In this paper, we propose a decentralized IoT authentication scheme combining edge computing and sidechain techniques. We named it $A^2$ Chain since the proposed authentication model includes two types of blockchains: application domain blockchain and alliance blockchain. Compared to past work, the innovations and contributions in this paper are as follows:

(1) Edge computing technology is adopted to decentralize the processing of authentication requests through the authentication service nodes deployed at the edge of the network. On the one hand, it reduces the server burden caused by a large number of IoT authentication requests. On the other hand, the authentication service processing is close to the terminal, which can reduce the network burden and authentication delay and improve communication efficiency.

(2) An IoT authentication structure based on application domain-alliance chains is proposed to deploy blockchains in different application domains as well as between application domains, respectively. The application domain blockchain acts as a sidechain for the alliance blockchain. Each application domain blockchain can run the intradomain authentication process independently within the application domain. The federation chain stores the authentication information index of the application domain devices and proves the existence of the authentication information by simplified payment verification (SPV) proof when cross-domain authentication is required. This structure occupies less storage space and improves the efficiency of searching for target information.

(3) Identity-based cryptography (IBC) is proposed for identity authentication without introducing any trusted third parties. In this case, public key certificates are no longer required, reducing the heavy workload of issuing, maintaining, and revoking digital certificates.

The remainder of this paper is organized as follows: an overview of the related work is given in Section 2. In Section 3, the problem is further stated. Section 4 provides an overview of the proposed solution, Section 5 details the certification process of the solution, and Section 6 verifies the effectiveness and efficiency of the solution through experiments. In Section 7, we discuss the shortcomings of the program and look forward to future research directions. Finally, in Section 8, we conclude the paper.

## 2. Related Work

*2.1. Authentication Scheme Based on Traditional Scheme.* At present, IoT usually adopts centralized authentication, which is more costly, prone to a single point of failure, and less efficient in the case of mass device authentication. [23]. Esfahani et al. [24] proposed a lightweight industrial Internet of Things (IIoT) device authentication mechanism, but it stores the authentication data on a local server. Therefore, it is susceptible to a single point of failure. In order to meet the authentication requirements of many IoT devices in the 5G network environment, Ni et al. [25] used fog computing and network slicing technology to propose an efficient and secure service-oriented authentication framework. Users can use the fog node to select the appropriate network slice according to the service type of the access service and efficiently establish a connection with the core network. Gross et al. [26] introduced an authentication method based on IPsec and TLS. However, the higher computational cost required by Gross' scheme is intolerable for resource-constrained IoT devices. Lai et al. [27] proposed the CPAL scheme in order to enable IoT devices to access the mobile Internet all the time. In CPAL, secure roaming authentication can be provided for IoT devices through group signature technology.

*2.2. Authentication Scheme Based on Blockchain.* Blockchain will play an important role in IoT device management and security due to its characteristics such as decentralized and untamperable. In [28], Hammi et al. discussed the current dilemma in IoT authentication and propose bubbles of trust, a decentralized authentication scheme based on blockchain, to create a secure virtual area in the blockchain, which enables IoT devices to communicate securely. However, due to the closed nature of its virtual region, IoT devices can only communicate with devices belonging to the same region and cannot communicate across domains. In [29], Bao et al. proposed the IoT Chain scheme, which consists of an authentication layer, a blockchain layer, and an application layer to achieve authentication, access control, privacy protection, lightweight features, regional node fault tolerance, denial-of-service resilience, and storage integrity. Khalid et al. [30] proposed

a lightweight decentralized IoT based on the fog computing and blockchain technology Internet authentication scheme, IoT devices will be tied to the IoT application system where they are located when they register, and the latter will issue tokens for them, thus enabling secure communication between devices. Zhang et al. [31] proposed a blockchain-based decentralized vehicle authentication scheme and designed collaborative authentication based on secret sharing and blockchain-based data tracking and trust management in a dynamic agent edge computing model. Cui et al. [32], in his study, a hybrid blockchain-based multi-WSN authentication scheme, designed a wireless-aware network hierarchical model and a hybrid blockchain model combining private and public blockchains. For different types of devices, different blockchains were used for authentication. The authentication information of all nodes was stored on the public blockchain at that time, which caused a certain storage burden.

## 3. Proposed Scheme

In this part, we design an application domain-alliance chain IoT authentication model called $A^2$ Chain to meet the need for secure authentication in IoT. Firstly, the problem presented in this paper is stated; secondly, reasonable assumptions are made about the scheme; and finally, based on the above assumptions, our proposed authentication scheme is presented.

*3.1. Motivation and Basic Idea.* Authentication is one of the indispensable means to ensure the security of network communication. 5G enables IoT to have higher transmission speed and capacity and lower transmission delay and can provide high coverage and massive device deployment for the Internet of Things applications [2, 3]. These massively connected terminal devices simultaneously initiate authentication requests, which will have a serious impact on the authentication server [9–11]. In traditional authentication mechanisms, a centralized mechanism is usually used, as in Figure 1(a), where all devices are authenticated through a centrally located authentication server. In the case of massive device access, centralized authentication will bring about a challenge to the availability of legitimate devices, or a weak link in resource exhaustion attacks [15, 33].

As a decentralized and distributed technology, blockchain provides a new solution to the problems that exist in IoT authentication [28, 34, 35]. As in Figure 1(b), blockchain-based authentication schemes decentralize IoT authentication by establishing a blockchain network at a gateway or authentication server in the system to achieve distributed management of the authentication process. These solutions work well to overcome the single point of failure of centralized authentication, third-party trust, and the difficulty of resisting DoS attacks. However, there are still some limitations and challenges of existing blockchain-based authentication schemes as follows:

(1) Low authentication efficiency: the authentication process for IoT applications requires an authentication server to handle authentication requests, although blockchain-based authentication schemes enable distributed management of the authentication process and no longer rely on a single centralized authentication center. However, authentication servers are usually deployed on the side away from the end device or on cloud servers. This imposes higher latency and bandwidth consumption on the authentication process [36]. And when in IoT applications, latency-sensitive applications have strict requirements on response time. In addition, as the number of IoT devices increases, the burden on the authentication server increases significantly, which will also bring bottlenecks and delays in the system communication, thus limiting the quality of the system service.

(2) Scalability problem: with the popularity of IoT, the identity authentication problem in IoT does not only exist in a single IoT application, but also in different IoT applications with the same authentication needs [21, 37], which we call cross-domain authentication. Blockchain-based authentication schemes are usually deployed in a single application domain or intelligent system, and authentication information from different application domains or systems is not interoperable, lacking an effective cross-domain authentication scheme.

(3) Storage overload: even though some schemes solve the cross-domain authentication problem to some extent by forming federated blockchains [38–40], due to the nature of the blockchain, the full node of the blockchain must store every block on the blockchain and the transactions it contains. Therefore, each authentication server has to store all registered IoT endpoint authentication information, which includes not only authentication information from this application, but also information from other application domains. The information from other application domains may include a large number of devices that do not require cross-domain authentication, resulting in a waste of storage space. All device information is stored in the federated blockchain, and frequent authentication operations consume a lot of resources and time, which does not meet the real-time requirements of IoT.

To overcome the aforementioned shortcomings of blockchain-based authentication, we propose to combine blockchain-sidechain technology as well as edge computing technology to authenticate IoT devices. In the decentralized authentication scheme proposed in this paper, the basic ideas include the following aspects. First, in the organization of the authentication architecture, we propose an authentication architecture based on edge computing to deploy authentication service nodes at the edge of the network, which we call edge authentication nodes (EAs). Since the authentication service is closer to the end device side, authentication requests from a large number of endpoints do

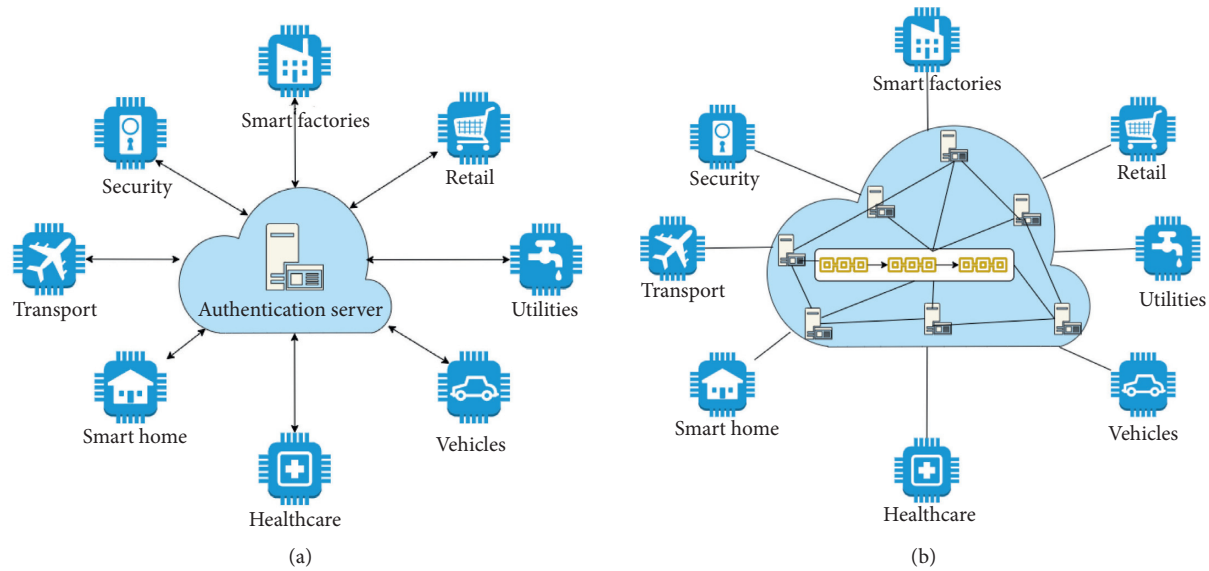



Figure 1: (a) Centralized and (b) decentralized authentication models.

not need to be sent to the core network, which can effectively reduce authentication latency and network burden [41, 42]. Second, we use blockchain and sidechain technology to build different application domain blockchains and alliance blockchains to share authentication information, instead of the trusted third-party authorization process. On the one hand, different application domains can ensure the independence of their own applications in different sidechains; on the other hand, sidechain technology provides a secure decentralized peer-to-peer data-sharing platform, and each application domain does not need to store unnecessary authentication information, which reduces the storage burden of blockchain nodes and improves the scalability of the authentication model [43, 44]. Finally, in terms of the signature algorithm, we propose to adopt an identity-based signature algorithm [45], which can determine the authenticity of the user without a trusted third party and reduce the overhead of storing certificates on the end device.

When a user's device wants to access the network of the application domain to which it belongs, it can initiate an authentication request to the nearest edge authentication server to quickly pass identity authentication. When the device wants to access the network or data of other application domains, it can use sidechain technology to prove the reliability of its authentication information through SPV to achieve cross-domain authentication. In the following chapters, we will describe our plan in detail.

*3.2. Assumptions.* We propose an IoT authentication model scheme based on an application domain-alliance blockchain based on several reasonable assumptions that can be satisfied under certain conditions. The assumptions are as follows:

(1) Each IoT device has a unique object identifier [21], the object identifier (OID) structure is <Domain_ID. Category_ ID. Entity_ID>, and Table 1 describes the meaning of each field

(2) All domain management nodes and edge authentication nodes are legitimate and trusted

(3) The system initialization and key distribution process is secure

*3.3. System Architecture.* According to different node functions, IoT nodes can be divided into domain management nodes, edge authentication nodes, and terminal devices. In order to facilitate the management of IoT devices and achieve their secure authentication, the system architecture is designed as shown in Figure 2 according to the different terminal device functions or usage scenarios. The entire architecture is divided into multiple application domains, and each network includes domain management nodes, edge authentication nodes, and end devices:

(1) Domain manage node (DM): the main function is to manage the nodes in the application domain. As a node manager, it is trusted by the nodes in the network, and the terminal devices in the application domain need to register with the domain manage node before entering the network, and the domain manage node generates the private key for them according to the identity information provided and returns to the terminal devices.

(2) Edge authentication node (EA): edge authentication node is used to authenticate the identity of end devices and has strong computational and storage capabilities. Edge authentication nodes are deployed decentralized at the edge of the network, near the end device side, and have low latency, which reduces the load on authentication services and the network [12]. Through distributed edge authentication nodes, we have realized the decentralization of the authentication structure.

Table 1: OID description.

| Field | Mandatory (M)/Option (O) | Interpretation |
|---|---|---|
| Domain ID | M | Registered domain ID |
| Category ID | O | Categories of entities in the security domain, such as devices and servers |
| Entity ID | M | The unique number assigned to the entity |

(3) Terminal device (TD): This consists mainly of a large number of IoT terminal devices deployed in various application scenarios for sensing, serving, and communicating. These devices can detect or generate data for transmission to different IoT applications. Authentication is required to access the network or to access data.

*3.3.1. Types of Authentication.* The various nodes in the network collaborate with each other to accomplish various IoT application tasks. When a terminal device accesses the network or needs to access data, it needs to be authenticated, and in our proposed scenario, two types of authentication scenarios are involved:

(A) Intradomain authentication: due to business requirements, the terminal device needs access to its registered application domain data, as shown in Request 1 in Figure 2. In this case, the edge authentication node can easily retrieve the public parameters of the terminal signature through the application domain blockchain to authenticate the identity of the terminal device. Such authentication type we call intradomain authentication.

(B) Cross-domain authentication: with the rapid development of IoT, the number of application domains is increasing rapidly, and the interaction between different application domains is becoming more frequent. In some cases, devices from different application domains need to collaborate to complete a task, and terminal devices need to access application domain data outside their registered application domain, such as request 2 in Figure 2. Unlike intradomain authentication scenarios, application domains do not necessarily trust each other, as a domain is usually reluctant to let others access its sensitive data. In addition, the edge authentication node and the terminal device belong to different application domains, and the public parameters of the system signature are also different; in order to achieve the secure transmission of data from different application domains and terminal communication, cross-domain authentication of IoT terminals needs to be implemented, and the detailed process is described in Section 4.

*3.3.2. $A^2$ Chain Model.* In the past blockchain-based authentication schemes, IoT nodes join the same blockchain network, but a large number of IoT nodes frequently undergoing authentication operations will bring a lot of resources and time consumption, cannot meet the real-time requirements of IoT devices [29]; at the same time, different application domains join the same blockchain, application domain authentication devices not only need to store the authentication information in this domain, but also need to store the authentication information of other application domains, greatly increasing the storage pressure of the authentication device and information search space, will also affect the efficiency of the authentication service.

To this end, in this paper, we propose an $A^2$ Chain authentication model, which consists of two main parts: the application domain blockchain and the alliance blockchain.

(1) Alliance blockchain: all domain management nodes are connected to the alliance blockchain as nodes of the alliance blockchain. The alliance blockchain is connected to multiple application domain blockchains via domain management nodes for secure management and sharing of authentication information between different application domains.

(2) Application domain blockchain: application domain blockchain consists of domain management nodes and edge authentication nodes according to the application domain and location, which avoids the consumption of computation and storage irrelevant transactions and reduces the delay caused by transmission. The application domain blockchain is used to store the authentication information of end devices within the domain.

In addition, through the noncentralized nature of the blockchain, $A^2$ Chain does not require a trusted third-party entity and achieves a good decentralized authentication.

*3.4. Signature Algorithm.* In the authentication process, the proposed system uses an identity-based cryptosystem [45]. Since there is no need to use public key certificates in identity-based cryptographic systems and no trusted third-party entities to issue certificates, it satisfies the need for decentralized authentication. Moreover, the use of identity-based signature algorithms can reduce the complexity of deployment and management and has unique advantages in protecting IoT applications.

We use the identity-based cryptographic standard SM9 [46] issued by the State Cryptography Administration for authentication, and the strength of SM9's encryption is equivalent to the RSA encryption algorithm for 3072-bit keys.

The SM9 signature algorithm consists of five steps: system parameter generation, master key generation, device key generation, signature generation, and signature verification. The signer holds an identity and a corresponding private key, which is generated by the key generation server

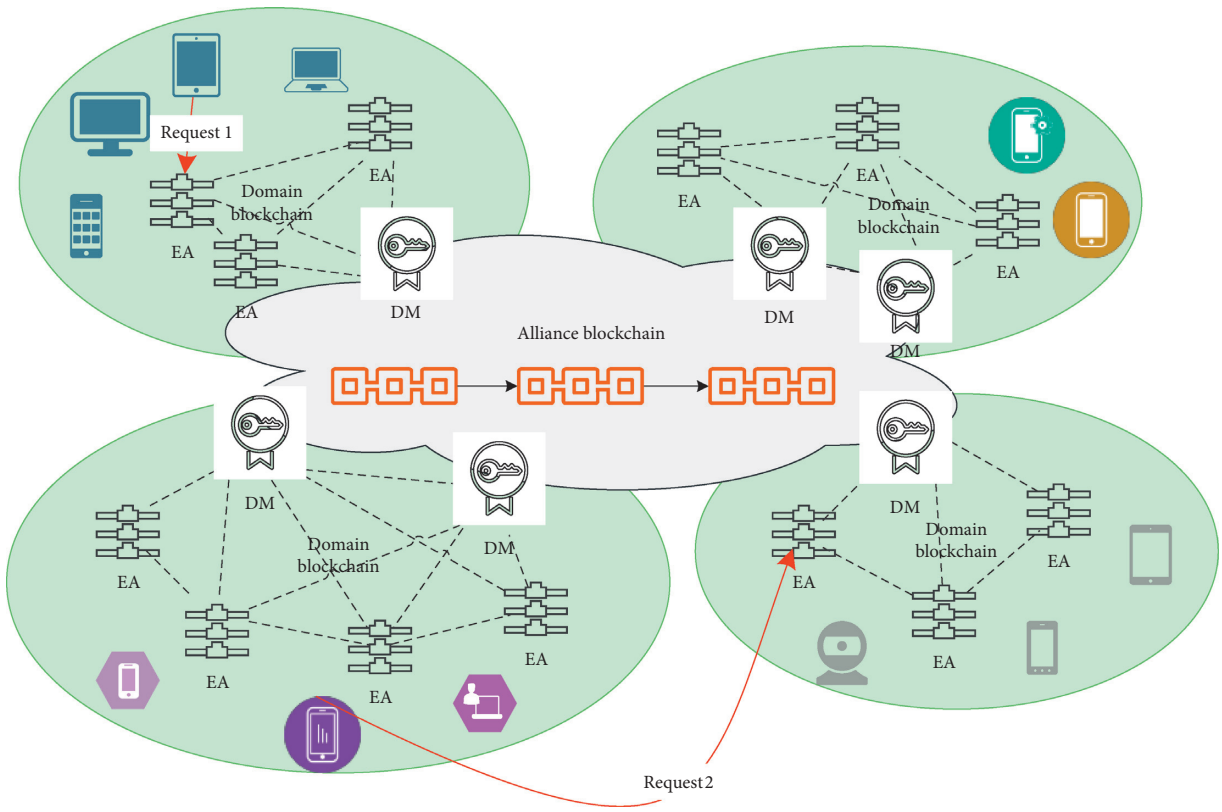



Figure 2: System architecture of A$^2$ Chain.

(KGS) through the combination of the master private key and the signer's identity. The signer uses its own private key to generate a digital signature on the data, and the verifier uses the signer's identity to generate its public key to verify the reliability of the signature, i.e., to verify the authenticity and integrity of the sent data and the identity of the data sender:

System parameter generation (SPG): it includes the curve identifier cid; the parameters $q$ of the base field $F_q$ of the elliptic curve; the parameters $a$ and $b$ of the elliptic curve equation; the prime factor $N$ of the curve order and the residual factor $cf$ relative to $N$; the embedding degree $k$ of the curve $E(F_q)$ with respect to $N$; the generator $P_1$ of the $N$ order cyclic subgroup $G_1$ of $E(F_{q^{d_1}})$, where $d_1$ divides $k$; the generator $P_2$ of the $N$-order cyclic subgroup $G_2$ of $E(F_{q^{d_2}})$, where $d_2$ divides $k$; identifier eid for bilinear pair $e$, bilinear pair $e$: $G_1 \times G_2 \longrightarrow G_T$, the order of $G_T$ is $N$; optionally, the homomorphic mapping $\Psi$ from $G_2$ to $G_1$.

System master key generation (MKGen): the KGS server generates a random number $s \in [1, N-1]$ as the system's master private key and computes the element $P_{\text{pub}} = [s]P_2$ in $G_2$ as the system's master public key pair $(s, P_{\text{pub}})$.

Device signature key generation (DKGen): KGS selects and exposes a byte to represent the private key generation function identifier hid. Assuming the device's identity is ID, to generate the device's private key $d$, KGS first computes $t_1 = H_1(\text{ID}\|\text{hid}, N) + s$ on the finite domain $F_N$. If $t_1 = 0$, then it is necessary to regenerate the master private key, compute and expose the master public key, and update the existing device's private key; otherwise compute $t_2 = s \cdot t_1^{-1}$, and then calculate $d = [t_2]P_1$. $d$ is the user's signed private key.

Signature generation (SigGen): assuming the message to be signed is a bit string $M$, perform the algorithmic steps given in Algorithm 1 as a signature device in order to obtain the digital signature $(h, S)$ of the message $M$.

Signature verification (SigVer): in order to verify the received message M and its digital signature (h, S), the verifier performs the following arithmetic steps:

## 4. Authentication Mechanism

In this section, we will present the working of the proposed A$^2$ Chain. The system consists of three main phases: the initialization phase, the registration phase, and the device authentication phase.

*4.1. Initialization Phase.* In the initialization phase, each domain management node uses the SPG algorithm to generate public parameter group *parameters* and calls MKGen and DKGen to generate master key pair $(s, P_{pub})$ and asymmetric key pairs $DM_{SK}, DM_{ID}$ of the domain management node, in which $DM_{SK}$ demonstrates the signed private key of the domain management node and $DM_{ID}$ represents the identity of the domain management node and the public key corresponding to $DM_{SK}$.

**Input**: message $M$, $P_{pub}$, and *parameters*m private key $d$
**Output**: signature $(h, S)$
(1) Compute the element $g = e(P_1, P_{\text{pub}})$ in group $G_T$;
(2) Generate a random number $r \in [1, N-1]$;
(3) Compute the element $= g^r$ in the group $G_T$, converting the data type of $\omega$ to a bit string;
(4) Compute the integer $h = H_2(M \| \omega, N)$;
(5) Compute the integer $L = (r - h) \bmod N$; if $L = 0$, then return 2;
(6) Compute the element $S = [L]d$ in group $G_1$;
(7) Convert the data types $h$ and $S$ to byte strings and the signature of the message $M$ is $(h, S)$.

Algorithm 1: Signature generation (SigGen) algorithm.

**Input**: message $M$, $P_{\text{pub}}$, *parameters*, *ID*, and signature $(h, S)$
**Output**: verification result—succeed or fail.
(1) Convert the data type of $h$ to an integer, check whether $h \in [1, N-1]$ holds, and if it does not, the verification fails;
(2) Convert the data type of $S$ to a point on an elliptic curve, and check whether $S \in G_1$ holds, and if not, the verification fails;
(3) Compute the element $g = e(P_1, P_{\text{pub}})$ in the group $G_T$;
(4) Compute the element $t = g^h$ in the group $G_T$;
(5) Compute the integer $h_1 = H_1(\text{ID} \| \text{hid}, N)$;
(6) Compute the element $P = [h_1]P_2 + P_{\text{pub}}$ in the group $G_2$;
(7) Compute the element $u = e(S, P)$ in the group $G_T$;
(8) Compute the element $\omega = u \cdot t$ in the group $G_T$, converting the data type of $\omega$ to a bit string;
(9) Compute the integer $h_2 = H_2(M \| \omega, N)$, check whether $h_2 = h$ is valid, if so the verification passes; otherwise the verification fails.

Algorithm 2: Signature verification (SigVer) algorithm.

After generating public parameters and keys, the domain management node broadcasts them in the application domain; creates blocks with the identity ID $DM_{ID}$, the public parameter group *parameters*, and the master public key *P_pub*; writes them into the application domain blockchain; and stores their block numbers in the alliance blockchain. The relevant steps are as follows:

(1) First, the domain management node generates the public parameter group *parameters*, the master key pair $(s, P_{pub})$, and the asymmetric key pair $DM_{SK}, DM_{ID}$
(2) The domain management node creates a transaction $T_1 = (D_{ID}, DM_{ID}, parameters, P_{pub})$ in the application domain blockchain and checks whether there is a $DM_{ID}$in the blockchain to verify the transaction, where $D_{\text{ID}}$ indicates the application domain ID
(3) If $DM_{ID}$ already exists in the application domain blockchain, the transaction validation fails and an error notification is returned to the domain management node
(4) If the $DM_{ID}$ does not exist in the application domain blockchain, the transaction will be allowed and a new block will be created for it
(5) When the domain management node initialization is successful, the domain management node $DM_{ID}$, domain ID, and its block number *Block_num* are uploaded to the alliance blockchain to form a reference record of the authentication information sharing process
(6) The domain management node calls the DKGen algorithm to generate the device's signature private key $TD_{SK}$ according to the device's identification $TD_{ID}$, and the domain management node calls the SigGen algorithm to sign the $TD_{SK}$ and generate the signature $Sig(TD_{SK})$ to send the device's signature private key $TD_{SK}$ with the signature $Sig(TD_{SK})$ to the device $TD_{ID}$in a secure manner

*4.2. Registration Phase.* During this phase, the IoT device will register with a domain management node within its application domain, and upon successful registration, the device will be associated with that application domain and the associated information will be counted in the alliance blockchain.

The main steps in the registration phase are as follows:

(1) The device sends the registration request message $M_1 = TD_{ID} \| DM_{ID} \| \text{Request}_{\text{Reg}} \| LT \| TS \| SigGen(TD_{ID} \| DM_{ID} \| \text{Request}_{\text{Reg}} \| LT \| TS)_{TD_{SK}}$ to the domain management node; $M_1$ includes the identifier $TD_{ID}$, the domain management node $DM_{ID}$, the registration request $\text{Reqest}_{\text{Reg}}$, life time $(LT)$, timestamp $(TS)$, and the corresponding signature

(2) Upon receipt of the message $M_1$, the domain management node verifies that the received $DM_{ID}$ is in this application domain and that the $TD_{ID}$ has not been registered

(3) If the $DM_{ID}$ does not belong to the application domain or a device with a $TD_{ID}$ that has been registered, registration will not be allowed and the registration process will be terminated

(4) If the $DM_{ID}$ belongs to the application domain and the $TD_{ID}$ is not registered, registration is allowed

(5) Call the SigVer algorithm to verify the validity of the message by verifying the signature. If the validation is valid, the registration process continues; otherwise the registration process is aborted

(6) The domain management node creates a transaction $T_2 = TD_{ID} \| DM_{ID} \| LT \| SigGen(U_{ID} \| DM_{ID} \| LT)_{DM_{SK}}$ in the application domain blockchain, and the block structure is shown in Figure 3

(7) When the new device registration is successful, the device $TD_{ID}$ and its block number *Block_num* will be uploaded to the alliance blockchain along with the corresponding domain management node $DM_{ID}$ to form a reference record of the authentication information sharing process, as shown in Figure 4

Due to the business requirements of IoT applications, etc., the terminal device may need access to the network or data of other application domains. When a terminal device requests access to other application domains, the terminal device needs to be authenticated, i.e., cross-domain authentication. At this point, it can be combined with our previous study—IRBA scheme [21], where an end device with cross-domain access needs can make a cross-domain authorization request to the domain management node and the end device obtains the authorization from the management node of the application domain it needs to access through a threshold signature and credits the authorization to the alliance blockchain.

Similar to the terminal device registration process, an edge authentication node registers with the domain management node of the application domain to which it belongs in the same way to obtain its signature key pair. $EA_{ID}$ and its block number *Block_num* and the corresponding domain management node $DM_{ID}$ are also uploaded to the alliance blockchain to form a reference record of the authentication information sharing process.

*4.3. Authentication Phase.* In the authentication phase, the authentication process is divided into intradomain and cross-domain authentication.

*4.3.1. Intradomain Authentication.* The edge authentication node authenticates the identity of the registered device. The edge authentication node authenticates the following conditions to allow the device to communicate and access to other devices or to the system: (1) the $DM_{ID}$exists in the application domain blockchain; (2) the $TD_{ID}$ is registered in the application domain blockchain; (3) the $TD_{ID}$ is in the life cycle of the registration; (4) verify that the registration information is valid; and (5) verify that the $TD_{ID}$ signature is valid.

The authentication process within the application domain is described as follows:

(1) The device TD sends authentication request message $M_2 = \text{Request}_{Auth\_local} \| D_{ID} \| DM_{ID} \| TD_{ID} \| TS \| Hash(Sig(TD_{ID} \| DM_{ID} \| LT)_{DM_{SK}}) \| Sig(\text{Request}_{Auth\_local} \| D_{ID} \| DM_{ID} \| TD_{ID} \| TS)_{TD_{SK}}$, $Request_{Auth\_local}$ on behalf of the application domain authentication request, $D_{ID}$ on behalf of the application domain to which the terminal device belongs,$DM_{ID}$ indicates the domain management node identity associated with the device, $TD_{ID}$indicates the identity of the device, TS for the timestamp, and Sig indicates the Signature for $TD_{ID}$.

(2) The edge authentication nodes check for the presence of the application domain blockchain for $DM_{ID}$.

(3) If $DM_{ID}$ does not exist in the application domain blockchain, the authentication process ends with an error; otherwise, the edge authentication node obtains the master public key $P_{pub}$ and the public parameter *parameters* of $DM_{ID}$ and proceeds to the next authentication step.

(4) Checking for the presence of $TD_{ID}$ in the application domain blockchain.

(5) If the given $TD_{ID}$ does not exist in the application domain blockchain, the authentication process stops due to an error. Otherwise, the process will continue to the next step.

(6) Check that $TD_{ID}$ is within the life cycle of the registration.

(7) If not in the life cycle, stop the authentication; otherwise continue to the next step.

(8) Check that the hash of the $\|DM_{ID}$ signature of the registration information is consistent.

(9) Verifying the validity of the signature $Sig(\text{Request}_{Auth\_local} \| D_{ID} \| DM_{ID} \| TD_{ID} \| TS)_{TD_{SK}}$.

(10) If the signature validation is successful, the authentication is successful; if the signature validation fails, the authentication fails.

(11) The edge authentication node returns the authentication results to the device and signs the results using the private key.

(12) The terminal device confirms the validity of the result by verifying the signature of the authentication result.

Because the device authentication process is implemented at the nearest edge authentication node rather than on a cloud-

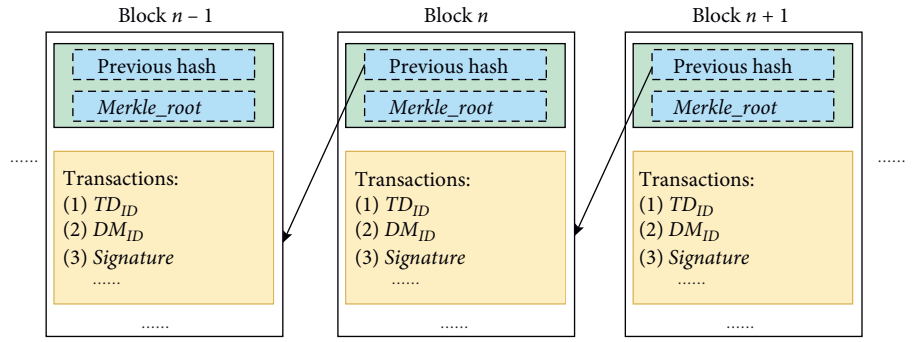



Figure 3: Structure and content of the application domain blockchain.

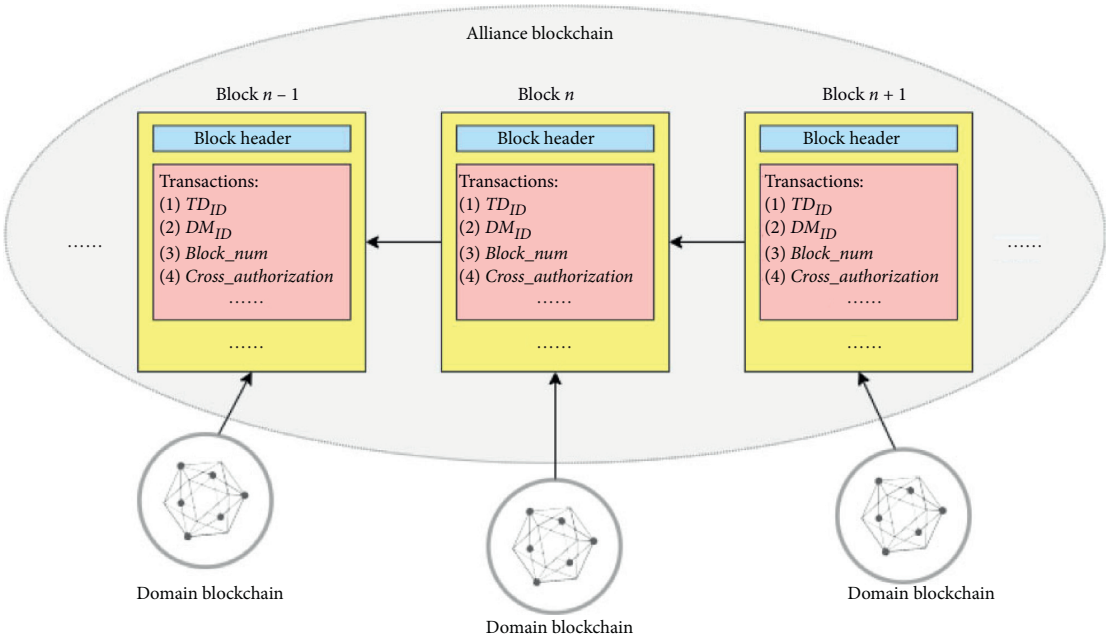



Figure 4: Store indexes in the alliance blockchain.

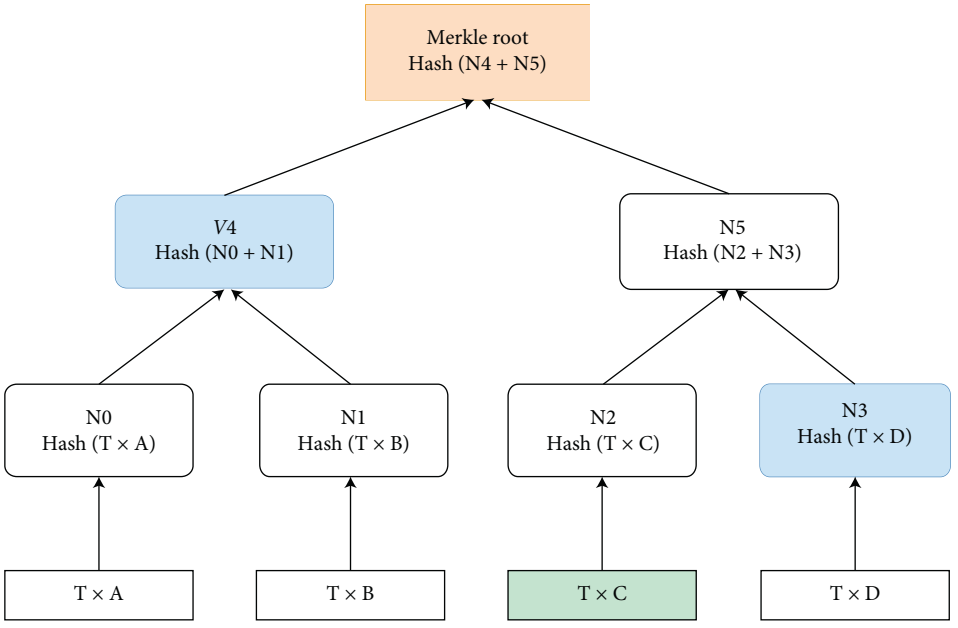



Figure 5: Merkle tree Path.

based server, the communication burden and latency are greatly reduced.

*4.3.2. Cross-Domain Authentication.* At present, a wide range of IoT applications are widely used and a large amount of IoT data is generated in different applications. Sharing data with other areas can be more useful as data sharing allows for a more rational allocation of resources and saves social costs. In order to achieve secure data sharing, future IoT networks need to implement a secure data sharing mechanism. In this case, if the terminal device needs to access data across application domains, the accessing application domain needs to be authenticated. However, if the previous authentication information block does not exist in the accessed application domain, the device will be required to re-register in the new application domain, which will take a lot of effort and time. Therefore, the authentication information should be able to be shared between application domains. In the cross-domain authentication process, we propose the use of sidechaining techniques to share authentication information from different application domains and use SPVs to prove the validity of the authentication information.

*(1) Sidechain and Merkel Tree.* Sidechain technology [47] was proposed to improve the scalability and extensibility of the blockchain, the basic idea being that digital assets can be transferred from one blockchain to another via sidechain protocols and to reduce the burden on the main chain, thereby increasing the throughput and speed of transactions [44, 48]. The flow of data between the main chain and the sidechains can be done using SPV (simple payment verification) proofs. SPV proofs consist of two parts: a list of block headers and a cryptographic proof, such as a Merkle proof, which indicates that a certain output occurred at a certain block in the list [49]. To prove that a certain transaction exists in a block, simply calculate the final Merkle root using the hash of this transaction against the hash of other related transactions and compare it to the root of the block header. If the result of the calculation agrees with the Merkle root of the block header, the transaction is proven to exist in this block. As shown in Figure 5, if we need to verify that a block contains a transaction Tx C and can get the hash value of the Merkle tree root, then we only need the Merkle path consisting of the hash values of N3 and N4 to prove it, as follows:

(1) First calculate the hash value of the transaction TX C, N2 = Hash (TX C).

(2) The hash value of the parent node is then obtained by summing the hash values of N2 and N3 and calculating the hash: N5 = Hash (N2 + N3).

(3) As above, calculate the hash value of the root node from the hash values of N4 and N5: root = hash (N4 + N5)

(4) Finally compare the hash value from the previous calculation with the root hash value of Merkle Tree in the block header; if it is the same then, the transaction TX C exists; otherwise it does not.

As we discussed in Section 3.1, existing blockchain-based authentication schemes suffer from authentication inefficiencies, scalability, and storage overloads. We propose to use a decentralized edge computing model to reduce authentication latency to improve authentication efficiency. To handle scalability and storage problems, we propose to build blockchains of different application domains using sidechain technology.

Sidechain, which is an extension of the blockchain, provides a decentralized peer-to-peer platform to maintain stored data while securely transferring authentication information between different application domains. The advantage of $A^2$ Chain's use of sidechain architecture is the independence of data and smart contracts, the alliance blockchain is primarily responsible for indexing, and the burden of the alliance blockchain does not increase with the number of application domains, avoiding the problem of rapid growth of data in the alliance blockchain. If the index between the alliance blockchain and the application domain blockchain is discarded, the application domain blockchain is an independently running blockchain that can run the domain authentication process independently. Based on the above, the sidechain technology not only improves the overall scalability of the system but also reduces the storage space of each application domain server and improves the search efficiency.

*(2) Cross-Domain Authentication Process.* The cross-domain authentication process is described below:

(1) The device TD sends an authentication request message $M_3 = \text{Request}_{Auth\_cross} \| D_{ID} \| DM_{ID} \| TD_{ID} \| TS \| parameters \| P_{pub} \| Sig(TD_{ID} \| DM_{ID} \| LT)_{DM_{SK}} \| Sig(\text{Request}_{Auth\_cross} \| D_{ID} \| DM_{ID} \| TD_{ID} \| TS)_{TD_{SK}}$ to the edge authentication node $EA_B$ to access the application domain B. The $D_{ID}$ terminal device belongs to the application domain, $\text{Request}_{Auth\_cross}$ is a cross-domain authentication request, $DM_{ID}$ indicates the domain management node identity associated with the device, $TD_{ID}$ indicates the identity of the device, $TS$ is a timestamp, $parameters, P_{pub}$ indicates the public parameters required for its signature, and Sig indicates the signature of $TD_{ID}$.

(2) The edge authentication node $EA_B$ receives the authentication request, and the authentication request is forwarded to the domain management node $DM_B$ of the application domain in which it is located.

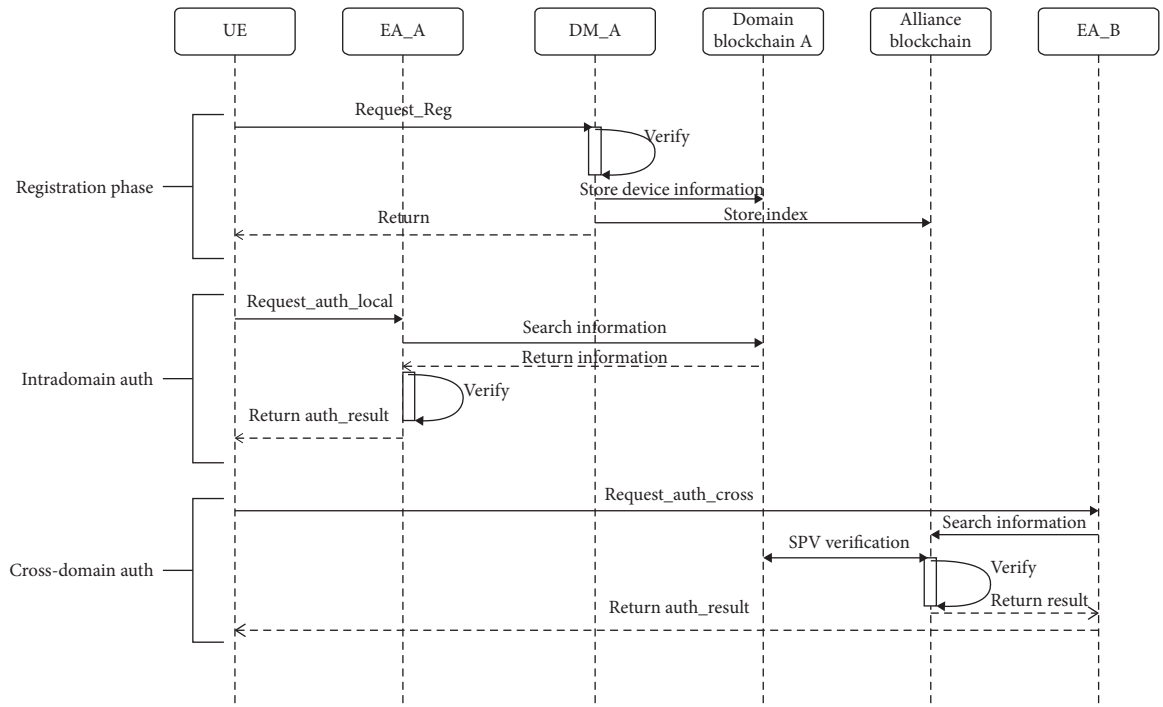



Figure 6: Overview of the authentication process. The device TD belongs to application domain A with intradomain authentication and cross-domain authentication, respectively.

(3) After the domain management node $DM_B$ receives the request, it searches the alliance blockchain containing the corresponding information (including$DM_{ID}$,$TD_{ID}$, and the corresponding *Block_num*) through the authentication terminal device ID and its management domain management node ID to judge that the terminal device has been registered. If it is not registered, then stop the authentication; otherwise, continue the authentication process.

(4) Get the *Merkle_Path* of $DM_{ID}$ and$TD_{ID}$ in the blockchain of the application domain to which it belongs via Block_num.

(5) Verify the authentication information provided by the device. Compute the hash value of the authentication information provided by the end device and the hash value of each node on its *Merkle_Path*, and compute the hash *Merkle_hash* of the Merkle root of the block in which it is located.

(6) Check that the computed *Merkle_hash* matches the *Merkle_Root* of the block header of the block in which it is located.

(7) Verify that the signature $Sig(\text{Request}_{Auth\_cross} \| D_{ID} \| DM_{ID} \| TD_{ID} \| TS)_{TD_{SK}}$ is valid.

(8) Get terminal device authorization information to verify that the signature is valid.

(9) Return authentication results to the edge authentication server.

(10) The edge authentication server forwards the results to the terminal device and attaches a signature that uses the private key.

(11) Upon receipt of the authentication result that has been signed by the $EA_B$ node, the terminal device will perform the same validation process to verify the signature to confirm the validity of the authentication result.

The authentication process and algorithm are shown in Figure 6 and Algorithm 3, respectively. At the end of the authentication process, the accessed application domain saves the end device's authentication information in the local blockchain so that the end device can later achieve fast authentication and simplify the cross-domain authentication process. (Algorithm 3).

## 5. Security Evaluation

In order to ensure the effective operation and service of the proposed authentication scheme, in this section, we analyse the proposed scheme for common security requirements and attacks in IoT applications:

(1) Integrity authentication: requests in the proposed system are signed by the requesting party using an identity-based signature algorithm before they are sent, and the final request message contains the data and the signature of the requestor. The receiving party can verify the message with the signature. In addition, authentication-related information is submitted to the alliance blockchain and the

```
Initialization phase
(1) if (DM_ID.exist = true) then
(2)     return error()
(3) else
(4)     creat.block(D_ID, DM_ID, parameters, P_pub, Domian_Chain)
(5)     creat.block(D_ID, DM_ID, Block_num, Alliance_Chain)
(6) end if
Registration phase
(1) if (D_ID.exist = true) then
(2)    get(parameters, P_pub)
(3)    If (TD_ID.exist = false) then
(4)       If (Sig.TD_ID = valid) then
(5)          creat.block(TD_ID, DM_ID, LT, Sig(TD_ID‖DM_ID‖LT)_(DM_sk), Domain_Chain)
(6)          creat.block(TD_ID, DM_ID, Block_num, cross_authorization, Alliance_Chain)
(7)          reg_sucess
(8)       end if
(9)    end if
(10) else
(11)    return error()
(12) end if
Intradomain authentication
(1) if (DM_ID.exist = true) then
(2)    get(parameters, P_pub)
(3)    if (TD_ID.exist&LT = true) then
(4)       if (Hash(Sig.DM_ID)&Sig.TD_ID = valid) then
(5)          return auth_sucess
(6)       end if
(7)    end if
(8) else
(9)    return error()
(10) end if
Cross-domain authentication
(1) if (DM_ID.exist&TD_ID.exit = true) then
(2)    get(parameters, P_pub)
(3)    DM_ID.Merkle_Path = getDomainChainPath(DM_ID.Block_num)
(4)    DM_ID.Merkle_Root = getDomainChainPath(DM_ID.Block_num)
(5)    DM_ID.Merkle_hash = computeMerkleTree(DM_ID_info, Merkle_Path)
(6)    if (DM_ID.Merkle_hash = DM_ID.Merkle_Root) then
(7)       TD_ID.Merkle_Path = getDomainChainPath(TD_ID.Block_num)
(8)       TD_ID.Merkle_Root = getDomainChainRoot(TD_ID.Block_num)
(9)       TD_ID.Merkle_hash = computeMerkleTree(TD_ID_info, Merkle_Path)
(10)      if (TD_ID.Merkle_hash = TD_ID.Merkle_Root) then
(11)         if (Sig.TD_ID)&cross_authorization = valid) then
(12)            return auth_sucess
(13)         end if
(14)      end if
(15)   end if
(16) else
(17)   return error()
(18) end if
```

Algorithm 3: Authentication mechanism.

application domain blockchain. Due to the features of the blockchain, the data cannot be tampered with once submitted, also ensuring the integrity of the message.

(2) Scalability: due to a large number of IoT applications and terminal devices, scalability is one of the important security requirements for IoT applications. In our proposed solution, terminal devices that can effectively authenticate access an identity-based signature scheme, and terminals do not need to store CA certificates, which is more flexible. The combination of application domain blockchain and alliance blockchain makes cross-domain authentication more convenient and business expansion of IoT applications more convenient and secure.

(3) Non-repudiation authentication: requests require a signature from the sender, and the private key used for the signature is generated by the sender's identifier and is kept by the sender. Therefore, the sender cannot repudiate the authentication request it has made.

(4) Authentication: for the terminal device that is going to access the network, it will first be registered in the system. The registration information will remain in the blockchain, and during the authentication process, the smart contract will check its legitimacy to allow the device to access the network.

(5) Mutual authentication: first, in our scheme, we assume that all domain management nodes and edge authentication nodes are trustworthy; if there are malicious nodes disguised as authentication nodes to perform phishing attacks on terminal devices, in our scheme, we require the authentication nodes to sign the authentication results, and the terminal devices can identify whether the authentication nodes are trustworthy and the validity of the authentication results by verifying the signatures.

(6) Sybil attack: in our proposed scheme, each endpoint device has a unique $TD_{ID}$ in the network and is associated with its registered application domain $D_{ID}$ and domain management node $DM_{ID}$ during the registration process, and each communication is preceded by endpoint authentication. Authentication takes place on the application domain blockchain and the federation blockchain. It is not possible for an attacker to forge legitimate nodes in the network to communicate with other nodes.

(7) Spoofing attack: because each communication must be authenticated and its signature must be verified each time to prove its unique identity, an attacker cannot fake the identity of another node for an attack.

(8) Message replay attack: in our scheme, authentication requests need to be signed with a timestamped token attached to them. A request with invalid signature validation will be rejected by the system.

(9) Denial of service attack: authentication servers are scattered around the edge of the network, and attackers cannot expend significant resources on denial of service attacks against all authentication nodes. Even if one or some of the nodes fail, the remaining nodes can still work without affecting the normal operation of the system.

Through the above analysis, we compare it with existing blockchain-based IoT authentication solutions and get the results as shown in Table 2. Our proposed solution is more comprehensive in terms of security.

Table 2: Security comparison of different schemes.

| | [37] | [17] | [18] | [28] | [29] | [50] | $A^2$ Chain |
|---|---|---|---|---|---|---|---|
| Sybil | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Message replay | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ |
| DOS | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Scalability | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ |
| Cross-domain authentication | ✓ | | | | | ✓ | ✓ |
| Decentralization | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

## 6. Performance Evaluation

*6.1. Experimental Setup.* We simulate two application domain blockchains and one alliance blockchain in our experiments, each containing the necessary entities, including domain management nodes and edge authentication nodes. The edge authentication node for each application domain runs on a separate host configured with an Intel (R) Core (TM) i7-6600U CPU with 8 GB of RAM. The domain management nodes run in a virtual machine that uses VMware Workstation 15 Pro hosted on an Intel (R) Core (TM) i7-6700 CPU 3.40 GHZ and 16 GB RAM. Four edge authentication nodes are set up per application domain. All machines are interconnected in a local network. The network connection of the virtual machines is configured to connect directly to the same LAN as the host in bridge mode. Terminal device operations are performed on a laptop with an Intel(R) Core(TM) i5-6300HQ and 4 GB RAM.

The blockchain platform we have chosen for the proposed system is Hyperledger Fabric [51]. Hyperledger Fabric provides a scalable and extensible architecture that provides the basis for developing blockchain applications with a modular architecture. Unlike public blockchains, the Fabric platform is license-based, meaning that the participants in the blockchain network are not completely trustless with each other, which ensures the trustworthiness of the nodes. Smart contracts in Fabric become chain codes, and the writing of chain codes in Fabric can be done using the Written in a common programming language (e.g., Go, Node.js, and Java) rather than being restricted to domain-specific languages (domain-specific language, DSL), and Fabric does not require any transaction fees to perform operations such as chain coding or querying blockchain information. For authentication services, we use remote authentication dial-in user service (RADIUS) [52] to build the authentication servicer, which is often used to provide AAA (authenticate, authority, and audit) services. We chose the open source project, YH-RADIUS [53]. This project implements an extensible development framework for RADIUS.

*6.2. Computing Consumption.* Each entity is involved in different cryptographic operations during the system operation. We summarize the cryptographic operations involved in the operation of the system, as shown in Table 3 (in statistics, the authorization issuance and verification of the cross-domain authentication process is not available yet). Also, Table 4 shows the computational burden of the different components of the system during its operation. It should be noted that operations such as hashing, integer addition, and multiplication are not taken into account, as they take very little time in the tests.

Table 3: Notation description of cryptographic operations.

| Notation | Description |
|---|---|
| $T_{PA1}$ | A point addition in $G_1$ |
| $T_{PA2}$ | A point addition in $G_2$ |
| $T_{SM1}$ | A scale multiplication in $G_1$ |
| $T_{SM2}$ | A scale multiplication in $G_2$ |
| $T_{SMT}$ | A scale multiplication in $G_T$ |
| $T_{ET}$ | A exponentiation in $G_T$ |
| $T_{BP}$ | A bilinear pairing |

Table 5 shows the computational overhead of the different components of the test in different processes. From the computational overhead, it can be seen that the main overhead of our proposed system lies in the initialization of the domain management nodes and the registration process of the end devices, which do not need to be performed in the authentication. The results show that common smart IoT devices can bear the computational burden of the system. In addition, the bilinear pair computation $g = e(P_1, P_{pub})$ in the used SM9 signature algorithm can be stored as a constant in advance in the end devices during the signature process to further reduce the computational burden.

To further demonstrate the advantage of our proposed system in terms of computational overhead, we compare it with existing authentication schemes ES$^3$A [25], CPAL [27], LCCH [54], and E-AUA [55]. We first compared the overhead on the user side, as shown in Figure 7(a). It is clear that our proposed scheme takes less time to implement on the user side than the other schemes, as shown in Figure 7(b). We compared the computational overhead on the service side, and our proposed scheme also outperforms the other schemes.

In order to assess the time cost of the relevant operations on the blockchain, we used the blockchain testing tool Hyperledger Caliper [56] to test each type of operation 10,000 times, with run times as shown in Table 6. The time cost of both the registration and the transaction process is about 200 ms, which may seem high, but it is acceptable. Therefore, registration and transactions do not happen frequently, but only when new devices are registered and device authentication information is changed. It takes about 10 milliseconds to look up the authentication information on the blockchain. In the authentication process, even taking into account the time spent querying the blockchain, the time spent is far less compared to other schemes.

We further counted the number of computational operations included in each scheme, as shown in Table 7. It can be seen from the table that ES$^3$A [25], CPAL [27], and LCCH [54] are the three schemes with more complex cryptographic operations. E-AUA [55] and our proposed scheme are simpler in terms of cryptographic operations compared to the other three schemes, thus achieving a better performance.

*6.3. Communication Consumption.* In this section, we analyse the communication overhead of the proposed scheme. The end device sends 192 bytes signed authentication request to the edge authentication node. In the intra-application domain authentication process, the edge authentication node obtains the relevant authentication information directly from the local blockchain to verify the signature and authenticate the end device. In the cross-domain authentication process, the edge authentication node forwards the request to the domain management node after receiving the authentication request. The domain management node obtains 196 bytes of authentication information in a two-way Peg protocol. At the end of the authentication, the 32-byte authentication result is returned to the interrupting device. Therefore, the communication overhead of our scheme during intradomain and cross-domain authentication is 228 bytes and 616 bytes, respectively.

The number of interactions of our proposed scheme is significantly less than other schemes in the authentication process, and there is no certificate exchange process. Therefore, compared with ES3A, LCCH, CPAL, and E-AUA whose communication cost is 1336 bytes, 2016 bytes, 1232 bytes, and 652 bytes, respectively, the communication cost of our scheme is much smaller and more efficient. In Figure 8, we list the cost comparison between the above schemes and our scheme, which shows more visually the advantages of our scheme in communication overhead performance.

*6.4. Storage Consumption.* Different from existing blockchain-based authentication schemes, in our scheme the alliance blockchain stores only the index information of the IoT devices, so only simplified blocks of information need to be stored additionally in the domain management nodes. In contrast, in the existing scheme, the entire blockchain is updated by all nodes each time a new device is registered.

In our scheme, it is assumed that there are 10 application domains, each containing 10,000 IoT devices. As described in the scheme, the authentication information of the blockchain storage device in the application domain is about 8 bytes in the federation blockchain storage device ID and block number. The block header is 80 bytes, and the authentication information is about 192 bytes. For the traditional blockchain-based case and our proposed scheme, the storage overhead is about 26.32 MB and 5.95 MB, respectively. Our proposed scheme is only 22.6% of the traditional blockchain-based scheme, which greatly reduces the available storage space.

## 7. Discussion

A$^2$ Chain builds a decentralized IoT authentication scheme by introducing blockchain-sidechain technology with edge computing technology. In this scheme, we utilize the edge computing model to reduce authentication latency. However, in IoT applications, there are some scenarios where the terminals are dense. In this case, the authentication service needs to implement load balancing and congestion control for authentication requests. This is one of our future research directions.

In addition, in our scheme, we apply an identity-based signature algorithm SM9, and the keys of the terminal

Table 4: Stats on time-consuming cryptographic operations.

| Phase | Computation cost | | |
|---|---|---|---|
| | UE | EA | DM |
| Setup and register | — | — | $2T_{SM1} + T_{SM2} + 2T_{SMT} + T_{PA2} + 2T_{ET} + 2T_{BP}$ |
| Intradomain authentication | $T_{SM1} + T_{SM2} + 2T_{SMT} + T_{PA2} + 2T_{ET} + 2T_{BP}$ | $T_{SM1} + T_{SM2} + 2T_{SMT} + T_{PA2} + 2T_{ET} + 2T_{BP}$ | \ |
| Cross-domain authentication | $T_{SM1} + T_{SM2} + 2T_{SMT} + T_{PA2} + 2T_{ET} + 2T_{BP}$ | — | $T_{SM1} + T_{SM2} + 2T_{SMT} + T_{PA2} + 2T_{ET} + 2T_{BP}$ |

Table 5: Computation cost on each entity.

| | Computation cost (ms) | | |
|---|---|---|---|
| | Setup and register | Intradomain authentication | Cross-domain authentication |
| UE | | 23.866 | 25.754 |
| EA | — | 15.872 | — |
| DM | 85.067 | — | 34.538 |

Table 6: Time costs (in s) of the blockchain.

| Operations | Registration | Query | Transfer |
|---|---|---|---|
| Max time (s) | 2.19 | 0.06 | 2.23 |
| Min time (s) | 0.03 | 0.01 | 0.03 |
| Avg time (s) | 0.18 | 0.01 | 0.16 |

Table 7: Comparison of time cost cryptographic operations in authentication.

| Scheme | User | Server |
|---|---|---|
| $A^2$ Chain | $T_{SM1} + T_{SM2} + 2T_{SMT} + T_{PA2} + 2T_{ET} + 2T_{BP}$ | $T_{SM1} + T_{SM2} + 2T_{SMT} + T_{PA2} + 2T_{ET} + 2T_{BP}$ |
| $ES^3A$ [25] | $6T_{SM1} + 3T_{SM2} + 3T_{BP} + T_{ET}$ | $3T_{SM1} + 8T_{BP} + 4T_{ET}$ |
| CPAL [27] | $16T_{ET} + 7T_{BP}$ | $10T_{ET} + 7T_{BP}$ |
| LCCH [54] | $30T_{E1}$[1] $+ 8T_{ET} + 8T_{BP}$ | $23T_{E1} + 6T_{ET} + 5T_{BP}$ |
| E-AUA [55] | $2T_{SM1} + 3T_{PA1} + T_{E1}$ | $2T_{BP} + 2T_{E1} + T_H$ |

[1] $T_{E1}$ indicates the exponentiation in $G_1$, and the meanings of other symbols are similar to the above definition.

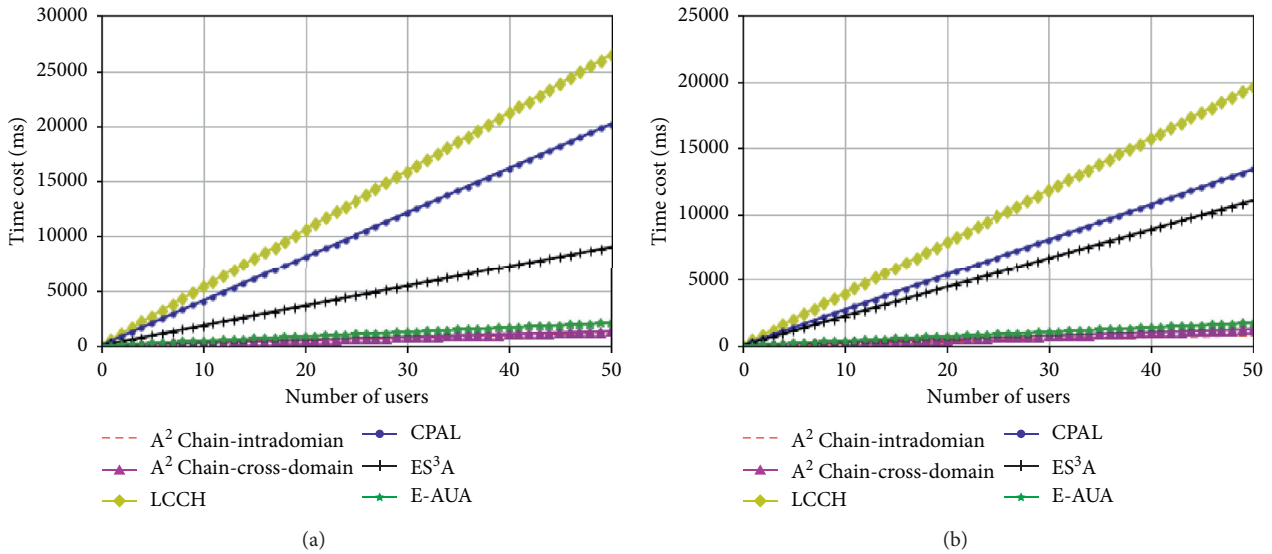



Figure 7: Comparison of computational overhead. (a) Cost on users for authentication. (b) Cost on servers for authentication.

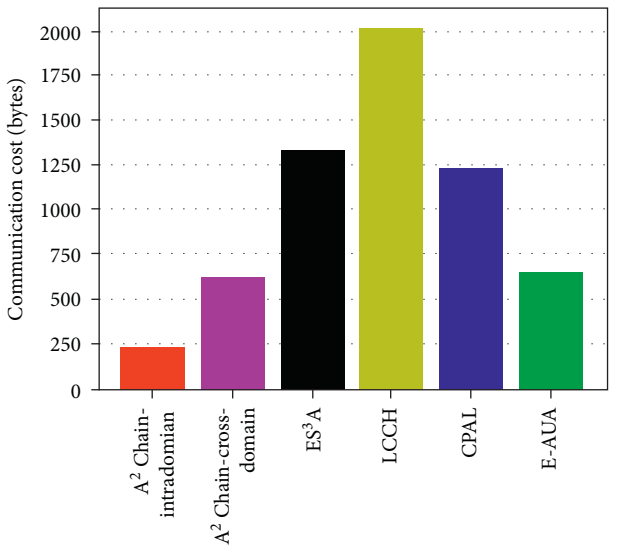



Figure 8: Communication cost comparisons of different schemes.

devices are generated by the domain management nodes. In our scheme, we assume that the domain management node is honest and trustworthy. In practice, there may be a malicious domain management node or a domain management node that is controlled by an adversary. In this case, it may lead to the leakage of the device's private key and jeopardize the security of the IoT application. In future work, we need to investigate the signature algorithm in case the key generation center is not fully trustworthy.

## 8. Conclusions

In this paper, we propose an application domain blockchain-alliance blockchain combined decentralized IoT authentication scheme called $A^2$ Chain, which enables a secure authentication information sharing process. Simulation results show that the scheme can significantly shorten the authentication time and reduce the communication cost and storage space compared to existing IoT authentication methods. In addition, we deploy the authentication server at the edge of the network through edge computing technology, which greatly reduces the authentication time and network latency.

## Data Availability

All relevant data are included within the article.

## Conflicts of Interest

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

## Acknowledgments

The authors would like to thank the Guangzhou University for the equipment support and the National Natural Science Foundation of China for the support. This research was supported by the National Natural Science Foundation of China (Grant No. 61976064), Project of National Defence Science, and Technology Innovation Zone (Grant No. 18-H863-01-ZT-005-027-02), and Equipment Pre-research Key Laboratory Fund Project (61421030203).

## References

[1] ITU-RM.2083-0, *IMT Vision Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, International Telecommunication Union, Geneva, Switzerland, 2015.

[2] J. Cao, P. Yu, M. Ma, and W. Gao, "Fast authentication and data transfer scheme for massive NB-IoT devices in 3GPP 5G network," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1561–1575, 2019.

[3] J. Cao, P. Yu, X. Xiang, M. Ma, and H. Li, "Anti-quantum fast authentication and data transmission scheme for massive devices in 5G NB-IoT system," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9794–9805, 2019.

[4] J. Li, M. Wen, and T. Zhang, "Group-based authentication and key agreement with dynamic policy updating for MTC in LTE-A networks," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 408–417, 2016.

[5] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for 5G and beyond networks: a state of the art survey," *Journal of Network and Computer Applications*, vol. 166, Article ID 102693, 2020.

[6] J. Xing zhong, X. Qingshui, M. Haifeng, C. Jiageng, and Z. Haozhi, "The research on identity authentication scheme of internet of things equipment in 5G network environment," in *Proceedings of the International Conference on Communication Technology, ICCT*, pp. 312–316, Xi'an, China, October 2019.

[7] I. Psaras, "Decentralised edge-computing and IoT through distributed trust," in *Proceedings of the MobiSys 2018—16th ACM International Conference on Mobile Systems, Applications, and Services*, pp. 505–507, Munich, Germany, June 2018.

[8] S. Behrad, E. Bertin, S. Tuffin, and N. Crespi, "A new scalable authentication and access control mechanism for 5G-based IoT," *Future Generation Computer Systems*, vol. 108, pp. 46–61, 2020.

[9] N. K. Pratas, S. Pattathil, C. Stefanovic, and P. Popovski, "Massive machine-type communication (mMTC) access with integrated authentication," in *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, May 2017.

[10] M. Nasimi, M. A. Habibi, B. Han, and H. D. Schotten, "Edge-assisted congestion control mechanism for 5G network using software-defined networking," in *Proceedings of the 2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 1–5, Lisbon, Portugal, August 2018.

[11] S. Hong, "P2P networking based internet of things (IoT) sensor node authentication by Blockchain," *Peer-to-Peer Networking and Applications*, vol. 13, no. 2, pp. 579–589, 2020.

[12] Z. Nezami, K. Zamanifar, K. Djemame, and E. Pournaras, "Decentralized edge-to-cloud load-balancing: service placement for the Internet of Things," 2020, http://arxiv.org/abs/2005.00270.

[13] M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of internet of things (IoT) authentication schemes," *Sensors*, vol. 19, no. 5, pp. 1141–1143, 2019.

[14] L. Kou, Y. Shi, L. Zhang, D. Liu, and Q. Yang, "A lightweight three-factor user authentication protocol for the information perception of IoT," *Computers, Materials & Continua*, vol. 58, no. 2, pp. 545–565, 2019.

[15] C. Ellison and B. Schneier, "Ten risks of PKI: what you are not being told about public key infrastructure," *Public Key Infrastructure: Building Trusted Applications and Web Services*, vol. 14, no. 1, pp. 299–306, 2004.

[16] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 33–39, 2018.

[17] S. Guo, X. Hu, S. Guo, X. Qiu, and F. Qi, "Blockchain meets edge computing: a distributed and trusted authentication system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1972–1983, 2020.

[18] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, and K. Salah, "A user authentication scheme of IoT devices using blockchain-enabled fog nodes," in *Proceedings of the 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–8, Aqaba, Jordan, October-November 2018.

[19] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: a state of the art survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 858–880, 2019.

[20] T. Hewa, A. Braeken, M. Ylianttila, and M. Liyanage, "Blockchain based Automated Certificate Revocation for 5G IoT," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Dublin, Ireland, June 2020.

[21] X. Jia, N. Hu, S. Su et al., "IRBA: an identity-based cross-domain authentication scheme for the internet of things," *Electronics*, vol. 9, no. 4, p. 634, 2020.

[22] K. Kaur, S. Garg, G. Kaddoum, F. Gagnon, and S. H. Ahmed, "Blockchain-based lightweight Authentication mechanism for vehicular fog infrastructure," in *Proceedings of the 2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, Shanghai, China, May 2019.

[23] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," in *Proceedings of the 2015 10th International Conference for Internet Technology and Secured Transactions, ICITST 2015*, pp. 336–341, London, UK, December 2015.

[24] A. Esfahani, G. Mantas, R. Matischek et al., "A lightweight Authentication mechanism for M2M communications in industrial IoT environment," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 288–296, 2019.

[25] J. Ni, X. Lin, and X. S. Shen, "Efficient and secure service-oriented authentication supporting network slicing for 5G-enabled IoT," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 644–657, 2018.

[26] H. Gross, M. Hölbl, D. Slamanig, and R. Spreitzer, "Privacy-Aware Authentication in the Internet of Things," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, M. Reiter and D. Naccache, Eds., vol. 9476, pp. 32–39, Springer International Publishing, Cham, Switzerland, 2015.

[27] C. Lai, H. Li, X. Liang et al., "CPAL: a conditional privacy-preserving authentication with access linkability for roaming service," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 46–57, 2014.

[28] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of Trust: a decentralized blockchain-based authentication system for IoT," *Computers & Security*, vol. 78, pp. 126–142, 2018.

[29] Z. Bao, W. Shi, D. He, and K.-K. R. Chood, "IoTChain: a three-tier blockchain-based IoT security architecture," 2018, http://arxiv.org/abs/1806.02008.

[30] U. Khalid, M. Asim, T. Baker, P. C. K. Hung, M. A. Tariq, and L. Rafferty, "A decentralized lightweight blockchain-based authentication mechanism for IoT systems," *Cluster Computing*, vol. 23, no. 3, p. 2067, 2020.

[31] H. Liu, P. Zhang, G. Pu, T. Yang, S. Maharjan, and Y. Zhang, "Blockchain empowered cooperative authentication with data traceability in vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4221–4232, 2020.

[32] Z. Cui, F. Xue, S. Zhang et al., "A hybrid BlockChain-based identity authentication scheme for multi-WSN," *IEEE Transactions on Services Computing*, vol. 13, no. 2, p. 1, 2020.

[33] N. Shahin, R. Ali, S. Y. Nam, and Y.-T. Kim, "Performance evaluation of centralized and distributed control methods for efficient registration of massive IoT devices," in *Proceedings of the 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, vol. 2018, pp. 314–319, Prague, Czech Republic, July 2018.

[34] O. Alphand, M. Amoretti, T. Claeys et al., "IoTChain: a blockchain security architecture for the Internet of Things," in *Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC*, pp. 1–6, Barcelona, Spain, April 2018.

[35] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar, and K.-K. R. Choo, "HomeChain: a blockchain-based secure mutual authentication system for smart homes," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 818–829, 2020.

[36] M. A. Jan, W. Zhang, M. Usman, Z. Tan, F. Khan, and E. Luo, "SmartEdge: an end-to-end encryption framework for an edge-enabled smart city application," *Journal of Network and Computer Applications*, vol. 137, pp. 1–10, 2019.

[37] M. Shen, H. Liu, L. Zhu et al., "Blockchain-assisted secure device authentication for cross-domain industrial IoT," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 942–954, 2020.

[38] M. Ma, G. Shi, and F. Li, "Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario," *IEEE Access*, vol. 7, pp. 34045–34059, 2019.

[39] M. A. Xiaoting, M. A. Wenping, and L. I. U. Xiaoxue, "A cross domain authentication scheme based on blockchain technology," *Acta Electronica Sinica*, vol. 46, no. 11, pp. 2571–2579, 2018.

[40] Y. Chen, G. Dong, J. Bai, Y. Hao, F. Li, and H. Peng, "Trust Enhancement Scheme for Cross Domain Authentication of PKI System," in *Proceedings of the 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 103–110, Guilin, China, October 2019.

[41] J. Cui, L. Wei, J. Zhang, Y. Xu, and H. Zhong, "An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1621–1632, 2019.

[42] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019.

[43] Y. Jiang, C. Wang, Y. Wang, and L. Gao, "A cross-chain solution to integrating multiple blockchains for IoT data management," *Sensors*, vol. 19, no. 9, pp. 2042–2060, 2019.

[44] G.-H. Hwang, P.-H. Chen, C.-H. Lu et al., "A multi-chain architecture with distributed auditing of sidechains for public blockchains," in *Proceedings of the Blockchain-ICBC*, vol. 10974, pp. 47–60, Springer International Publishing, Honolulu, HI, USA, September 2018.
[45] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, vol. 196 LNCS, pp. 47–53, Springer Berlin Heidelberg, Berlin, Heidelberg, 1985.
[46] F. Yuan and Z. Cheng, "Overview on SM9 identity-based cryptographic algorithm," *Journal of Information Security Research*, vol. 2, no. 11, pp. 1008–1027, 2016.
[47] A. Back, M. Corallo, and L. Dashjr, "Enabling blockchain innovations with pegged sidechains," pp. 1–25, 2014, http://newspaper23.com/ripped/2014/11/, http://www.blockstream.com.sidechains.pdf.
[48] A. Garoffolo, D. Kaidalov, and R. Oliynykov, "Zendoo: a zk-SNARK verifiable cross-chain transfer protocol enabling decoupled and decentralized sidechains," 2020, http://arxiv.org/abs/2002.01847.
[49] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2009, https://bitcoin.org/en/bitcoin-paper.
[50] M. Li, H. Tang, A. R. Hussein, and X. Wang, "A sidechain-based decentralized authentication scheme via optimized two-way Peg protocol for smart community," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 282–292, 2020.
[51] "Hyperledger fabric," https://www.hyperledger.org/projects/fabric.
[52] Remote authentication dial in user service (RADIUS).
[53] "yh-radius," https://github.com/cometowell/yh-radius.
[54] J. K. Liu, C.-K. Chu, S. S. M. Chow, X. Huang, M. H. Au, and J. Zhou, "Time-bound anonymous authentication for roaming networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 178–189, 2015.
[55] X. Zeng, G. Xu, X. Zheng, Y. Xiang, and W. Zhou, "E-AUA: an efficient anonymous user authentication protocol for mobile IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1506–1519, 2019.
[56] "Hyperledger caliper," https://github.com/hyperledger/caliper.

Hindawi
Mobile Information Systems
Volume 2020, Article ID 8863385, 9 pages
https://doi.org/10.1155/2020/8863385

*Research Article*

# HAL-Based Resource Manipulation Monitoring on AOSP

**Thien-Phuc Doan, Jungsoo Park, and Souhwan Jung**

*Communication Network Security Laboratory, Soongsil University, Seoul 06978, Republic of Korea*

Correspondence should be addressed to Souhwan Jung; souhwanj@ssu.ac.kr

Received 25 September 2020; Revised 29 October 2020; Accepted 23 November 2020; Published 2 December 2020

Academic Editor: Vishal Sharma



Nowadays, Android malware uses sensitive APIs to manipulate an Android device's resources frequently. Conventional malware analysis uses hooking techniques to detect this harmful behavior. However, this approach is facing many problems, such as low coverage rate and computational overhead. To solve this problem, we proposed *HALWatcher*, an alternative technique to monitor resource manipulation on Android Open Source Project (AOSP). By modifying Hardware Abstract Layer (HAL) resource accessing interfaces and their implementation, we can embed more monitoring functions at critical methods that are in charge of transferring data between the Hardware Driver and the Framework Layer. Hence, *HALWatcher* provides a lightweight and high coverage rate system that can perform resource manipulation monitoring for Android OS. In this paper, we prove that the hooking technique is limited in detecting resource manipulation attacks. Besides that, *HALWatcher* shows an outperform detection rate with a low computational effort.

## 1. Introduction

Many studies have been published in recent years on malicious code on Android devices [1–5]. They have invested a huge effort to generate effective architectures to defend against Android malware [6]. Generally, a detection solution has two main parts: Malware analysis and Detection algorithm. The analysis part can be done by two main approaches: Static and Dynamic Approach. In the end, this part provides a set of patterns or features which are fed to the Detection algorithm.

Both Static and Dynamic approaches try to figure out the malicious behaviors of malware. Resource manipulation is one of the most popular harmful attacks. By different techniques, the malicious apps manipulate user's device resources, e.g., Camera, Phone, and SMS, to steal sensitive information or send messages to premium numbers without user awareness [7]. The privilege escalation attack is even more dangerous. They can take over device resources without user interaction. So there is a great need to design a detecting model for resource manipulation attacks.

Existing analysis techniques can be applied to solve this problem. One of the typical static analysis approaches is building a flowgraph based on critical API calls. Based on that, we can deduce what resources the malicious app manipulates. However, Android malicious samples have been obfuscated or encrypted using various evasion techniques. This difficulty significantly decreases detection accuracy. Dynamic behavior analysis seems to be a good supplementary solution. This approach uses hooking tools, such as xPosed, Frida, etc., to monitor and trace malware behaviors. There are many sensitive APIs that are related to controlling the device's resources. Hooking into all of these APIs may cause the computational overhead problem due to mobile devices' limited computing resources. Therefore, malicious apps are often crashed while the analysis is operating. Moreover, hooking tools are detectable due to its direct interference with the process's memory.

To address the limitation of hooking techniques, we design *HALWatcher*, a general method for monitoring Android hardware resources inside Hardware Abstract Layer (HAL). The idea is that HAL provides the interface for the communication between the Android Framework Layer, which handles the requests of getting resources from applications or processes, and the Hardware Driver inside the kernel layer. By modifying HAL interfaces and

implementation codes, we can keep track of all the manipulating hardware resources without any knowledge requirement about various vendors' hardware drivers. Moreover, *HALWatcher* does not require root permission because it is already working as a part of the Android system. Furthermore, the detection of *HALWatcher* in the system is almost impossible because of the various ways of modifying HAL in such a large number of developers. Although the flexibility is not high, our method dramatically reduces the amount of data collection work from dynamic analysis while also providing sufficient information for Android devices' protection service against bad actors.

*HALWatcher* architecture can be applied to develop various applications both in research and industry field. It is a useful technology to track malware behaviors, then constructing a complete dataset for dynamic analysis is achievable. Besides, this technology is suitable for all Android mobile device hardware because it only interferes with Hardware Abstract Layer. Therefore, developing a hardware resources manipulation system on real-world Android devices is uncomplicated. Moreover, root privilege is nonessential for *HALWatcher*, in which other hooking frameworks are strongly dependent.

In summary, this paper has the following contributions.

We demonstrated that the hooking techniques might not be useful for detecting resource manipulation attacks.

We proposed *HALWatcher*, an efficient and lightweight method to detect resource manipulation attacks. By modifying HAL, this module runs along with the Android system so that it is almost undetectable.

The rest of this paper is organized as follows: the background of HAL and resource manipulation monitoring is introduced in the second section. In the third section, we discuss how to detect resource manipulation attacks. After that, we present *HALWatcher*, a HAL based resource monitoring system running along with the Android OS. The implementation and the design of experiments are shown in the next section. Finally, we will discuss future work and conclusions about our work.

## 2. Background and Related Work

*2.1. Resources Manipulation.* Android malware analysis is well-studied nowadays [8]. Analytical techniques include dynamic analysis [9–13] and static analysis [5, 14–17]. Some frameworks seek to classify malicious code through application behavior following signature [9, 14], while others track data flow [15, 18]. Static analysis is the way to understand the application by finding the signatures of malicious code (e.g., permissions that the application has declared, APIs that the application uses). Dynamic analysis directly executes malicious apps in a sandbox environment [3], then collects the necessary information and organizes them to process. The data from dynamic analysis and static analysis are then fed into algorithms to assess application behavior. In particular, the application of machine learning and deep learning in the classification of malicious apps is trendy due to its high accuracy [6, 12, 19–22].

Data collection from the dynamic and static analysis has always faced many difficulties and obstacles [4]. Evasion techniques make it difficult for static analysis to locate the used APIs or to figure out the execution flow of data [15]. Meanwhile, dynamic analysis has difficulty finding ways to execute all the behavior of the application being analyzed automatically [10], along with a large amount of information that may not be needed after running malicious apps. However, we must recognize the flexibility that current dynamic analytical techniques are very high. The hybrid approach combines static analysis and dynamic analysis to solve the limitations of each technique [4, 23]. Wong et al. proposed IntelliDroid [24] that generates input for the dynamic analysis using the static analysis technique.

Malware behavior tracking is a common problem. One of the efficient ways to track malicious behavior is to detect resources that are manipulated by malware from an Android device. Almost all attackers' purposes are trying to steal sensitive data from the user by manipulating the phone resources such as CAMERA, MICROPHONE, PHONE, and SMS. Jiang et al. designed a resource management system architecture to collect data for behavior detection [25]. Static analysis is limited to detect unauthorized resource usage. Meng et al. constructed a graph-based model to describe the control flow of an application. However, their approach does not seem much effective with 89.5% precision [26]. Zhao et al. leveraged the power of Androguard to extract a set of sensitive APIs to represent the application's behavior [27]. The dynamic analysis uses hooking techniques. Using Java function hooking technology, Soewito and Suwandary successfully illustrate that their proposal is applicable to data leakage prevention [28]. Hooking technologies are easy to install and detectable to monitor resource manipulation. For instance, Frida, a hooking framework, interferes with the application's memory, which process it needs to analyze. The agent and then needs high privileged access because the Linux kernel does not allow any processes to interfere with each other's memory without authorization. Therefore, to use the hooking techniques, the device must be rooted, or the agent of the hooking framework must be attached to the application they want to analyze.

Frida framework has two ways to hook the function of target APK. First, it needs to run *frida server* inside the devices as root permission or nonroot permission with enough capability to access other processes 'memory. The *frida server* then modifies the memory to overwrite the functions which are specified in the JavaScript-based hooking script. The target app will use the overwritten function instead of the original function for its execution. For the second way of using Frida, we need to attach the *frida-agent.so* library into the target APK and repackage the APK. *frida-agent.so* then acts as *frida server* but with no root privileges requirement because the attaching agent is now a part of the application to fully access its memory.

*Strace* is a possible solution to hide from evasion malware. However, the massive log of the system call is quite complex to process.

*2.2. Hardware Abstract Layer.* A HAL (Hardware Abstract Layer)[1] defines a standard interface for hardware vendors to implement, enabling Android to be agnostic about lower-level driver implementation. Using a HAL allows you to implement functionality without affecting or modifying the higher-level or lower-level system. The legacy HALs is the old architecture for Android Nougat (7.0) and the previous versions. In Android 8.0 and higher, the architecture is designed to meet the requirements of modularity.

## 3. Resource Manipulation Attack Detection

Currently, the dynamic analysis approach can use the hooking technique for detecting resource manipulation attacks. Figure 1(a) describes the method of using Frida to keep track of the *SendSMS* function. To know whether the app manipulates SMS resources by requesting SendSMS or not, we hook into *sendTextMessage()*. The logging method is used in this example to gather the manipulation information. We found some disadvantages to this method.

*Detectable.* Hooking techniques require access to application memory during the analysis. Self-checking memory is one way to figure out the strange agents (e.g., *frida-agent.so*). Besides, the requirement of root privileges (i.e., in the case of *frida server*) makes it exposed to the Android system. Darvin claims in his blog[2] that there are many ways for an application to detect the existence of Frida inside the execution environment. Szczepanik et al. proposed an algorithm using stack-trace on detecting hooking tools[29].

*Messy or Imperfect Data.* Some SDK APIs call each other when the app requests a resource. Even the analyzer tries to reduce the number of sensitive APIs, but this is hard work. On the other hand, some APIs might be missed in the hooking list leading to the increasing of *False-positive* and *False-negative*.

*Inapplicable to End-User Products.* Most of the hooking techniques are applied in solving behavior analysis problems. It is hard to include these techniques into real end-user products due to the risk of misappropriation for wrong usage (i.e., bypass the protection mechanisms of apps).

Modifying HAL is the best choice for monitoring manipulation resources for many reasons. Firstly, all hardware resource requests go through HAL. Therefore, monitoring resources based on HAL gives a high coverage rate. Secondly, HAL is independent of the hardware driver. The monitoring module in HAL can work correctly for a wide range of Android devices. Last but not least, even the attack aimed to get rooted in the Android device, it cannot disable the monitoring module in HAL because this module is not running as a service, a part of the Android Operation System. We started to investigate the HAL source code and then came to these conclusions.

First, we can simply add more functions to monitor the manipulating resources with a small coding effort. Listing 1 shows a simple logging code of *sendSms()* function inside HAL. *Line 5* is the only code that we need to add. On the other side, Frida needs more effort (i.e., see Listing 1 to hook into *sendTextMessage(),* which will request for sending SMS (i.e., the same resource of example in Listing 1).

Second, the information collected from HAL interfaces or functions is sufficient for detecting resource manipulation attacks. There are multiple Android APIs that act the same behavior. For instance, both *sendTextMessage()* and *sendMultipartTextMessage()* can be used to send SMS through radio network. Moreover, *sendTextMessage()* have 2 different overloading methods. Therefore, there is a need to develop two hooking functions for each *sendTextMessage()* to cover all the resource manipulation APIs. Besides, HALWatcher performs monitoring procedure accurately by adding one line of code (i.e., for logging) into the *sendSms()* implementation function (i.e., for the sendSms interface) as shown in Listing 2.

## 4. *HALWatcher*: Resource Manipulation Monitoring Module

*HALWatcher*, as shown in Figure 2, then works as a part of the Android Operation System. Therefore, there is no requirement of the rooted system or repackaging the target application. All of the installed packages from the Play store or other Vendor market can be monitored. Besides, the process generated from a Remote Code Execution (RCE) attack is also under monitoring. Moreover, because of the diversity of vendor Android firmware (or ROM) types, the detection of *HALWatcher* is almost impossible. Our model generates information whenever the resource requests the hardware. Therefore, the amount of information (e.g., logs) is significantly reduced but ensures that all resource manipulation behavior is recorded and reported. In the next subsection, we will give some examples of how to build *HALWatcher* in many types of hardware resources.

Based on the previous section's conclusions examining the HAL source code, we provide a detailed design for *HALWatcher*. First of all, all requests to access hardware information and resources will start from the Framework Layer, namely, the Java Native Interface (JNI). We consider the Malware or RCE attack in equal measure because all hardware resource manipulating requests must go through the JNI. The information will then be moved down to the Hardware Abstraction Layer (HAL). In HAL, interfaces are feature independent; that is, there are no interfaces that share the same purpose. At the critical methods of each resource type (which we discuss in more detail in the next section), we embed one or more code lines to record any action and related information about the manipulated resource. These code lines are called *resource monitoring modules*. Listing 2 shows an example of one resource monitoring module. At line number 5, we add one line of code into the *RadioImpl::sendSms* interface to monitor the SMS resource by logging whenever this interface is called. In short, all resource monitoring modules are developed following three steps: figure out resources implementing interface source code in HAL, embed monitoring functions into the interfaces, manage, and send monitoring information to monitoring service.

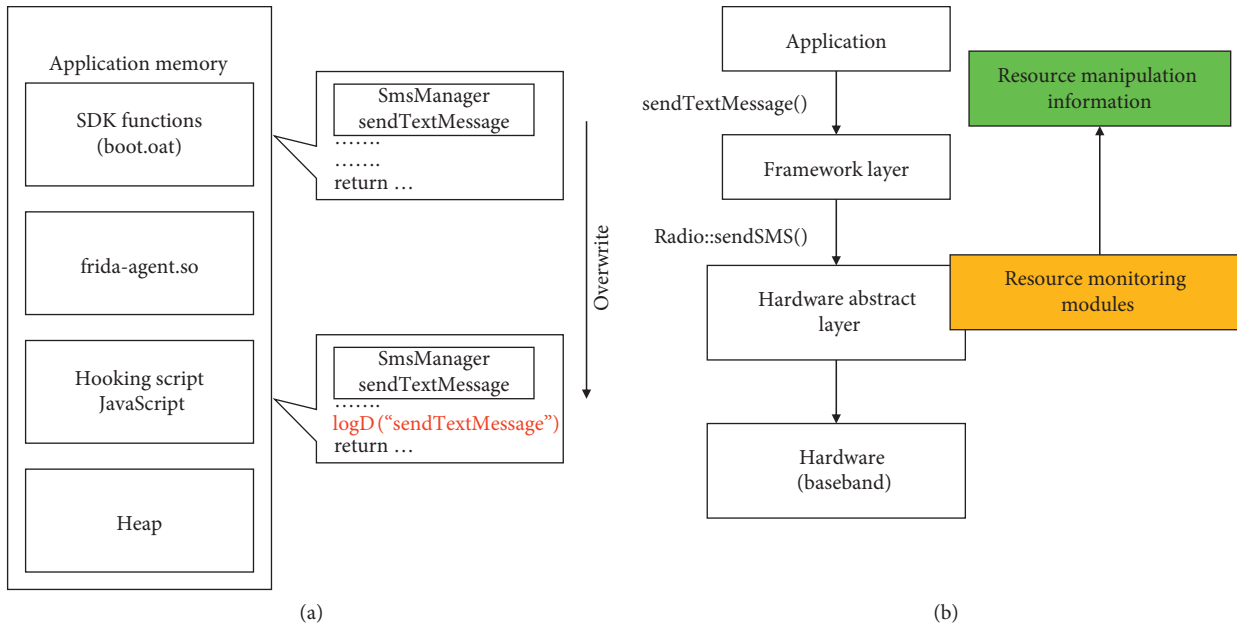



Figure 1: Frida vs. *HALWatcher* architecture for SMS resource monitoring. (a) Using Frida to monitor whether sending SMS API is called. Note that there are several APIs to send SMS. (b) HAL based resource monitoring.

```
(1) var hook = Java.use( android.telephony.SmsManager );
(2) hook.sendTextMessage.overload( java.lang.String , java.lang.String , java.lang.String ,
    android.app.PendingIntent , android.app.PendingIntent ).implementation =
(3) function(arg_0, arg_1, arg_2, arg_3, arg_4){
(4)   var olog = Java.use( android.util.Log );
(5)   olog.d( sendTextMessage is called );
(6)   return this.sendTextMessage(arg_0, arg_1, arg_2, arg_3, arg_4);
   }
```

Listing 1: Example of Frida hooking into *sendTextMessage()* function (Android SDK).

```
(1) Return<void> RadioImpl:sendSms(int32_t serial, const GsmSmsMessage& message) {
(2) #if VDBG
(3)   RLOGD( sendSms: serial %d , serial);
(4) #endif
(5)   RLOGD( [%d] [HALWatcher] RIL_REQUEST_SEND_SMS: serial %d ,(int)time(NULL), serial);
(6)   dispatchStrings(serial, mSlotId, RIL_REQUEST_SEND_SMS, false,
(7)     2, message.smscPdu.c_str(),
(8) message.pdu.c_str());
      return Void();
   }
```

Listing 2: Example of HAL modifying in *sendSms()* function (HAL).

## 5. Resources Manipulation Monitoring Module in HAL

The resource monitoring modules are basically located in many critical HAL interfaces and implementation functions. In this work, we illustrate *HALWatcher* in monitoring SMS, PHONE, and CAMERA resources in detail. For other resources, we conduct a list of modules' locations of *HALWatcher* for further works.

*5.1. SMS and Phone.* Both SMS and Phone permissions on the Android device allow the application of the right to access carrier service. Android communicates with carrier providers through SIM (subscriber identity module), which needs a radio baseband device to run radio service. Each baseband device has its own vendor's driver represented as "*libril-vendor.so*" file. Android HAL performs a RIL (Radio Interface Layer) connecting Android Framework and vendor's driver. Therefore, all resource usage related to SMS and

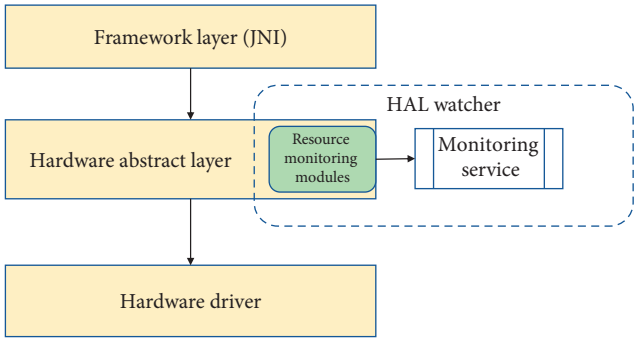



Figure 2: *HALWatcher* general design.

Phone permissions go through RIL. While analyzing the RILD, we found two types of RIL command: *REQUEST* and *UNSOL*. *REQUEST* command is used by the Android Framework Layer (i.e., *rilj*), to request data (e.g., signal strength) or functions (e.g., send SMS, conduct a call). *UNSOL* stands for *unsolicited* responses, which originate from the baseband (e.g., new SMS).

*5.2. Camera.* Recently, sensitive information leaked through Camera. The Android application might run as a service that has no activity screen. In that case, the malicious application or vulnerable application attacked by an intruder can handle a camera resource silently without any notification to the user. HAL Camera interfaces have been implemented inside *hardware/camera/device/1.0/default/CameraDevice.cpp*.

These interfaces provide methods to communicate to Camera hardware driver such as *getCameraInfo(), dumpState()*, etc. Some focus on managing the memory resources for Camera device (e.g, *CameraHeapMemory()*), others open or close Camera device (e.g., *Camera::open(), Camera::close()*), and others provide normal task of the Camera functions such as *startRecording(), stopRecording(), takePicture(), cancelPicture()*. In order to keep an eye on Camera resources, we create logs about the function when the Camera is opened and closed. We also keep a log for the working time of the Camera because of the irregular using period.

*5.3. Other Resources Monitoring.* Similar to Camera resources, other resources also have the implementation code inside/hardware/resource_name. Table 1 shows the list of hardware resources and their implementation source code. Regarding resource monitoring, we can add more features than logging the needed information for those resources.

*5.4. Resources Manipulation Monitoring Service.* This service is responsible for getting data from the monitoring module inside HAL. By using the logging method, this component is not required because the log data can be got from *logcat* command. HAL resources manipulation module does not log for any sensitive data of the user or the phone so that this log data can be public. The service can be any application inside or outside the phone, which is the only convenient purpose for the user or analysis researcher.

## 6. Implementation and Evaluation

*6.1. Implementation.* We implemented our approach using AOSP version 9.0.0_r47 on a Hikey960 board[3]. The limitation of the Hikey960 is that it does not support full hardware that usually exists on a real phone (e.g., Vibrator). Therefore, to prove that the HAL modifying method for resource monitoring is possible, we focused on SMS and PHONE resources. These resources HAL interface are implemented in hardware/ril/ as known as Radio Interface Layer Deamon, which stands in the middle of the communication between the Android Framework Layer (RILJ) and the Hardware driver (i.e., carrier baseband).

Hikey960 board has a list of hardware devices that need the corresponding HAL modules to work with. These components are defined in *hikey/device-common.mk* and *hikey/hikey960/device-hikey960.mk* config file. Because of the limitation of hardware that the Hikey960 board supports, we can only see the impact of HAL modifying when we change the source code of the component that we listed in Table 2.

Hikey960 does not have a SIM card reader. Therefore, we used the Huawei 4G E173 USB stick as a SIM card reader. Then we added the Huawei lib-ril (i.e., the driver that supports E173 USB stick to work as a baseband device) at the driver layer of the final compiled AOSP. Typically, the Linux kernel will accept USB as a storage device. Therefore, to make the kernel recognize the dongle as a 3 G/4G USB device, we switched the USB mode of the device using *usb_modeswitch* tool. Then we needed to customize the Hikey960 kernel (i.e., kernel 4.9) and add more kernel module that supports the *usb_modeswitch* function. We also customized the RIL daemon to automatically switch the USB device to PPP mode before loading the driver.

We evaluate our method by modifying directly to log information that goes through radio methods in RIL. In total, we customize several requesting methods, which are most related to send SMS and conduct phone calls. We also can hook the other implementation functions in the same way.

*6.2. Evaluation*

*6.2.1. High Coverage Rate.* We used the default Messaging of AOSP to send normal SMS, Premium SMS, conduct Phone calls, and send USSD. Then, we tried to send an SMS without using the application. We found a way to send SMS messages through *iSms service*-a default service in AOSP. We denote that Frida can not work in this situation of monitoring SMS resources. The iSms service is called by service call command through ADB shell[4] with the form: *adb shell service call isms 7 i32 0 s16* "*com.android.mms.service*" *s16* "*+1234567890*" *s16* "*null*" *s16* "*Hello*" *s16* "*null*" *s16* "*null*". To prove that *HALWatcher* can work without root privileges, we removed/*xbin/su* binary and built a *non-userdebug* version of AOSP to unroot the ADB shell. Finally, we ran *Trojan-SMS* on both

Table 1: HAL implementation source code related to some group of Dangerous and Protection permission.

| Resource hardware | Permission level | HAL interface |
|---|---|---|
| CAMERA | Dangerous | Hardware/interface/camera/device/1.0/default/ |
| LOCATION | Dangerous | Hardware/interface/GNSS/1.0/default/ |
| PHONE/SMS | Dangerous | Hardware/ril/ |
| SENSORS | Dangerous | Hardware/interface/sensors/1.0/default/ |
| AUDIO | Dangerous | Hardware/interface/audio/core/2.0/default/ |
| NFC | Protection | Hardware/interface/nfc/1.0/default/ |
| BLUETOOTH | Protection | Hardware/interface/bluetooth/1.0/default/ |
| WIFI | Protection | Hardware/interface/wifi/1.2/default/ |
| VIBRATOR | Protection | Hardware/interface/vibrator/1.0/default/ |

Table 2: Hikey960 HAL components.

| Hardware | Package name |
|---|---|
| WIFI | https://android.hardware.wifi@1.0-service |
| AUDIO | https://android.hardware.audio@2.0-impl<br>https://android.hardware.audio.effect@2.0-impl<br>https://android.hardware.broadcastradio@1.0-impl<br>https://android.hardware.soundtrigger@2.0-impl |
| PHONE/SMS | rild |
| DRM | android.hardware.drm@1.0-impl |
| BLUETOOTH | android.hardware.bluetooth@1.0-service.btlinux |
| POWER | android.hardware.power@1.0-impl |
| LOCATION | android.hardware.gnss@1.0-impl |
| KEYMASTER | android.hardware.keymaster@3.0-impl |
| SENSORS | android.hardware.sensors@1.0-service |

Table 3: *HALWatcher* vs. Frida in resource manipulation monitoring.

| Test cases | HALWathcer | Frida |
|---|---|---|
| Ability to hook into the process which | | |
| Send SMS with normal app on rooted device | 100% | 100% |
| Send SMS with normal app on nonrooted device | 100% | 0% |
| Trojan-SMS request sendSMS on rooted device | 100% | 76% |
| Trojan-SMS request sendSMS on nonrooted device | 100% | 93% |
| Send SMS using ADB shell on rooted device | 100% | 0% |
| Send SMS using ADB shell on nonrooted device | 100% | 0% |
| Log size retrieved in the test of (#line) | | |
| Normal SMS apps | 1 | 6.8 |
| Trojan-SMS apps | 1.84 | 3.84 |

rooted and nonrooted systems to prove the limitation of Frida hooking.

The result in Table 3 shows that 100% of data can be logged using *HALWatcher*. For some samples of *Trojan-SMS*, they detect root device so that the app immediately crashes. Some malware samples use obfuscation techniques, so the *frida-agent.so* may be wrongly embedded and leads to crash the app after spawning. Obviously, *HALWatcher* performs resource monitoring better than the Frida framework.

*6.2.2. Compact and Complete Data.* In comparison with hooking techniques, *HALWatcher* is less flexible (i.e., the need for rebuilding AOSP). However, this method gives compact and complete data that can be used for real-time hardware resources manipulation and malware analysis. To observe this possibility, we compared *HALWatcher* with Frida by hooking into sensitive APIs that require the use of SMS [27]. We looked at two datasets, benign and malicious applications (e.g., FakePlayer). For *HALWatcher*, we recorded RIL requests related to SMS. Both Frida and *HALWatcher* used the same logging method, which logged only the called function's name and the timestamp of the calling. We ran and triggered the app to send SMS, then terminate the apps.

The result in Figure 3 shows that the log from *HALWatcher* is less than hooking techniques, while the testing application manipulates the same hardware resources. Listing 3 shows the example of different log sizes between *HALWatcher* and Frida on monitoring send SMS resources. The log size retrieved from hooking on the normal app is much larger than malware. We found that the normal message application has call *getSubscriptionId()* API

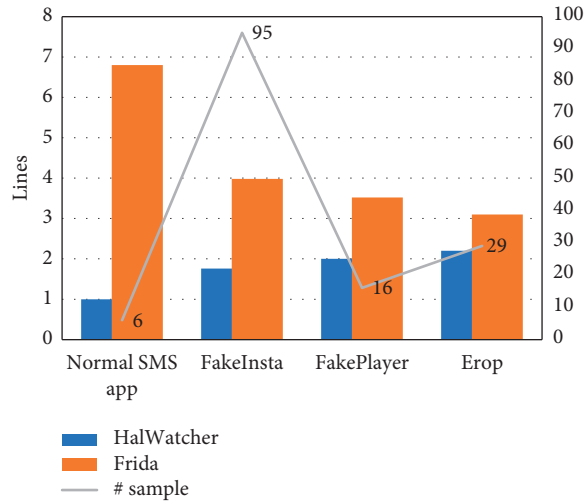



Figure 3: Average log size retrieved by *HALWatcher* and Frida.

```
(a) HALWatcher log of the default SMS app
(1)   [1590389070] [HALMonitor] RIL_REQUEST_SEND_SMS: serial 485
(b) Frida hooking log of the default SMS app
(1)   [1590389069] [android.telephony.SmsManager] [getSubscriptionId]
(2)   [1590389069] [android.telephony.SmsManager] [getSubscriptionId]
(3)   [1590389069] [android.telephony.SmsManager] [getSubscriptionId]
(4)   [1590389069] [android.telephony.SmsManager] [getSubscriptionId]
(5)   [1590389069] [android.telephony.SmsManager] [sendTextMessage]
(6)   [1590389069] [android.telephony.SmsManager] [sendMultipartTextMessage]
```

Listing 3: Example of log content of *HALWatcher* (a) and Frida (b) c. Low computational effort.

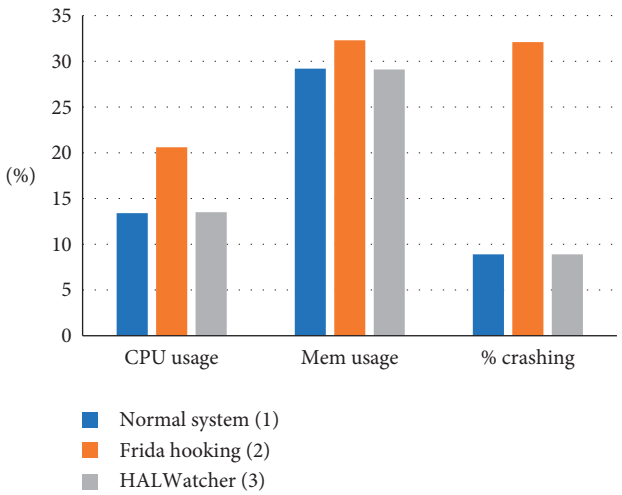



Figure 4: Computational evaluation result for (1) Normal system, (2) System with Frida hooking framework, (3) System with *HALWatcher*.

multiple times before calling *sendTextmessage()*. Moreover, the default Messaging app of AOSP calls both *sendTextmessage()* and *sendMultipartMessage()*. Note that the log is always collected from the start of opening the applications.

We compared the total CPU usage and Memory usage in three scenarios: (1) run the samples without Frida hooking and *HALWatcher*; (2) run the samples with Frida hooking; (3) run the samples with *HALWatcher*. For Frida hooking on (2), we conducted hooking progress on the target process only.

As shown in Figure 4, both CPU and Memory usage rates in (2) are more 5% higher than (1) and (3). We denote that the hooking script is injected into only the target application. However, in a practical resource monitoring system, we should implement instrumentation for all running processes. At that time, the system might be crashed (i.e., 5% extra computational resource for each process). Meanwhile, the indicators are almost no different in (1) and (3) whether the sample is malware or benign. Besides, the crashing samples rate is dramatically increased while we were running the test. This crashing happens because some samples are not suitable in AOSP 9.0; some samples are only crashed while we start Frida for hooking progress.

## 7. Conclusions and Discussion

Resource manipulation attacks are the most widespread malicious behaviors of Android malware. Current solutions, including static and dynamic analysis, are not efficient

enough to detect this attack. *HALWatcher* is a new approach that modifies Hardware Abstract Layer to monitor resources. This approach addresses the limitations of hooking techniques. *HALWatcher* provides a high coverage rate in monitoring hardware resources with low computational effort. In addition, *HALWatcher* can be applied to build a protecting mechanism in real-world devices because of the nonrooted environment requirement. However, *HALWatcher* faces some limitations. First, there is a need for strong knowledge about the Hardware Abstract Layer development to extend and deploy *HALWatcher*. Second, *HALWatcher* is only capable of monitoring hardware resources, not for others, which are already stored in system storage (e.g., CALENDAR, CONTACT, PHOTO, etc.). For our future research, we plan to research the new approach to monitoring other system resources that can integrate with *HALWatcher* to make a complete resource manipulation defending framework.

## Data Availability

The source code and log file can be found at https://github.com/josebeo2016/HALModifying.

## Disclosure

This paper is a revised version of the presented Poster: "HAL Based Resource Manipulation Monitoring on AOSP" in WISA 2020, Jeju, South Korea.

## Conflicts of Interest

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

## Acknowledgments

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (no. 2020-0-00952, Development of 5G Edge Security Technology for Ensuring 5G+ Service Stability and Availability) and supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2019-0-00477, Development of android security framework technology using virtualized trusted execution environment).

## References

[1] M. Fan, J. Liu, X. Luo et al., "Android malware familial classification and representative sample selection via frequent subgraph analysis," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 1890–1905, 2018.

[2] L. Nguyen-Vu, J. Ahn, and S. Jung, "Android fragmentation in malware detection," *Computers & Security*, vol. 87, Article ID 101573, 2019.

[3] N.-T. Chau and S. Jung, "Dynamic analysis with Android container: challenges and opportunities," *Digital Investigation*, vol. 27, pp. 38–46, 2018.

[4] A. T. Kabakus and I. A. Dogru, "An in-depth analysis of Android malware using hybrid techniques," *Digital Investigation*, vol. 24, pp. 25–33, 2018.

[5] H. Zhou, W. Zhang, F. Wei, and Y. Chen, "Analysis of android malware family characteristic based on isomorphism of sensitive API call graph," in *Proceedings of the IEEE Second International Conference on Data Science in Cyberspace*, pp. 319–327, DSC), Shenzhen, China, June 2017.

[6] J. Qiu, S. Nepal, W. Luo et al., "Data-driven android malware intelligence: a survey," in *Machine Learning for Cyber Security. Lecture Notes in Computer Science*, X. Chen, X. Huang, and J. Zhang, Eds., Springer International Publishing, Midtown Manhattan, New York, pp. 183–202, 2019.

[7] Y. Zhou and X. Jiang, "Dissecting android malware: characterization and evolution," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 95–109, NW Washington, DC; USA, 2012.

[8] S. Y. Mahmud, A. Acharya, B. Andow, W. Enck, and B. Reaves, "Cardpliance:${$PCI$}$${$DSS$}$ compliance of android applications," in *Proceedings of the 29th ${$USENIX$}$ Security Symposium (${$USENIX$}$ Security 20)*, San Diego, CA, United States, May 2020.

[9] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: behavior-based malware detection system for Android," in *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 15–26, Chicago, Illinois, USA, October 2011.

[10] L. K. Yan and H. Yin, "Droidscope: seamlessly reconstructing the ${$OS$}$ and dalvik semantic views for dynamic android malware analysis," in *Proceedings of the Presented as Part of the 21st ${$USENIX$}$ Security Symposium (${$USENIX$}$ Security 12)*, pp. 569–584, Berkeley, CA, August 2012.

[11] A. Machiry, R. Tahiliani, and M. Naik, "Dynodroid: an input generation system for android apps," in *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, pp. 224–234, ACM, Saint Petersburg, Russia, August 2013.

[12] M. K. Alzaylaee, S. Y. Yerima, and S. Sezer DL-Droid, "DL-Droid: deep learning based android malware detection using real devices," *Computers & Security*, vol. 89, Article ID 101663, 2020.

[13] A. De Lorenzo, F. Martinelli, E. Medvet, F. Mercaldo, and A. Santone, "Visualizing the outcome of dynamic analysis of Android malware with VizMal," *Journal of Information Security and Applications*, vol. 50, Article ID 102423, 2020.

[14] Y. Feng, O. Bastani, R. Martins, I. Dillig, and S. Anand, "Automated synthesis of semantic malware signatures using maximum satisfiability," 2020, https://arxiv.org/pdf/1608.06254.pdf.

[15] W. Enck, P. Gilbert, S. Han et al., "TaintDroid: an information-flow tracking system for real-time privacy monitoring on smartphones," *ACM Transactions on Computer Systems (TOCS)*, vol. 32, no. 2, p. 5, 2014.

[16] C. Zheng, S. Zhu, S. Dai et al., "Smartdroid: an automatic system for revealing ui-based trigger conditions in android applications," in *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 93–104, CA, USA, October 2012.

[17] H. Fereidooni, M. Conti, D. Yao, and A. Sperduti, "ANASTASIA: ANdroid mAlware detection using STatic analySIs of Applications," in *Proceedings of the 2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, Larnaca, Cyprus, November 2016.

[18] Z. Ma, H. Ge, Y. Liu, M. Zhao, and J. Ma, "A combination method for android malware detection based on control flow graphs and machine learning algorithms," *IEEE Access*, vol. 7, pp. 21235–21245, 2019.

[19] P. Faruki, A. Bharmal, V. Laxmi et al., "Android security: a survey of issues, malware penetration, and defenses," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 998–1022, 2015.

[20] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in *Proceedings of the 10th International Conference on Malicious and Unwanted Software (MALWARE)*, pp. 11–20, Fajardo, PR, USA, October 2015.

[21] T. Kim, B. Kang, M. Rho, S. Sezer, and E. G. Im, "A multimodal deep learning method for android malware detection using various features," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 773–788, 2018.

[22] M. Ijaz, M. H. Durad, and M. Ismail, "Static and dynamic malware analysis using machine learning," in *Proceedings of the 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 687–691, Islamabad, Pakistan, January 2019.

[23] T. Vidas, J. Tan, J. Nahata, C. L. Tan, N. Christin, and P. Tague, "A5: automated analysis of adversarial android applications," in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, pp. 39–50, Scottsdale, AZ, USA, November 2014.

[24] M. Y. Wong and D. Lie, "IntelliDroid: a targeted input generator for the dynamic analysis of android malware," *National Down Syndrome Societ*, vol. 16, pp. 21–24, 2016.

[25] J.-G. Jiang, Z.-S. Liu, M. Yu, and C. Liu, "A resource management system design for malware behavior detection," in *Proceedings of the Computer Science, Technology and Application*, pp. 467–474, World Scientific, Changsha, China, March 2016.

[26] G. Meng, R. Feng, G. Bai, K. Chen, and Y. Liu, "DroidEcho: an in-depth dissection of malicious behaviors in Android applications," *Cybersecurity*, vol. 1, no. 1, p. 4, 2018.

[27] C. Zhao, W. Zheng, L. Gong, M. Zhang, and C. Wang, "Quick and accurate android malware detection based on sensitive APIs," in *Proceedings of the 2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 143–148, Xi'an, China, August 2018.

[28] B. Soewito and A. Suwandaru, "Android sensitive data leakage prevention with rooting detection using Java function hooking," *Journal of King Saud University - Computer and Information Sciences*, 2020, Published online July 21, 2020.

[29] M. Szczepanik, M. Kędziora, and I. Jóźwiak, "Android methods hooking detection using dalvik code and dynamic reverse engineering by stack trace analysis," in *Theory and Applications of Dependable Computer Systems. Advances in Intelligent Systems and Computing*, W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, Eds., Springer International Publishing, Midtown Manhattan, New York, pp. 633–641, 2020.

Hindawi
Mobile Information Systems
Volume 2020, Article ID 8878088, 11 pages
https://doi.org/10.1155/2020/8878088

*Research Article*

# Investment Priority Analysis of ICS Information Security Resources in Smart Mobile IoT Network Environment Using the Analytic Hierarchy Process

**Jiho Shin**,[1,2] **Ilsun You**,[2] **and Jung Taek Seo**[2]

[1]*Police Science Institute, Korean National Police University, Asan, Republic of Korea*
[2]*Department of Information Security Engineering, Soonchunhyang University, Asan, Republic of Korea*

Correspondence should be addressed to Jung Taek Seo; seojt@sch.ac.kr

Received 16 July 2020; Revised 9 September 2020; Accepted 5 November 2020; Published 27 November 2020

Academic Editor: Vinod Karar



The industrial control system (ICS) inherits the attributes of the traditional information system, but because it has its own characteristics that availability of triad (CIA) of information security should be a top priority, it needs to be set differently from the traditional information security requirements. In response to the issue, TTAK.KO-12.0307 (Standard for Industrial Control System Information Security Requirements) proposed by the National Security Research Institute (NSRI) and established by the Telecommunications Technology Association (TTA) is being used. However, it is difficult to apply security requirements of TTAK.KO-12.0307 uniformly because of the reason that the characteristics of the ICS in each layer are different. There is also a limit to invest the security resources with equivalent priority for all requirements and ICS layers. It is still unresolved in the previous research studies which are related to information security resources, for example, Choi (2013), Ko et al. (2013), and Nah et al.'s (2016) studies. Therefore, this study tried to focus on what a top priority of information security requirements by the ICS in each layer is, using the analytic hierarchy process. As a result, we derived that the top priority requirement in the operation layer is "Identification Authentication Access Control," in the control layer is "Event Response," and in the field device layer is "Physical Interface Protection" with the highest importance. The results of this study can be utilized as a guideline for the security strategy and policy design by determining security requirements that should be prioritized in each layer of the ICS.

## 1. Introduction

Our society has achieved rapid industrial development based on the use of the industrial control system (ICS) in the core infrastructure such as automated processes, power generation, energy supply, transportation, and smart cities and factories [1]. ICS with closed characteristics (air-gap) from the external network that is completely different from the traditional information systems were considered relatively safe from cyberattacks and did not consider security in system design and deployment. However, in recent years, the ICS has been actively adopting IT technologies [2]. Although the digital transformation of ICSs represents the foundation for resource-efficient and flexible industrial plants, this change increases the attack surface, leading to the emergence of new threats [3]. The convergence of the ICS and the latest IT technology creates more complex problems in the security environment, and the emergence of Internet of things (IoT) technology, in particular, makes the need related security functions (e.g., key management, intrusion detection, access control, privacy protection, and wireless sensor networks security) [4, 5] more urgent [6]. IoT technologies such as beacons, for example, may have security vulnerabilities such as spoofing, DoS, and hijacking [7]. Substantial recent investment for the ICS has been directed towards the development of the ICS, that is, relies on the creation of a bridge between digital and physical environments through IoT technologies, as well the ICS itself [8]. In other words, many IoT devices are installed in the field device layer area of the ICS system and are operated based on communication with the control layer. In response, the ICS includes a smart IoT mobile environment that supports IoT-based mobility,

so secure computing should be guaranteed. If the ICS is exposed to cyber threats, serious disasters can occur throughout society. In 2010, 1,000 centrifuges were destroyed in an attack on Iran's nuclear facilities using Stuxnet, known as the first malicious code for the ICS, in which the programmable logic controller (PLC), a controller that controls field devices at nuclear facilities, was infected [9]. A lot of research studies on information security of the ICS have been invested, and a lot of efforts have been made to apply relevant security measures since the Stuxnet incident case.

It is necessary to develop and apply exclusive security requirements because the security requirements for the traditional information system are not applicable to the ICS. The biggest differences between the ICS and the information system are the purpose of cyberattacks and the priority of information security triad (CIA). In IT systems, the security is generally defined in terms of three key principles: confidentiality, integrity, and availability (also known as the CIA triad). Confidentiality focuses on ensuring assets are not disclosed to those entities who are not authorized to view it; integrity relates to protecting assets from unauthorized modifications; and availability is defined in terms of making the assets accessible to authorized entities at all permitted times [8]. Availability is known as a top priority and is also the main target of cyberattacks, as the collapse of the ICS could cause great damage. Availability is known as a top priority and is also the main target of cyberattacks, as the collapse of the ICS could cause great damage. In response to the issue, the National Security Research Institute (NSR) proposed Security Requirements for Industrial Control System by defining the features of the ICS, and it was established as a standard (TTAK.KO-12.0307) [10] by the Telecommunications Technology Association (TTA).

However, it is difficult to apply uniformly security requirements of TTAK.KO-12.0307 because the features of the ICS in each layer are different, and security resources are always not enough. In addition, it is still unresolved in the previous research studies which are related to information security resources, for example, Choi [11], Ko et al. [12], and Nah et al. [13]. Choi proposed an appropriate security assessment methodology and a checklist for the ICS, but the checklist does not provide a priority based on the characteristics of the devices; so, it is difficult to determine which areas focus more in terms of security resources. Ko et al. proposed an assessment method for measuring the security threat on smart grid based on the priority, but a limit of their study was the mean time-to-compromise (MTTC) model; they used to determine simply the number of security vulnerabilities that exist on the attack path when calculating an important weight. HoonNah and JungChan suggested the need to establish an ICS security standard same as TTAK.KO-12.0307, but there is no specific discussion of what level of security each component or layer should respond to. So, it is necessary to prioritize and apply security requirements with the standard TTAK.KO-12.0307. In particular, the ICS is a huge system divided into layers which are operated by exchanging data with each other.

Therefore, security requirements priorities should be derived and applied for each layer suitably. For this, the security requirements of TTAK.KO-12.0307 are used to analyze the priority of security requirements for each layer in this paper. Based on this, it is intended to help determine where the portion of information security resources investment should be prioritized. The results of this study provide a guideline to avoid uniform security requirements for all layers. Prioritization can be derived through the assessment of security requirements for each layer using the analytic hierarchy process method, thereby contributing to effective investment in information security resources. The results of this study are also expected to be an important contribution to IoT security and privacy protection as well as to the ICS. To discuss this, ICS security and prior research are discussed in Section 2, and the design of the research model to be used for priority analysis using AHP is discussed in Section 3, and empirical analysis conducted based on this is discussed in Section 4. The implications of the analysis results are discussed in Section 5 and finally concluded in Section 6.

## 2. Background

*2.1. Information Security of ICS.* The ICS basically inherits many attributes of the traditional IT system. However, in order to derive an information security investment priority, we need to look at a variety of different aspects from the traditional IT system [6]. First, in hardware and software aspects, the IT system operates on a short-term replacement cycle, but the ICS has at least 15 years of long-term replacement cycles generally. The IT system also uses universal operating systems (general-purpose) such as Windows and Linux, but the ICS uses exclusive operating systems. In addition, maintenance and repair, such as system patches, on the ICS are more difficult than traditional IT systems. At last, in network performance aspect, the IT system focuses on overall performance, such as the reliability of responses is important and tolerability exists for some communication delays, but the ICS focuses on real-time responsiveness and is inflexible for communication delays. For risk management objectives, the ICS does not allow the control device to be shut down, and system availability is very important, but the integrity of the data is more important, and some failures can be allowed in the IT system. As a result, the IT system can end up with relatively minor economic damage, such as inconvenience or delay, due to cyberattacks or incidents caused by its own defects. However, the ICS could immediately halt operations at industrial sites, leading to human casualties and massive disasters, which could result in huge social and economic damages. This means that among the CIA triad of information security, the traditional IT system should prioritize "Confidentiality" and "Integrity," while the ICS should prioritize "Availability."

These characteristics set the cyberattacker's goals differently. While cyberattacks on the traditional IT system were primarily aimed at leaking classified information, attacks targeting on the ICS are mainly focused on operational paralysis. This is because stopping the ICS will cause great

damage. In the 2010 Stuxnet case, the attack was carried out by infecting Siemens PLC to paralyze operations by manipulating the number of rotations of connected centrifuges, and the main objective in subsequent series of major cyberattacks against the ICS was to disrupt normal operations.

*2.2. Literature Review.* The past ICS was recognized as safe by configuring an independent network, but the vulnerability was revealed in a bypass attack by the malicious code. In order to respond, HoonNah and JungChan insisted that comprehensive and systematic security measures are needed to defend themselves in depth from cyberattacks and specifically suggested the need to establish standards for security of the ICS. Particularly important is that they took the same argument as this paper, judging that it is unrealistic to take measures at an equal security level for all vulnerabilities [13]. However, there is a limit to driving their arguments because there is no specific discussion of what level of security each component or layer should respond to.

Since the ICS operates in various environments, including major national infrastructure and social overhead capital facilities, the security assessment and security resource investment are of great importance. Therefore, the security assessment of the ICS should be carried out using an objective and feasible inspection process. Choi [11] proposed an appropriate security assessment methodology and checklist for the ICS, taking into account the characteristics of the ICS environment, devices, and operation methods. However, its usefulness could not be verified because there were no examples to verify the proposed methodology, and moreover, the proposed methodology does not provide a checklist that should be prioritized based on the characteristics of the devices; so, it is difficult to determine which areas to focus more on security resources.

Ko et al. proposed an assessment method for measuring the security threat on smart grid [12]. In particular, the ICS network has a hierarchical structure, and security sensitivity of the produced data by each layer is different; so, they suggested that it is necessary to make a level as layers with similar data sensitivity into one area. And they used these levels (consumer level, advanced metering infrastructure head end level, and control center level) to prioritize what needs to be protected in that network. They explained that if protection is relatively unnecessary or if it is difficult for an attacker to access for attack, they can increase efficiency by excluding it from the vulnerability target. Their research can be seen as a previous study of the need investment priorities of information security resource for the ICS to be discussed in this paper. They used a quantified network model to assess security threats applied to advanced metering infrastructure and validated the security threat assessment for the proposed model using mean-time-to-compromise (MTTC) proposed by Leversage and Byres [14] for the resulting attack scenario. However, there is a limitation that the evaluation method using MTTC does not evaluate the overall security threat to the ICS. This is because MTTC simply determines that the number of security vulnerabilities that exist on the attack path is an important weight.

As such, many methodologies for security assessment are important to effectively respond to security threats for the ICS. Although many studies have been conducted, it is difficult to find a discussion that the security assessment uses the information security standard for the ICS. This is because there has been no definition of specific information security requirements to the ICS. It is also understood that although there are already established ICS information security requirements, there is a lack of discussion on the methodology for applying them to each ICS. Therefore, in this paper, we want to provide guidelines for efficient investment of security resources by analyzing the priorities of each layer when using TTAK.KO-12.0307.

## 3. Design of Analysis

*3.1. Analytic Hierarchy Process Methodology.* In this study, the analytical hierarchy process (AHP) method was used to analyze the investment priority of information security resources in the ICS. The AHP was developed by Tomas in the 1970s as part of the decision-making method through the multiple assessment criteria for multiple alternatives [15]. In general, decision-making problems should be solved by choosing the optimal alternative under multiple criteria, and many existing decision-making problems have been solved using statistical models under controlled assumptions [16]. In addition, decision-making problems often include qualitative criteria, which led to the need to quantify criteria with subjective values [16]. In other words, many other real-world problems involve the need to combine quantitative measures with qualitative concerns [17]. This has the problem of prioritizing ICS information security requirements, depending on the responder with different levels of awareness and expertise of information security. In particular, since a big part of information security relates to qualitative and nonfinancial concerns, traditional economic approaches are severely constrained [17]. Saaty developed the AHP to analyze multicriteria decision problems involving both quantitative and qualitative criteria [17–19]. AHP methodology uses the concept of hierarchy to lay out the different elements (purpose, alternatives, and factors) needed to make decisions, thereby providing a more detailed and logical view of the relationship between the different elements [20]. The AHP methodology for performing pairwise comparisons between elements of each layer has been widely used in multidecision-making problems, with two typical advantages: first, weighting between assessment elements can be determined through systematic quantitative procedures. In addition, the choice of optimal alternatives has the advantage of being easier to understand than conventional statistical decisions and being able to use the subjective and objective information of experts comprehensively. Second, it provides indicators to determine the consistency of decision makers (experts). And the analysis procedures are consistent with reasonable decision-making procedures [21].

*3.2. Analysis Model Design.* In order to analyze the relative investments priorities of information security resource in the ICS, this study has established assessment criteria based on the

classification divided in ICS information security requirements (TTAK.KO-12.0307) of TTA. However, the method of prioritizing information security investment for the entire ICS has a wide range of coverage, and there is ambiguity in the selection of priorities. Therefore, it is desirable to perform an analysis of the investment priorities of information security resources by classifying the ICS into each layer.

There are several definitions for layers in the ICS. Irfan Ahmed et al. suggested that information security of ICS/supervisory control and data acquisition (SCADA) should be classified into six layers, based on connectivity between components in the system and connectivity between other networks, such as the system network and the Internet [22]. However, it is difficult to use it as an analysis model because it does not include field devices such as sensors and actuators, and wired and wireless devices are not considered.

On the other hand, TTAK.KO-12.0307 presents the "Security Reference Model" to define the information security requirements of the ICS and is divided into 3 layers which consisted of the "Operation Layer," "Control Layer," and "Field Device Layer" (Figure 1). The "Operation Layer" uses the data received from the control layer to monitor the status of the field devices or send control commands, including engineering workstation (EWS) and human-machine interface (HMI) [10]. The "Control Layer" is responsible for transferring the measured and collected data from the field devices to the operation layer. And the layer is also responsible for controlling the field devices with command from the operation layer, including the PLC, distributed control system (DCS), and remote terminal unit (RTU) [10]. The "Field Device Layer" includes a field device used to measure, collect, and control status data, such as sensors and actuators, and the field device is connected to the control layer by wired and wireless networks or by serial cables [10]. The priority assessment criteria of this paper are based on the classification of TTAK.KO-12.0307.

However, some of information security requirements of TTAK.KO-12.0307 were merged because there were many assessment criteria to be used as the AHP method. Then "Identification Certification" and "Access Control" were merged among security functions on hierarchy I, and "Transmission Data Protection" and "Stored Data Protection" were merged in the same way. Finally, the analysis model to be used for priority assessment is shown in Figure 2.

TTAK.KO-12.0307 set different assessment criteria for the operation layer and control/field device layer. So, there were also two models for investment priority of information security resource analysis. Figure 2 was used in the operation layer, and Figure 3 was used in the control layer and field device layer.

*3.3. Analysis Criteria.* Table 1 shows the assessment criteria and its descriptions in TTAK.KO-12.0307.

## 4. Empirical Analysis

*4.1. Analysis Method and Tool.* The analysis of this study uses the AHP, a hierarchical decision analysis method, but also provides a description of each assessment criteria to help the survey respondents understand. In the AHP analysis method, it is very important to ensure objectivity and expertise in response. The AHP survey was conducted by selecting researchers, practical experts, and a professor related to the ICS, cyber physics system (CPS), and SCADA system. They are affiliated in National Security Research Institute, Electronics and Telecommunications Research Institute (these 2 are government-based research institutes), Incheon International Airport, Naonwork, OnSecurity, Coontec (these 4 are corporations related on ICS information security), and Ajou University.

The assessment criteria were based on TTAK.KO-12.0307 as described above, but there is only one assessment criterion in the network robustness section of the "Operation Layer;" so, the pairwise comparison was not conducted (Figure 2). In addition, as discussed above, "Identification·Authentication" and "Access Control," which are classified as the security functions section, were set by merging into "Identification·Authentication·Access Control" due to similarity in content. In the same way, "Transmission Data Protection" and "Stored Data Protection" were also set by merging into "Data Protection." Finally, the survey was conducted by setting up 3 assessment criteria in hierarchy I and 10 assessment criteria in hierarchy II (but 8 assessment criteria for the "Operating Layer") (Table 1).

*4.2. Verification Consistency of Survey Responses.* The AHP survey of this study was conducted for a month from December 2019 to January 2020 and was conducted on industry, academia, and research experts related to information security of the ICS. There are a few of discussions regarding the appropriate sample size in order to carry out the AHP analysis. Melillo and Pecchia insisted that smaller sample size is required in case of equally important alternative [23]. The reliability of AHP results is more relevant to the respondents' expertise rather than the number of response samples. In this study, the experts responded to the survey in the field of information security of the ICS with at least more than five years of related experience. A total of 19 experts were surveyed, and 19 responses were collected. AHP analysis of response data used the DRESS tool.

The AHP analysis method determines consistency index (CI) of the response to ensure reliability of the analysis results. Due to the characteristic of the pairwise comparison, the lower the CI, the more consistent it is, which is related to the respondents' expertise. Generally, responses with a CI value of 10% or less are considered consistent. In this study, 4 surveys with a CI value of 0.1 or higher were excluded from the results analysis, so only 15 responses were used for the analysis.

## 5. AHP Analysis Result

In this study, the results of AHP analysis were divided into the "Operation Layer," "Control Layer," and "Field Device Layer" for the investment priority of information security resources in the ICS.

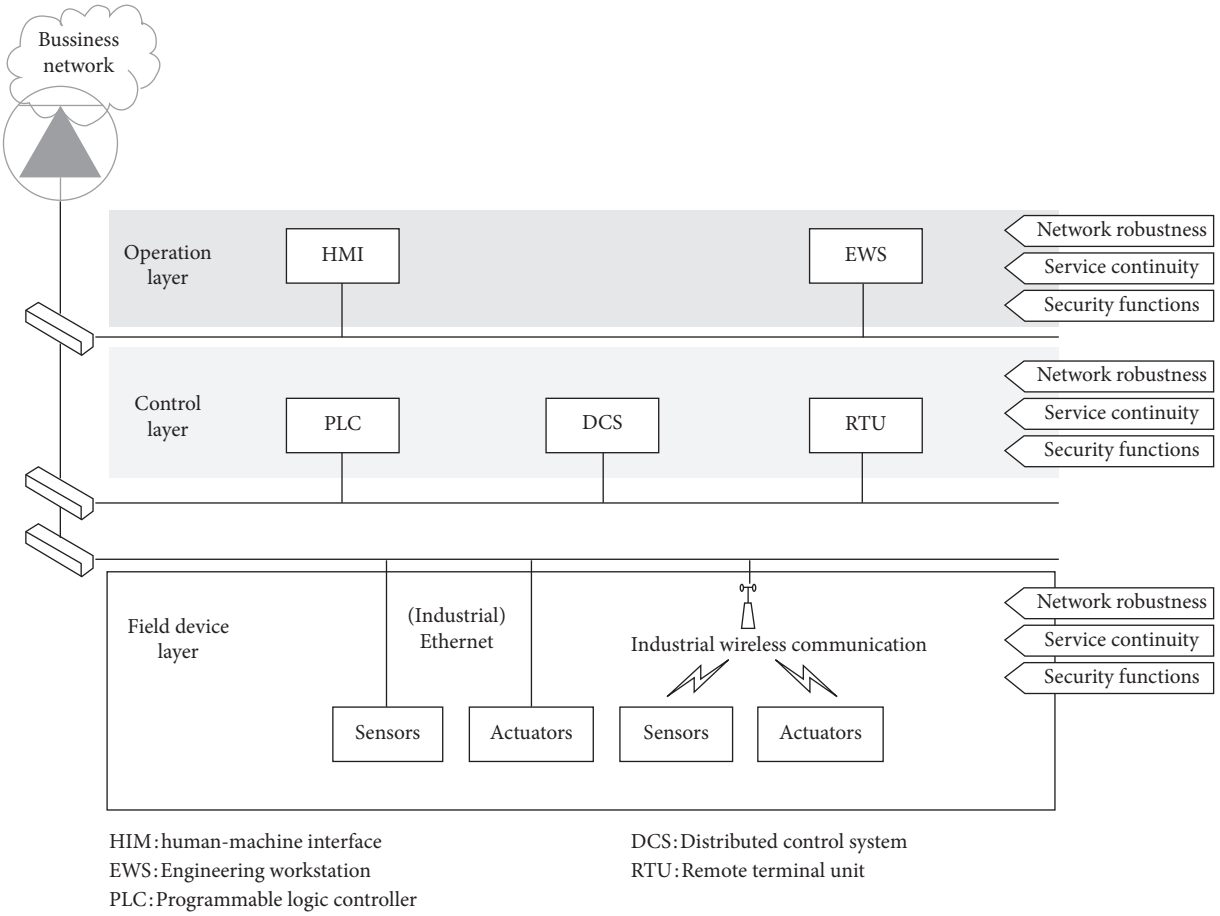



Figure 1: Security reference model of TTAK.KO-12.0307 [13].

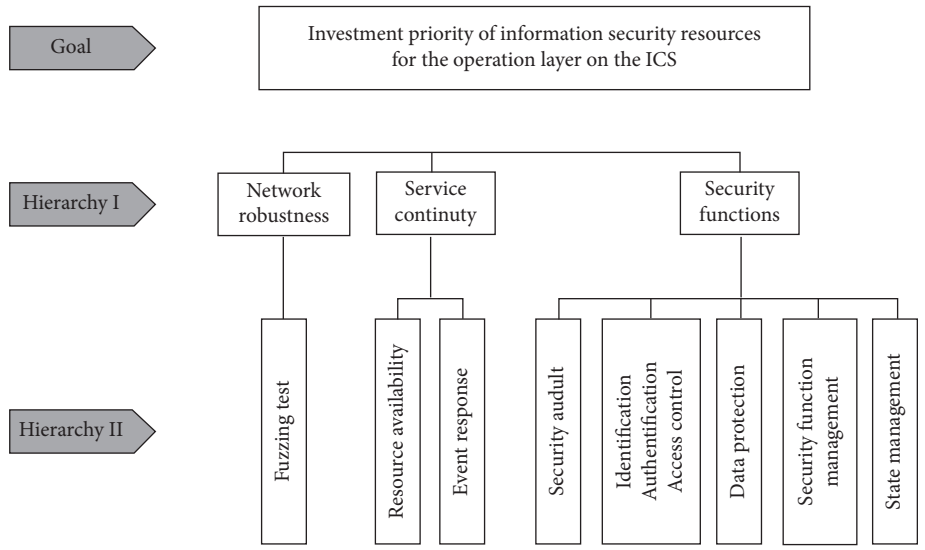



Figure 2: Investment priority of information security resources for the "Operation Layer" on the ICS.

*5.1. Operation Layer.* The AHP results for the analysis of investment priority of information security resource by the "Operation Layer" of the ICS are as follows.

An analysis result of the priority on hierarchy I showed that "Security Functions" was the highest priority with an importance 0.371, "Service Continuity" was the second priority with an importance 0.358, and "Network Robustness" was the third priority with an importance 0.271 (Table 2).

In "Security Functions" section, which was ranked the highest priority in hierarchy I, "Identification·Authentication·Access Control" was the highest priority with an importance 0.291, "Security Function Management" was the second priority

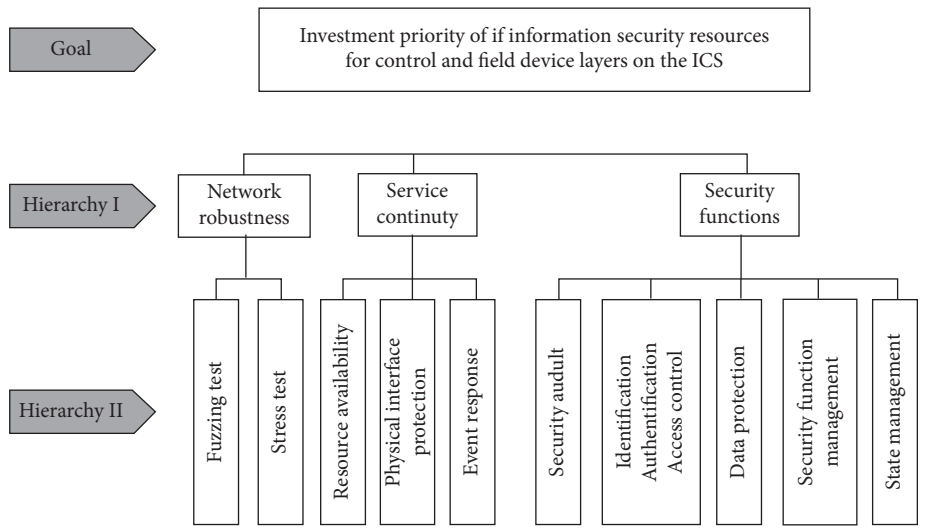



Figure 3: Investment priority of information security resources for the "operation layer" on the ICS.

Table 1: Criteria and descriptions of ICS security requirements.

| | Hierarchy I | | Hierarchy II | |
|---|---|---|---|---|
| | Criteria | Description | Criteria | Description |
| ICS security requirements | Network robustness | Require network robustness against external cyberattacks or internal abnormal behavior | Fuzzing test | Require handling capability to sustain the ICS service when receiving abnormal network packet |
| | | | Stress test | Require providing ICS service even when overloading the network traffic |
| | | | Resource availability | Require resource management procedures, such as backup and recovery, so that resources can perform their normal functions |
| | Service continuity | Require stable and continuous service | Physical interface protection | Require resource management procedures, such as backup and recovery, so that resources can perform their normal functions |
| | | | Event response | Require checking the status of devices, systems, and networks in real-time and responding to failures |
| | Security functions | Require security features such as component identification, authentication, and access control | Security audit | Require security audits through creating and encrypting audit-logs for major events |
| | | | Identification, authentication, and access control | Require separation or restriction about identification and access authority of devices/users with a user authentication procedure |
| | | | Data protection | Require confidentiality and integrity of sensitive transmission or stored data |
| | | | Security functions management | Require network and security settings of the control software, secure encryption algorithms, and key management |
| | | | State management | Require state management such as integrity verification of the execution code, normal operation test, and vulnerability response |

with an importance 0.195, and "Data Protection" was the third priority with an importance 0.194, followed by "State Management" and "Security Audit" in order (Table 3). In "Service Continuity" section, which was ranked the second priority in hierarchy I, "Event Response" was the highest priority with an importance 0.534 and "Resource Availability" was the second priority with an importance 0.466. In "Network Robustness" section, which was ranked the third priority in hierarchy I, there is only one assessment criterion, which is the "Fuzzing Test" in the sector, so the pairwise comparison survey was not

Table 2: AHP result of hierarchy I on all layers.

| Layer | Operation layer | | Control layer | | Field device layer | |
|---|---|---|---|---|---|---|
| Hierarchy I | Importance | Priority | Importance | Priority | Importance | Priority |
| Network robustness | 0.271 | 3 | 0.281 | 2 | 0.258 | 3 |
| Service continuity | 0.358 | 2 | **0.439** | **1** | **0.463** | **1** |
| Security functions | **0.371** | **1** | 0.280 | 3 | 0.279 | 2 |
| Consistency index | 0.02 | | 0.02 | | 0.03 | |

The highest priority of each layer is shown in bold.

Table 3: AHP result of hierarchy II on the operation layer.

| Hierarchy I | Hierarchy II | Importance | Priority | C.I. |
|---|---|---|---|---|
| Network robustness | Fuzzing test | — | **1** | — |
| Service continuity | Resource availability | 0.466 | 2 | 0.00 |
| | Event response | **0.534** | **1** | |
| | Security audit | 0.153 | 5 | |
| Security functions | Identification authentication access control | **0.291** | **1** | 0.07 |
| | Data protection | 0.194 | 3 | |
| | Security function management | 0.195 | 2 | |
| | State management | 0.168 | 4 | |

The highest priority of each hierarchy on operation layer is shown in bold.

conducted, but the importance can be very high. Because TTAK.KO-12.0307 security requirements require network robustness even in the following cases through the "Fuzzing Test." (1) In case of the order of the field in packets is changed, (2) in case of a part of the field in packets is cut, (3) in case of the field size in packets is different, (4) in case of the fixed value of the field in packets is different, and (5) in case of the value of the field in packets is not within the valid range [10].

As a result of the priority pairwise comparison of all criteria in the "Operation Layer" of the ICS, "Identification·Authentication·Access Control" was the highest priority with an importance 0.171, "Event Response" was the second priority with an importance 0.168, and "Resource Availability" was the third priority with an importance 0.122, followed by "Security Function Management," "State Management," "Data Protection," "Fuzzing Test," and "Security Audit" in order (Table 4).

*5.2. Control Layer.* The AHP results for the analysis of investment priority of information security resource by the "Control Layer" of the ICS are as follows.

An analysis result of the priority on hierarchy I showed that "Service Continuity" was the highest priority with an importance 0.439, "Network Robustness" was the second priority with an importance 0.281, and "Security Functions" was the third priority with an importance 0.280 (Table 2).

In "Service Continuity" section, which was ranked the highest priority in hierarchy I, "Physical Interface Protection" was the highest priority with an importance 0.362, "Resource Availability" was the second priority with an importance 0.336, and "Event Response" was the third priority with an importance 0.302 (Table 5). In "Network Robustness" section, which was ranked the second priority in hierarchy I, the "Fuzzing Test" was the highest priority with an importance 0.510, and the "Stress Test" was the second priority with an importance 0.490. In "Security Functions" section, which was ranked the third priority in hierarchy I, "Identification·Authentication·Access Control" was the highest priority with an importance 0.256, "State Management" was the second priority with an importance 0.215, and "Security Function Management" was the third priority with an importance 0.203, followed by "Data Protection" and "Security Audit" in order.

As a result of the priority pairwise comparison of all criteria in the "Control Layer" of the ICS, "Event Response" was the highest priority with an importance 0.128, "Resource Availability" was the second priority with an importance 0.122, and "Identification·Authentication·Access Control" was the third priority with an importance 0.119, followed by the "State Management," "Physical Interface Protection," "Security Function Management," "Data Protection," "Stress Test," "Security Audit," and "Fuzzing Test" in order (Table 6).

*5.3. Field Device Layer.* The AHP results for the analysis of investment priority of information security resource by the "Field Device Layer" of the ICS are as follows.

An analysis result of the priority on hierarchy I showed that "Service Continuity" was the highest priority with an importance of 0.463, "Security Functions" was the second priority with an importance 0.279, and "Network Robustness" was the third priority with an importance 0.258 (Table 2).

In "Service Continuity" section, which was ranked the highest priority in hierarchy I, "Physical Interface Protection" was the highest priority with an importance 0.375, "Event Response" was the second priority with an importance 0.333, and "Resource Availability" was the third priority with an importance 0.292 (Table 7). In "Security Functions" section, which was ranked the second priority in hierarchy I, "State Management" was the highest priority

Table 4: Final priorities among all criteria on the operation layer.

| Hierarchy I | Hierarchy II | Importance | Priority | C.I. |
|---|---|---|---|---|
| Network robustness | Fuzzing test | 0.095 | 7 | |
| Service continuity | Resource availability | 0.122 | 3 | 0.75 |
| | Event response | 0.168 | 2 | |
| Security functions | Security audit | 0.094 | 8 | |
| | Identification authentication access control | **0.171** | **1** | |
| | Data protection | 0.113 | 6 | |
| | Security function management | 0.120 | 4 | |
| | State management | 0.118 | 5 | |

The highest priority among all criteria on the operation layer is shown in bold.

Table 5: AHP result of hierarchy II on the control layer.

| Hierarchy I | Hierarchy II | Importance | Priority | C.I. |
|---|---|---|---|---|
| Network robustness | Fuzzing test | **0.510** | **1** | 0.00 |
| | Stress test | 0.490 | 2 | |
| Service continuity | Resource availability | 0.336 | 2 | |
| | Physical interface protection | **0.362** | **1** | 0.02 |
| | Event response | 0.302 | 3 | |
| Security functions | Security audit | 0.123 | 5 | |
| | Identification authentication access control | **0.256** | **1** | |
| | Data protection | 0.203 | 4 | 0.04 |
| | Security function management | 0.203 | 3 | |
| | State management | 0.215 | 2 | |

The highest priority of each hierarchy on the control layer is shown in bold.

Table 6: AHP result of hierarchy II on the control layer.

| Hierarchy I | Hierarchy II | Importance | Priority | C.I. |
|---|---|---|---|---|
| Network robustness | Fuzzing test | 0.061 | 10 | |
| | Stress test | 0.080 | 8 | |
| Service continuity | Resource availability | 0.122 | 2 | |
| | Physical interface protection | 0.103 | 5 | 0.08 |
| | Event response | **0.128** | **1** | |
| Security functions | Security audit | 0.078 | 9 | |
| | Identification authentication access control | 0.119 | 3 | |
| | Data protection | 0.097 | 7 | |
| | Security function management | 0.101 | 6 | |
| | State management | 0.112 | 4 | |

The highest priority among all criteria on the control layer is shown in bold.

Table 7: AHP result of hierarchy II on the field device layer.

| Hierarchy I | Hierarchy II | Importance | Priority | C.I. |
|---|---|---|---|---|
| Network robustness | Fuzzing test | 0.473 | 2 | 0.00 |
| | Stress test | **0.527** | **1** | |
| Service continuity | Resource availability | 0.292 | 3 | |
| | Physical interface protection | **0.375** | **1** | 0.02 |
| | Event response | 0.333 | 2 | |
| Security functions | Security audit | 0.115 | 5 | |
| | Identification authentication access control | 0.243 | 2 | |
| | Data protection | 0.219 | 3 | 0.05 |
| | Security function management | 0.167 | 4 | |
| | State management | **0.256** | **1** | |

The highest priority of each hierarchy on the field device layer is shown in bold.

with an importance 0.256, and the "Identification·Authentication·Access Control" was the second priority with an importance 0.490, followed by "Security Function Management" and "Security Audit" in order. In "Network Robustness" section, which was ranked the third priority in hierarchy I, "Network Robustness" was the

Table 8: AHP result of hierarchy II on the field device layer.

| Hierarchy I | Hierarchy II | Importance | Priority | C.I. |
|---|---|---|---|---|
| Network robustness | Fuzzing test | 0.074 | 9 | |
| | Stress test | 0.081 | 8 | |
| Service continuity | Resource availability | 0.110 | 3 | 0.07 |
| | Physical interface protection | **0.159** | **1** | |
| | Event response | 0.105 | 5 | |
| Security functions | Security audit | 0.069 | 10 | |
| | Identification authentication access control | 0.118 | 2 | |
| | Data protection | 0.105 | 4 | |
| | Security function management | 0.087 | 7 | |
| | State management | 0.090 | 6 | |

The highest priority among all criteria on the field device layer is shown in bold.

Table 9: Implication of the AHP analysis result on the ICS in each layer.

| Considerations | Layers | Operation layer | Control layer | Field device layer |
|---|---|---|---|---|
| Practical environments | Security aspect | A lot of user accesses that need to be identified for security | Various events of devices, systems, and networks that need to be handled for service continuity | Control end-point devices along with ethernet or the IoT network |
| | Risks | Social engineering attacks or user carelessness | Service abort or collapse availability | Manipulating command attack to end-point devices |
| Top priority for security resources investment | Hierarchy I | Security functions | Service continuity | Service continuity |
| | Hierarchy II | Identification Authentication Access control | Event response | Physical interface protection |
| TTAK.KO-12.0307 standard | | Based on Priority analysis | | Analytic hierarchy process |

highest priority with an importance 0.527, and the "Fuzzing Test" was the second priority with an importance 0.473.

As a result of the priority pairwise comparison of all criteria in the "Field Device Layer" of the ICS, "Physical Interface Protection" was the highest priority with an importance 0.159, "Identification·Authentication·Access Control" was the second priority with an importance 0.118, and "Resource Availability" was the third priority with an importance 0.110, followed by the "Data Protection," "Data Protection," "State Management," "Security Function Management," "Stress Test," "Fuzzing Test," and "Security Audit" in order (Table 8).

*5.4. Implications.* It is difficult to deploy effective resources in applying the uniform security requirements because the ICS has a wide range of areas and, above all, different characteristics of each layer. In this study, it was intended to avoid applying the uniform security requirements for ICS and to contribute to the effective investment in information security resources by deriving the priority of security requirements for each layer on the ICS.

As a result of analyzing the priority of assessment criteria for each layer using AHP, "Identification Authentication Access Control" was the most important security requirement that should be prioritized on the "Operation Layer." This emphasizes that these criteria are the most important to prepare for information security from the risks of social engineering attacks or exposure due to user carelessness, mainly because the operation layer has a lot of user access. "Event Response" was the most important security requirement that should be prioritized on the "Control Layer." This emphasizes the need for various events in the control layer to be properly handled in order for the service to continue to operate. Because "Event Response" is an item that requires real-time identification of the status of devices, systems, and networks and is responsive in the event of various failures. "Physical Interface Protection" was the most important security requirement that should be prioritized on the "Field Device Layer." The "Field Device Layer" has a variety of devices, including sensors and actuators, and is an important layer of control over end-point devices using industrial ethernet or wireless IoT networks, requiring a high-level protection from the physical interface accessible to this layer (Table 9).

On the contrary, it is also necessary to point out the commonly lowest assessment criteria for investment priority of information security resources for the ICS. "Security Audit" was analyzed with the lowest importance in the "Operation Layer" and "Field Device Layer." In terms of investment of information security resources, "Security Audit" performs audits by creating audit log for major events and encrypting the log data, mainly as part of long-term security functions rather than real-time response or service continuity, so "Security Audit" can be analyzed as

Table 10: Comparison of the approaches for the ICS information security.

| Parameters | Approaches | | | | |
|---|---|---|---|---|---|
| | Choi [12] | Ko et al. [13] | Hoon Nah and Na [11] | TTAK.KO-12.0307 [10] | Proposed approach |
| Proposed exclusive security requirements | No | Yes | Partially yes | Yes | Yes |
| Utilization of a standard | No | No | Yes | Yes | Yes |
| Concept of ICS layers | Not considered | 3 levels (according to data sensitivity) | Not considered | 3 layers (security reference model) | 3 layers (security reference model) |
| Security requirements priority analysis | Yes | Yes | Not considered | Not considered | Yes |
| Security resource investment decision | Using the risk evaluation checklist | Using the number of vulnerabilities in the attack path | Not considered | Not considered | Using the priorities |
| Usability as a guidelines | Partially possible | Partially possible | Partially possible | Impossible | Possible |

relatively low importance due to the availability aspect in the ICS.

In summary, we have successfully derived security investment priorities using AHP techniques and TTAK.KO-12.0307 standards for ICS information security priorities that have not been addressed in our previous research. The biggest advantage of this result is that it can be used as a guideline when establishing ICS security policies. In addition, the results of this study contribute significantly to the effective distribution of information security resources that were not addressed in previous studies (Table 10).

## 6. Conclusion and Further Research

The ICS inherits the attributes of the traditional information system, but because it has its own characteristics such as availability and continuity, it needs to be set differently from the information security requirements of the traditional information system. For appropriate information security requirements and assessment on the ICS, TTAK.KO-12.0307 proposed by the NSR and established by the TTA is being used.

In this study, the priorities of assessment criteria by hierarchy were analyzed to enhance the efficiency of investment in information security resources on the ICS. There are many difficulties in operating an industrial control system to establish security policies for all the requirements set forth in the standards. Therefore, the results of this study can be used to design security strategies and policies by selecting the security elements that should be relatively prioritized for each layer in the operation of the industrial control system. It can also be used as a guideline for determining the investment priority of security resources to the ICS that are currently in operation or are being redesigned. However, in the course of carrying out this study, experts who responded to the survey commented on whether TTAK.KO-12.0307 standard, which was used as assessment criteria, was suitable for the security requirements, so it will remain a future research.

## Data Availability

The data used to support the findings of the study are available at Security Requirements for Industrial Control System (TTAK.KO-12.0307) and Telecommunication Technology Association http://www.tta.or.kr/data/weeklyNoticeView.jsp?pk_num=5621.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2018-0-01799) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation) and supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT: Ministry of Science and ICT) (NRF-2020R1A2C1012187).

## Supplementary Materials

Supplementary figures of the survey sheet are provided as a separate file under the Supplementary Materials section (Figures 1–3). (*Supplementary Materials*)

## References

[1] S. Keith, V. Pillitteri, and S. Lightman, *Guide to Industrial Control Systems (ICS) Security*, NIST Special Publication, National Institute of Standard and Technology, Gaithersburg, MD, USA, 2015.

[2] J.-H. Lee and W.-N. Kim, "Security requirements for industrial control system," *Telecommunication Technology Association*, vol. 173, pp. 62–66, 2017.

[3] M. Eckhart, B. Brenner, and A. Ekelhart, "Quantitative security risk assessment for industrial control systems: research opportunities and challenges," *Journal of Internet Service and Information Security*, vol. 9, no. 3, pp. 52–73, 2019.

[4] H. Hui, X. An, and H. Wangetal, "Survey on blockchain for internet of things," *Journalof Internet Service-andInformationSecurity*, vol. 9, no. 2, pp. 1–30, 2019.

[5] V. Korzhuk, A. Groznykh, and M. Alexander, "Identification of attacks against wireless sensor networks based on

behaviour analysis," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 10, no. 2, pp. 1–21, 2019.

[6] M. StJohn-Green, R. Piggin, and J. A. McDermid, "Combined security and safety risk assessment—what needs to be done for ICS and the IoT," in *Proceedings of the 10th IET System Safety and Cyber-Security Conference*, pp. 1–7, Bristol, UK, 2015.

[7] H. K. Almathami, A. Majed, and E. Vlahu-Gjorgievska, "An analytical approach to using and implementing beacons: opportunities and challenges," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 10, no. 1, pp. 57–74, 2019.

[8] N. Tuptuk and S. Hailes, "Security of smart manufacturing systems," *Journal of Manufacturing Systems*, vol. 47, pp. 93–106, 2018.

[9] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," in *Proceedings of the IECON 2011—37th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4490–4494, Melbourne, Australia, November 2011.

[10] TTA, Security Requirements for Industrial Control System, *TTAK.KO-12.0307, Telecommunication Technology Association*, 2015, http://www.tta.or.kr/data/weeklyNoticeView.jsp?pk_num=5621.

[11] M. Choi, "A study on security evaluation methodology for industrial control systems," *Journal of the Korea Institute of Information Security and Cryptology*, vol. 23, no. 2, pp. 287–298, 2013.

[12] J. Ko, S. Lee, and T. Shon, "Security threat evaluation for smartgrid control system," *Journal of the Korea Institute of Information Security and Cryptology*, vol. 23, no. 5, pp. 873–883, 2013.

[13] J. HoonNah and N. JungChan, "Industrial control system security standardization trend," *Review of the Korea Institute of Information Security & Cryptology*, vol. 26, no. 4, pp. 28–35, 2016.

[14] D. J. Leversage and E. J. Byres, "Estimating a system's mean time-to-compromise," *IEEE Security & Privacy*, vol. 6, no. 1, pp. 52–60, 2008.

[15] L. S. Thomas, "Decision making with the analytic hierarchy process," *International Journal of Services Sciences*, vol. 1, no. 1, pp. 83–98, 2008.

[16] J. Hyo-Jung, "Analysis on the information security manpower policy with analytic hierarchy process," in *Proceedings of the Symposium of the Korean Institute of communications and Information Sciences*, pp. 468–471, Seoul, Republic of Korea, 2003.

[17] L. D. Bodin, L. A. Gordon, and M. P. Loeb, "Evaluating information security investments using the analytic hierarchy process," *Communications of the ACM*, vol. 48, no. 2, pp. 78–83, 2005.

[18] L. S. Thomas, "A scaling method for priorities in hierarchical structures," *Journal of Mathematical Psychology*, vol. 15, no. 3, pp. 234–281, 1977.

[19] L. S. Thomas, *The Analytic Hierarchy Process*, McGraw-Hill, New York, NY, USA, 1980.

[20] T.-S. Kim, "Analysis on information security manpower policy by the analytic hierarchy process," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 31, pp. 486–493, 2006.

[21] W. Sung, "A study on information security policy priority using AHP (analytic hierarchy process)," in *Proceedings of the Symposium of the Korean Association for Public Administration*, pp. 1614–1634, Seoul, Republic of Korea, October 2011.

[22] I. Ahmed, S. Obermeier, M. Naedele, and G. G. Richard III, "Scada systems: challenges for forensic investigators," *Computer*, vol. 45, no. 12, pp. 44–51, 2012.

[23] P. Melillo and L. Pecchia, "What is the appropriate sample size to run analytic hierarchy process in a survey-based research?" in *Proceedings of the International Symposium of the Analytic Hierarchy Process*, pp. 4–7, London, UK, August 2016.

Hindawi
Mobile Information Systems
Volume 2020, Article ID 8885269, 13 pages
https://doi.org/10.1155/2020/8885269

*Research Article*

# Facilitating User Authorization from Imbalanced Data Logs of Credit Cards Using Artificial Intelligence

**Vinay Arora**,[1] **Rohan Singh Leekha**,[2] **Kyungroul Lee**,[3] **and Aman Kataria**[4]

[1]*Computer Science & Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab, India*
[2]*Associate Application Support, IT-App Development/Maintenance, Concentrix, Gurugram, India*
[3]*School of Computer Software, Daegu Catholic University, Gyeongsan, Republic of Korea*
[4]*Optical Devices and Systems (Visiting Research Scholar), CSIR-CSIO, Chandigarh, India*

Correspondence should be addressed to Kyungroul Lee; lisa.sch.k@gmail.com

Received 14 July 2020; Revised 9 September 2020; Accepted 29 September 2020; Published 30 October 2020

Academic Editor: Zengpeng Li



An effective machine learning implementation means that artificial intelligence has tremendous potential to help and automate financial threat assessment for commercial firms and credit agencies. The scope of this study is to build a predictive framework to help the credit bureau by modelling/assessing the credit card delinquency risk. Machine learning enables risk assessment by predicting deception in large imbalanced data by classifying the transaction as normal or fraudster. In case of fraud transaction, an alert can be sent to the related financial organization that can suspend the release of payment for particular transaction. Of all the machine learning models such as RUSBoost, decision tree, logistic regression, multilayer perceptron, *K*-nearest neighbor, random forest, and support vector machine, the overall predictive performance of customized RUSBoost is the most impressive. The evaluation metrics used in the experimentation are sensitivity, specificity, precision, *F* scores, and area under receiver operating characteristic and precision recall curves. Datasets used for training and testing of the models have been taken from kaggle.com.

## 1. Introduction

For this study, the term "credit" refers to a method of e-commerce without having funds. A credit card is a thin, rectangular metal or plastic block provided by the banking institution, allowing card users to borrow cash to pay for products and services. Credit cards enforce cardholders to repay the financial leverage, interest payment, and any other fees decided from time to time. The credit card issuer often offers its customers a line of credit (LOC), allowing them to lend cash withdrawals. Issuers usually preset lending thresholds depending on specific creditworthiness [1, 2]. The use of credit cards is vital these days, and it plays a significant role in e-commerce and online funds transfer [3, 4]. The ever-increasing use of credit cards has posed many threats to the users and the companies issuing such cards. Fraudsters keep on finding new ways to commit cheating, which can cause considerable losses to card users and these companies as well [5, 6].

*1.1. Credit Card Payment Processing Steps.* Figure 1 illustrates how payments are transferred to the vendor's bank account, whenever the clients make purchases through the credit card [7]:

(a) A client sends a credit card purchase via Internet of Things- (IoT-) enabled swipe devices/POS/online sites.

(b) Payment gateway collects and transfers the transaction details safely to the merchant's bank computer-based controller system

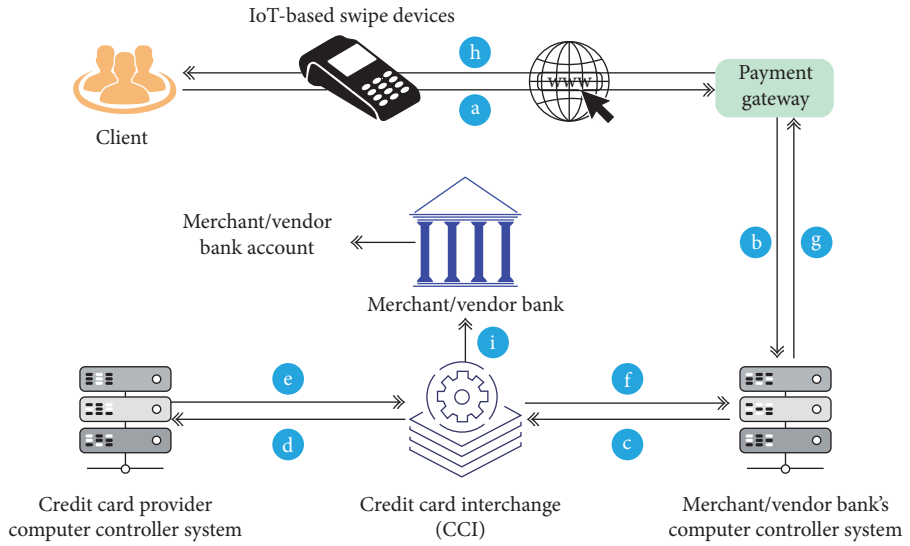



Figure 1: Payment process in the credit card system [7].

(c) The bank processor forwards the verification (i.e., processing, clearing, and settlement) process to the Credit Card Interchange (CCI)

(d) The CCI transfers the transaction to the client's credit card provider

(e) The card provider accepts or rejects the purchase based on current funds in the client's account and passes back the transaction information to the CCI

(f) The CCI transmits transaction information to the vendor's bank computer-based controller system

(g) The controller system of the vendor's bank transmits transaction details further to the payment gateway

(h) The payment gateway keeps and delivers transaction details to the vendor and/or client

(i) The CCI transfers the required funds to the vendor's bank, which further transfers funds into the merchant's account [7]

*1.2. Fraud in Credit Card Transaction.* Fraud and illegal behavior have various perspectives. The Association of Certified Fraud Examiners (ACFE) is a professional fraud examiner organization. Its activities include producing information, forming tools, and imparting training to avoid frauds. The ACFE has termed "fraud" as usage of one's profession for self-benefit via deliberate misapplication or misuse of assets of the organization [3]. A fraud is committed with the chief intention to acquire access by illegal means. It adversely affects the economic growth, governance, and even fundamental social values. Any technical infrastructure involving money and resources can be breached by unethical practices, e.g., auction site systems, medical insurance, vehicle insurance, credit cards, and banking. Cheating in these applications is perceived as cyber crime, potentially causing significant economic losses [3, 8].

Fraud can lower the trust in the industry, disturb the economic system, and significantly impact the overall living costs [9, 10]. IoT-enabled systems maintain the trace of their operational activities, which can be beneficial for analyzing some specific patterns. The previous methods based on manual processing such as auditing were cumbersome and ineffective due to large-size data or its attributes. Data mining techniques are considered effective in assessing small outliers in large datasets [9, 11, 12]. Frauds lead to heavy business losses. The credit card frauds contribute hundreds of millions of dollars per year for the lost revenue, and some estimates have indicated that US cumulative annual costs could surpass $400 billion [9].

*1.3. Types of Credit Card-Related Frauds.* The advancements in technology such as the Internet and mobile devices have contributed to increased fraudulent activities in recent times [13]. Fraudsters keep on finding new techniques, and therefore, monitoring systems are required to evolve correspondingly. Frauds related to credit cards can be broadly categorized into offline and online frauds [14]:

(i) Offline credit card fraud occurs whenever fraudsters stole the credit card and used it as the rightful owner in outlets. This is unusual as financial firms will promptly block the missing card whenever cardholders suspect the theft [3].

(ii) Online credit card frauds are more common and serious as compared with offline frauds in which credit card details are compromised by fraudsters through phishing, website cloning, and skimming and used in digital transactions [3, 15].

Global connectivity through new and advanced technology has exponentially increased the credit card frauds. Thus, the issue has acquired an alarming dimension in the present scenario, and a suitable system needs to be developed for detecting and avoiding such frauds.

*1.3.1. Fraud Prevention System (FPS).* FPS is the first form of defense for technological systems toward forgery. The aim of this phase is to suppress first-place fraud. The techniques in this phase prohibit, destroy, and respond to cyber attacks in computer servers (software and hardware), networks, or data, for example, encryption algorithms and firewall to decipher data and to block inner private networks from outside world, respectively [3, 16].

*1.3.2. Fraud Detection System (FDS).* FDS becomes the next safety measure to spot and recognize the fraudulent practices when they reach the networks and notify these to a network administrator [17]. Earlier, manual auditing methods such as discovery sampling were used to detect any such fraud [18]. This method had to tackle different environmental, political, legal, and business practices. To improve detection efficiency, computerized and automatic FDSs were developed. FDS capacities have been constrained however, as identification is primarily based on predefined rules set by the experts. Different data mining approaches are being developed to detect the frauds effectively. Oddity or outlier identification in FDS depends on behavioral profiling methods that model the pattern of behavior for every entity and assess any divergence from the normal [19]. Many authors have adopted anomaly-based FDSs in different areas of fraud detection [20–23].

*1.4. Distributed Deployment of Security-Related Aspects.* Financial firms have indeed acknowledged that the deployment of isolated control systems on solo delivery channels apparently no longer implements the requisite degree of vigilance toward illegal account operation. An additional layer of security, i.e., "Fraud Management," is enhancing the robustness by combining with security protocols at the level of standard channel [24]. The implemented fraud detection strategy can be distributed as reactive and proactive, depending on the point where data analysis is implemented in different transaction orders. Fraud identification approaches derived from data processing, neural networks, and/or various deep learning algorithms conduct sophisticated model processing via collected datasets in reactive fraud management to identify suspect transfers.

The newly arrived operations are evaluated "on the fly" in proactive fraud management before proper authorization and finalization, to allow the detection of unusual occurrences prior to any financial value movement. Proactive fraud detection is accomplished by relocating the inherent security which allows real-time scanning prior to completion of the transaction. Statistical analysis and data mining-related approaches have been implemented on classed post-transactional data to derive common traits correlated to suspicious occurrences in fraud strategic management.

*1.5. Data Imbalance Is a Major Concern.* Skewed distribution is regarded as one of the chief sensitive problems of FDS [3]. Usually, the skewed data problem is the scenario where there are far fewer instances of fraudulent cases than usual [25], making it difficult for learners to uncover trends in minority class data [26]. Moreover, class imbalance has a significant influence on the efficiency of classification models, which are normally dominated by majority class labels. Imbalanced datasets have a detrimental effect on classification performance that tends to be overshadowed by the majority class, thereby ignoring the minority class. As shown in Figure 2, the data-balancing methods can be divided into two subcategories, viz., data level methods and algorithmic level methods [27].

*1.5.1. Data Level Methods.* Such methods are taken as preprocessing to reorient the collected data before applying the classification algorithms. Many investigators have used the balancing methods, viz., undersampling or oversampling, in FDS-related studies [3]. In undersampling, a portion of the dataset of the dominant class is eliminated [28]. A broad range of FDS has used the undersampling technique to equalize training samples. The oversampling method duplicates minority class data samples. The oversampling technique is not frequently used because it induces overfitting of a model, especially for noisy data [29]. Synthetic minority oversampling technique (SMOTE) [30] is being used for fraud detection and considered as a superior complement to its current peers. SMOTE synthesizes new minority instances in the reported zone. Investigators, in their study [31], have conducted many simulations using various data level methods (SMOTE and EasyEnsemble) to identify the most suitable credit card FDS [3].

*1.5.2. Algorithmic Level Methods.* In this category, classifiers have been used to detect suspicious classes in a sample dataset. The algorithmic level approach uses cost-sensitive learning (CSL) to counter unequal class distribution. CSL places a cost variable to misinterpret the various classes by presuming that a cost matrix is present for various errors. Cost matrix structure is significantly correlated with these observations: false negative/positive and true negative/positive [32]. Another algorithmic approach followed in the FDS literature would be to use learners to manage imbalanced distribution. Such learners are either immune to class inequality by the learner's intrinsic characteristics as with Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [33] or the learners are reinforced against the issue by intrinsic alterations [3].

Falsified transactions have a narrow percentage in the overall dataset that may hinder the efficiency of FDS. In credit card systems, misclassifying legitimate transactions causes dissatisfied customers, which itself is regarded more detrimental than fraud itself. As mentioned above, two approaches, viz., algorithmic and data levels, were used to fix class imbalances. The researchers, in their works [34–38], have used undersampling techniques while dealing with the concern of class skewness in credit card FDS. However, Stolfo et al. [26] have used the oversampling method in the preprocessing stage of credit card FDS.

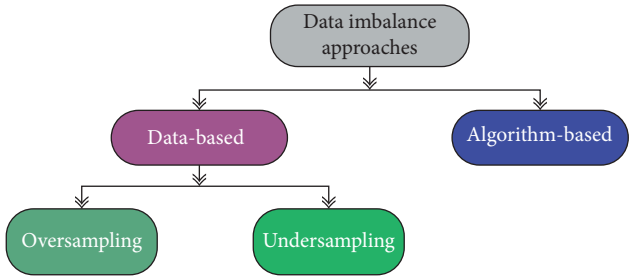



Figure 2: Various techniques of handling the concern related to data imbalance.

On the contrary, an algorithmic level approach has been followed using cost-sensitive learning techniques or by using the learner itself to manage uneven distribution. Sahin et al. [39] have used cost-sensitive classifiers to address the class imbalance. Dorronsoro et al. [21] have used nonlinear discriminant analysis (NLDA) neural models to tackle the class with imbalances. Ju and Lu [40] have used an enhanced imbalance class weighted support vector machine (ICW-SVM) to handle the skewness of the dataset. Bentley et al. [41] have given a fraud density map to enhance detection accuracy. In a study by Pozzolo et al. [42], the authors have suggested a race model to choose the right approach for an imbalance dataset. Chen [28] has used the binary support vector system (BSVS) and genetic algorithm (GA) to achieve a higher prediction accuracy from imbalance inputs. Minegishi and Niimi [43] have suggested the creation of a very fast decision tree (VFDT) learner, which could be tailored for extremely unbalanced datasets. Seeja and Zareapoor [44] have proposed FraudMiner for managing class imbalance via explicitly entering unbalanced data to the classification model. G.C. de Sá et al. have customized the bayesian network classifier (BNC) algorithm for credit card fraud detection [45]. Husejinovic has introduced a methodology to detect credit card fraud using naive bayesian and C4.5 decision tree classifiers [46]. Arya et al. have proposed deep ensemble learning to identify fraud cases in real-time data streams. The proposed model is capable of adapting to data imbalance as well as is robust to innate transaction patterns such as purchasing behavior [4].

## 2. Scope of the Study

This manuscript explores the concern of classifying imbalanced data by merging data level and algorithm level techniques to detect the fraudster from the log files generated for credit cards used at IoT-enabled terminals. Furthermore, an appropriate alert message can be sent to either the credit card holder or the issuer for reverting/blocking the transaction. Here, the random undersampling (RUS) approach has been deployed at the data level and boosting at the algorithmic level. The merger of these two components is RUSBoost [47]. Here, RUS is a data sampling technique that aims to mitigate class inequality by modifying the training dataset's class distribution. RUS eliminates instances from the majority class completely at random before a reasonable class distribution is reached [48, 49]. The boosting method helps in improving the classification precision of weak classifiers by combining weak hypotheses. Initially, all training dataset examples are given equal weights. Base learner forms a weak hypothesis during each iteration of adaptive boosting (AdaBoosting). Boosting is said to be adaptive since poor learners are subsequently tweaked in support of cases which are not classified by former classifiers. The inconsistency connected with the hypothesis is determined, and the weight of each instance is modified in such a manner that incorrectly classified cases raise their weights, whereas correctly classified samples decrease their weights. Thus, successive boosting steps will produce hypotheses which are able to correctly classify the previous incorrectly labeled instances. After all repetitions, a weighted vote would be used to allot a class to samples in the dataset [48]. RUSBoost is less costly than oversampling and bagging when used for classification (like SMOTEBagging).

## 3. Methodology

Figure 3 highlights the various phases, taking credit card transactional logs (imbalanced dataset) as input and giving an alert to the bank or the credit card holder regarding the status of the transactions performed at some IoT-based terminals.

Figure 3 shows that on the credit card transactional logs, the customized RUSBoost (CtRUSBoost) gets applied and results into showing the status of the transaction held. Here, the approach constitutes random undersampling and boosting using decision tree as per the normal RUSBoost algorithm with a further add-on/customization of having bagging process using SVM. CtRUSBoost can be deployed at the stage/step of either Credit Card Interchange or Credit Card Provider Computer Controller System (as shown in Figure 1), and from these controlling systems, an alert message can be escalated for suspending or stopping the financial transaction. The various symbolic notations used in the proposed algorithm CtRUSBoost have been defined in Table 1.

The RUSBoost given by Seiffert et al. [48, 49] has been modified by the authors here in this research work. The rounded rectangles at steps 2d, 2e, 3a, 3b, and 4 show the customization proposed by the authors here, which has resulted in comparatively better outcomes. In step 1, the weights of each sample are initialized to $(1/x)$, where $x$ is the total of instances in the training dataset. The weak hypotheses, viz., DT and SVM, are iteratively trained in steps 2a–2i. In step 2a, random undersampling has been implemented to suppress the class labels until the required minority class proportion is reached in the current (temporary) training dataset $\mathrm{SEG}'_z$. For example, if the required class proportion is 50:50, then most class instances are predictably excluded until majority and minority class instances are comparable. Therefore, $\mathrm{SEG}'_z$ will have a new distribution of weight as $\mathrm{DIS}'_z$. Step 2b moves $\mathrm{SEG}'_z$ and $\mathrm{DIS}'_z$ to the decision tree, generating the weak hypothesis $h_z$ (step 2c). In step 2d, support vector machine has been employed to compute the weak hypothesis $h_z^{\mathrm{svm}}$ in step 2e. The pseudo loss $\varepsilon_t$ (based on SEG and $\mathrm{DIS}_z$) has been determined in step 2f.

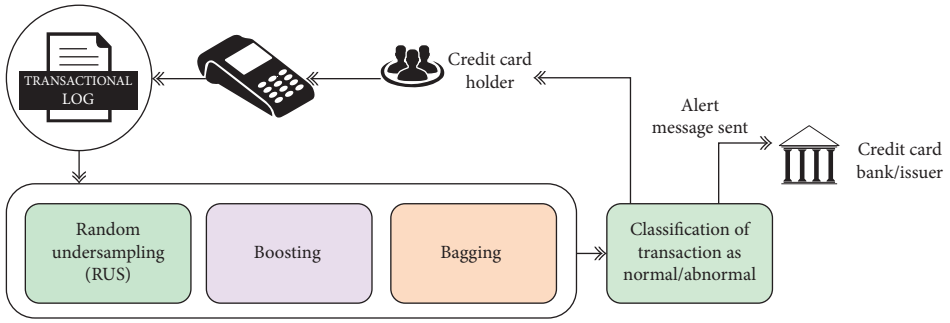



Figure 3: Steps involved in classification of imbalanced transactional logs as normal or abnormal.

Table 1: Symbolic notations used in the proposed algorithm CtRUSBoost.

| | |
|---|---|
| SEG | Dataset segment under consideration |
| $h_z^{\text{svm}}(p_k)$ | Hypothesis value obtained through support vector machine in $z^{\text{th}}$ iteration for the instance $p_k$ (this serves as a numeric confidence rating) |
| $h_z(p_k)$ | Hypothesis value obtained through decision tree in $z^{\text{th}}$ iteration for the instance $p_k$ (this serves as a numeric confidence rating) |
| $\varepsilon_z$ | Cumulative pseudo loss |
| $\alpha_z$ | Parameter to update the weight factor |
| $C_z$ | Factor for normalizing the $(z+1)^{\text{th}}$ distribution of weights taking the full training dataset/or normalized value for the distribution |
| $\text{DIS}_z(k)$ | Distribution of weights at $z^{\text{th}}$ iteration taking the full training dataset for the $k^{\text{th}}$sample |
| $\text{DIS}_{z+1}$ | Distribution of weights at $(z+1)^{\text{th}}$ iteration taking the full training dataset |
| $\text{DIS}_z'$ | Distribution of weights for $z^{\text{th}}$ temporary training dataset |
| $\text{SEG}_z'$ | $z^{\text{th}}$temporary training dataset |
| $p_i$ | $i^{\text{th}}$ row with values of all columns except the last one (i.e., label) |
| $q_i$ | A label for the $i^{\text{th}}$ row |
| $q^r$ | Minority class label |
| $Z$ | Total number of iterations employed in the ML model |
| $k$ or $x$ | Total counts of samples present in the SEG |
| $P$ | Rows/tuples in the dataset (excluding the last column having labeled entries) |
| $Q$ | Total available labels in the dataset |

In step 2f, the hypothesis values for those tuples have only been considered where there is a misclassification. Here, in the subexpression $q_k \neq q$, $q_k$ means the original label/class of the $k^{\text{th}}$ row/tuple in the dataset and $q$ is the label/class obtained after employing/deploying the weak learner decision tree. Subexpression $h_z(p_k, q_k)$ is the numeric confidence value in $z^{\text{th}}$ iteration for the instance $p_k$, where the label is $q_k$, and subexpression $h_z(p_k, q)$ is the numeric confidence value in the same $z^{\text{th}}$ iteration for the instance $p_k$ considered earlier, where the label is mismatched and obtained as $q$ instead of $q_k$. In step 2g, the parameter $\alpha$ is computed as $(\varepsilon_z/(1-\varepsilon_z))$which symbolizes the weight update. In step 2h, the weight distribution gets updated $\text{DIS}_{z+1}$. Step 2i normalizes the value computed in the previous step. After the completion of $Z$ iterations, in step 3a, the maximum value of $h_z$ has been computed among the ones given by decision tree under boosting, where the knowledge/learning from the previous dataset segment has been used for getting the hypothesis value of the next dataset segment, but in the last step, all the results have not been merged to obtain the final one. Instead, the final value of the hypothesis has been obtained from the last dataset segment. In step 3b, hypothesis values as obtained by employing SVM for each dataset segment in $Z$iterations have been finalized by performing voting or averaging among all the values of $h_z^{\text{svm}}$. In step 4, the final hypothesis $H(p)$ has been computed taking the maximum of the value obtained for $h_z$ and $h_z^{\text{svm}}$.

## 4. Results and Experiment

The results obtained after using the three different datasets, viz., (i) Abstract Dataset for Credit Card Fraud Detection [50], (ii) Default of Credit Card Client Dataset [51], and (iii) Credit Card Fraud Dataset [52] are shown in this section. Customized RUSBoost results were compared using RUSBoost, decision tree (DT), logistic regression (LR), multilayer perceptron (MLP), $K$-nearest neighbors (KNN), random forest (RF), AdaBoost, and support vector machine (SVM).

Three separate datasets based on the number of tuples were taken for the current work. Datasets of less than five thousand tuples were considered as small; tuples with a range of over five thousand and less than ten thousand were considered as medium; and those with a range of over ten thousand entries were considered as large. All the datasets have been divided into two partitions, i.e., 80% and 20% of the full dataset, where the bigger portion has been taken for training and the smaller one for testing of the machine learning models.

*4.1. Small Dataset.* The dataset called Abstract Dataset for Credit Card Fraud Detection (Dataset A) [50] has been taken from the kaggle.com database. The authors classified this as a small dataset with less than 5,000 tuples. The dataset included the usage of 3,075 clients and 11 attributes. Of the 3,075 samples, 2,627 represent nonfraudulent transactions and 448 are fraudulent transactions (about 6:1). The eleven variables taken in this dataset are described in Table 2.

*4.2. Medium Dataset.* The dataset called Default of Credit Card Client Dataset (Dataset B) [51] has also been taken from the kaggle.com database. This includes details on default payments, demographic factors, credit data, payment history, and credit card company bills in Taiwan from April 2005 to September 2005. Among the 30,000 observations, 23,364 are cardholders with default payment as no and 6,636 with status as yes (about 4:1). Default payment in the finance domain is known as nonrepayment of debt such as interest or principal toward credit or estate. A default can result when a purchaser could not render payments on time, slows payouts, or declines or drops payment [53].

This dataset used a binary variable default payment as the answer variable. Table 3 explains the twenty-four variables taken up in Dataset B.

*4.3. Large Dataset.* The dataset called Credit Card Fraud Detection (Dataset C) [52] was taken again from the kaggle.com database. This dataset includes purchases by European cardholders in September 2013. This sample dataset outlined two-day activities, with 492 frauds out of 284,807 total transactions. The dataset is highly imbalanced, where the positive class (fraud) constitutes 0.172% of all transactions deemed. The details of the dataset's features are given in Table 4 and include all numeric values.

It includes only numerical variables resulting from PCA transformation. Kaggle did not provide any original features as well as additional details due to privacy concerns. Features $V_1, V_2, \ldots,$ and $V_{28}$ are the key PCA components with untransformed attributes as "time" and "amount."

*4.4. Evaluation Metrics.* Assessment measures are employed to calculate statistic or machine learning model efficiency. A confusion matrix gives us the output matrix that characterizes the model's complete efficiency. Here, in the proposed model, the security context is said to be robust if the model is capable of finding/classifying fraudster transactions accurately. The metrics used for comparing ML models for their accuracy are sensitivity and specificity from the confusion matrix, precision, $F1$ score, receiver operating characteristic (ROC), and area under precision recall (AUPR).

*4.4.1. Confusion Matrix.* The confusion matrix is a representation of an algorithm's performance in the field related to machine learning. The term "Confusion" has appeared from the fact that if the machine learning model causes confusion between two classes, it is easy to see. Figure 4 depicts a confusion matrix providing sensitivity, specificity, recall, and fall-out information. The column in this matrix represents instances in the actual class, while each row represents instances in one expected class.

Sensitivity is an estimate of the total of truly positive instances expected to be positive. The larger sensitivity value will have a high true positive value and less false negative value. Models with high sensitivity are required for health and financial purposes. Specificity is defined as the share of actual negatives, predicted to be negative. This ratio may also be called the false positive rate. The higher specificity value will mean the higher true negative and lower false positive rate.

*4.4.2. Precision and F1 Score.* Precision and $F$-measurements are considered more suitable for estimating the performance of a classification algorithm when the dataset is imbalanced, where precision is characterized as the positive predictive value. $F$-measure in the confusion matrix is the weighted harmonic mean of sensitivity and precision [54]:

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ F1 &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \end{aligned} \tag{1}$$

Precision is the percentage of true positives to all positives. For our problem statement here, the precision would be the measure of fraudster transactions that we correctly identified as fraud out of all the transactions, which are actually fraud. Recall refers to the proportion of the overall predictions of the algorithm being accurately categorized. Furthermore, the value of $F1$ gives a single score that balances out both recall and the precision.

Here, decision tree, logistic regression, multilayer perceptron (MLP), $K$-nearest neighbor (KNN), random forest (RF), AdaBoost, and support vector machine (SVM) models have been compared w.r.t. sensitivity, specificity, precision, and $F1$ score. Decision tree is a nonparametric, supervised learning system for classification and regression tasks. The decision tree is designed using an algorithmic method that recognizes ways of splitting data based on different conditions. Logistic regression is an algorithm for machine learning that is based on the probability principle. It is an algorithm for classification used to attribute observations to a specific class set. Using the logistic sigmoid function, logistic regression transforms the output to return a probability value. A multilayer perceptron is a neural network that links different layers in a directed graph, meaning the signal path through nodes only goes one directional. In MLP, every node is having a nonlinear activation function, except the input nodes. $K$-nearest neighbor is a single algorithm that holds all existing cases in a similarity measure (i.e., distance function) and classifies new cases. The random forest algorithm generates decision trees on data samples and then obtains predictions from each and finally, picks the best option by voting. In AdaBoost, a sequence of weak learners is linked so that each weak classifier attempts to enhance the

(i) Input: $x$, SEG, $P \times Q$(with $q^r \in Q$, $|Q| = 2$)
(ii) Output: maximum of [(maximum of $h_z$ value), (maximum of $h_z^{\text{svm}}$value)]
Begin
(1) Initialization of $\text{DIS}_1(k) = 1/x$ for all $k$
(2) Do for $z = 1, 2, 3, \ldots, Z$
(a) Create temporary training dataset $\text{SEG}_z'$ with weight distribution $\text{DIS}_z'$ by using random undersampling
(b) Call decision tree, considering the sample set as $\text{SEG}_z'$ and distribution of weight $\text{DIS}_z'$
(c) Compute a hypothesis $h_z$: $P \times Q \longrightarrow [0, 1]$
(d) Call support vector machine considering the sample set as $\text{SEG}_z'$ and distribution of weight as $\text{DIS}_z'$
(e) Compute a hypothesis $h_z^{\text{svm}}$: $P \times Q \longrightarrow [0, 1]$
(f) Compute the pseudo loss for SEG and $\text{DIS}_z$
$\varepsilon_z = \sum_{(k,q):\, q_k \neq q} \text{DIS}_z(k)(1 - h_z(p_k, q_k) + h_z(p_k, q))$
(g) Compute the parameter to update the weighing factor:
$\alpha_z = (\varepsilon_z / 1 - \varepsilon_z)$
(h) Update $\text{DIS}_z$:
$\text{DIS}_{z+1}(k) = \text{DIS}_z(k)\alpha_z^{(1/2)(1+h_z(p_k,q_k)-h_z(p_k,q:\, q_k \neq q))}$
(i) Normalize $\text{DIS}_{z+1}$: Let $C_z = \sum_z \text{DIS}_{z+1}(k)$
$\text{DIS}_{z+1}(k) = (\text{DIS}_{z+1}(k)/C_z)$
(3) Find the values for $h_z$ and $h_z^{\text{svm}}$
(a) For each value of $h_z$, where $z = \{1, 2, \ldots, Z\}$, find out the maximum value of $h_z$
(b) For each value of $h_z^{\text{svm}}$, where $z = \{1, 2, \ldots, Z\}$, apply bagging either by performing voting or averaging among all the values of hypothesis obtained
(4) Compute the final hypothesis $H(p)$ as the maximum value between $h_z$and $h_z^{\text{svm}}$
End

Algorithm 1: CtRUSBoost (customized RUSBoost).

Table 2: Attribute number, name, and definition of Dataset A.

| Attribute | Description |
|---|---|
| $X1$ | Merchant ID : ID of the merchant |
| $X2$ | Average amount/transaction/day |
| $X3$ | Total amount of transaction |
| $X4$ | Is declined: declining or falling transaction (yes or no) |
| $X5$ | Total number of declines/days: total transaction numbers declined daily |
| $X6$ | Is foreign transaction: transaction carried out is or is not a foreign transaction |
| $X7$ | Is high-risk country: transaction is performed in countries under high risk |
| $X8$ | Average daily chargeback amount |
| $X9$ | Average chargeback (taken for six months) |
| $X10$ | Frequency of chargeback (taken for six months) |
| $X11$ | Is fraudulent: transaction is a fraud or not |

classification of observations incorrectly labeled by the preceding weak classifier. Support vector machine uses a kernel trick to transform data and then determines an optimal boundary between potential outputs. The results showing comparison among customized RUSBoost, decision tree, logistic regression, multilayer perceptron (MLP), $K$-nearest neighbor (KNN), random forest (RF), AdaBoost, and support vector machine (SVM) models have been presented in Tables 5–7.

In Table 7, the value that has been observed for the precision and $F1$ score is NaN under SVM because the zero divided by zero is undefined as a real number, and in computing systems, it can be represented as NaN.

*4.4.3. Receiver Operating Characteristic (ROC).* In machine learning, measuring efficiency is an integral activity. ROC is considered the most significant measurement to test the efficiency of any classification model. It tells how much the model can differentiate between classes. The higher the AUC, the better it would be to predict 0 s as 0 s and 1 s as 1 s. The curve for ROC is plotted with TP rate vs. FP rate, taking TP and FP rates at $y$-axis and $x$-axis, respectively [55]. Figures 5–7 depict the ROC for the customized RUSBoost and its peer techniques, i.e., simple RUSBoost, DT, LR, MLP, KNN, RF AdaBoost, and SVM, indicating the optimality of the proposed customization in RUSBoost on the benchmark datasets A, B, and C, respectively.

Table 3: Attribute number, name, and definition of Dataset B (amount in New Taiwan or NT dollar).

| Attribute | Description |
|---|---|
| $X1$ | Credit amount |
| $X2$ | Gender of the borrower<br>1 for male<br>2 for female |
| $X3$ | Level of education<br>1 Graduate school<br>2 University<br>3 High school<br>4 Others<br>5/6 Unknown |
| $X4$ | Marital status of the borrower<br>1 Married<br>2 Single<br>3 Others |
| $X5$ | Age of the credit card holder (in years) |
| $X6$–$X11$ | PAY_1 to PAY_6: status of payment return in September to April 2005<br>Paid on-time payment = −1<br>One-month payment delay = 1<br>Two-month payment delay = 2<br>.<br>.<br>.<br>Nine or above months of payment delay = 9 |
| $X12$–$X17$ | BILL_AMT1-6: amount of bill for the months April to September 2005 |
| $X18$–$X23$ | PAY_AMT1-6: previous payment in April to September 2005 |
| $X24$ | Status as 1 for yes and 0 for no under the default payment |

| | | True condition | |
|---|---|---|---|
| | Total population | Actual condition positive | Actual condition negative |
| Predicted condition | Predicted condition positive | True positive (TP) rate, sensitivity $= \frac{\Sigma\ True\ positive}{\Sigma\ Condition\ positive}$ | False positive (FP) rate $= \frac{\Sigma\ False\ positive}{\Sigma\ Condition\ negative}$ |
| | Predicted condition negative | False negative (FN) rate $= \frac{\Sigma\ False\ negative}{\Sigma\ Condition\ positive}$ | True negative (TN) rate, specificity $= \frac{\Sigma\ True\ negative}{\Sigma\ Condition\ negative}$ |

Figure 4: Sensitivity, specificity, FP rate, and FN rate formulas in the confusion matrix.

Table 4: Attribute number, name, and definition of Dataset C.

| Attribute | Description |
|---|---|
| $V_1 \ldots V_{28}$ | The parameters have been anonymized with principal component analysis (PCA) to protect the user identities |
| Time | Time intervened between transactions (in seconds) |
| Amount | Amount of the transaction |
| Class | Final label; 1 = fraud, 0 = otherwise |

Besides ROC, the precision recall (PR) curves are also considered better for evaluating the algorithmic efficiency when the sample set is highly biased. The results of the current work are also presented through an AUPR curve obtained on various machine learning models.

*4.4.4. Area under Precision Recall (AUPR).* The ROC curve has some drawbacks, including class skew decoupling. That is why the precision recall (PR) curve, which plots precision against recall and is equivalent to the false discovery rate curve, has gained attention in recent years. This output

Table 5: Sensitivity, specificity, precision, and *F*1 scores obtained on Dataset A executing RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

| Model name | Sensitivity | Specificity | Precision | *F*1 score |
|---|---|---|---|---|
| RUSBoost | 50.6 | 99.8 | 33.4 | 40.2 |
| Customized RUSBoost | 96.3 | 85.6 | 94.2 | 88.6 |
| DT | 76.5 | 97.9 | 72.6 | 75.4 |
| LR | 57.0 | 99.0 | 86.0 | 68.7 |
| MLP | 70.4 | 99.5 | 95.8 | 81.1 |
| KNN | 80.6 | 99.9 | 95.1 | 87.2 |
| RF | 53.2 | 99.0 | 82.3 | 64.5 |
| AdaBoost | 73.4 | 99.0 | 83.7 | 78.2 |
| SVM | 61.2 | 99.9 | 96.8 | 75.7 |

Table 6: Sensitivity, specificity, precision, and *F*1 scores obtained on Dataset B executing RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

| Model name | Sensitivity | Specificity | Precision | *F*1 score |
|---|---|---|---|---|
| RUSBoost | 34.6 | 98.3 | 85.9 | 59.4 |
| Customized RUSBoost | 99.6 | 98.7 | 95.7 | 97.6 |
| DT | 40.6 | 81.0 | 49.5 | 50.7 |
| LR | 23.6 | 97.0 | 69.6 | 35.0 |
| MLP | 38.5 | 93.2 | 61.4 | 47.3 |
| KNN | 37.8 | 89.4 | 50.0 | 43.1 |
| RF | 5.5 | 99.2 | 68.2 | 10.2 |
| AdaBoost | 30.8 | 95.8 | 67.3 | 42.3 |
| SVM | 33.2 | 95.2 | 67.8 | 44.5 |

Table 7: Sensitivity, specificity, precision, and *F*1 scores obtained on Dataset C executing RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

| Model name | Sensitivity | Specificity | Precision | *F*1 score |
|---|---|---|---|---|
| RUSBoost | 34.6 | 98.3 | 85.9 | 59.4 |
| Customized RUSBoost | 99.6 | 98.7 | 95.7 | 97.6 |
| DT | 40.6 | 81.0 | 49.5 | 50.7 |
| LR | 23.6 | 97.0 | 69.6 | 35.0 |
| MLP | 38.5 | 93.2 | 61.4 | 47.3 |
| KNN | 37.8 | 89.4 | 50.0 | 43.1 |
| RF | 5.5 | 99.2 | 68.2 | 10.2 |
| AdaBoost | 30.8 | 95.8 | 67.3 | 42.3 |
| SVM | 33.2 | 95.2 | 67.8 | 44.5 |

metric has been widely used in various fields such as computer vision, computational biology, data analysis, medicine, and natural language processing. As a single score, the AUPR summarizes the precision recall curve and can be used to easily compare different binary classification models. The AUPR ′s value for a perfect classifier is 1. The high precision and recall system will provide correctly labeled results [55]. Figures 8–10 depict the AUPR for the customized RUSBoost and its peer techniques, i.e., simple RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM, indicating the optimality of the algorithm on the benchmark datasets A, B, and C, respectively.

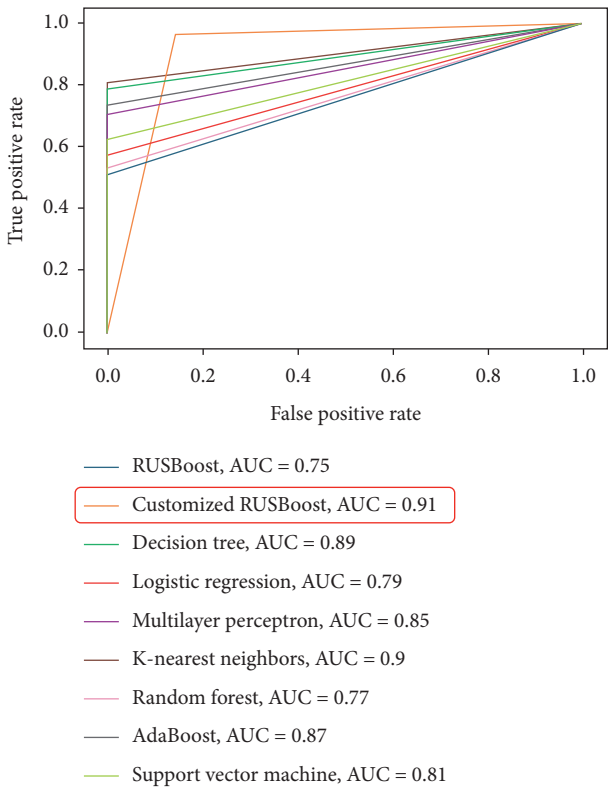



Figure 5: ROC curve obtained on the Default of Credit Card Client Dataset after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

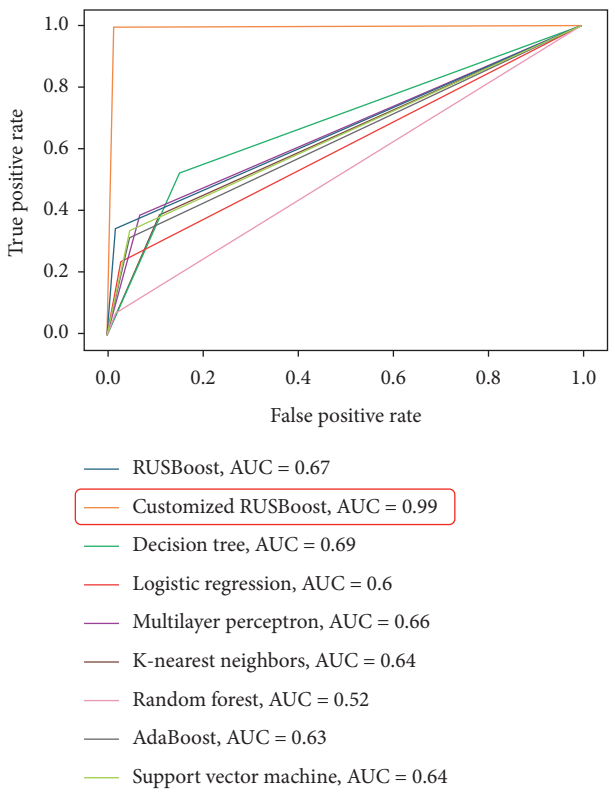



Figure 6: ROC curve obtained on the Credit Card Fraud Detection Dataset after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

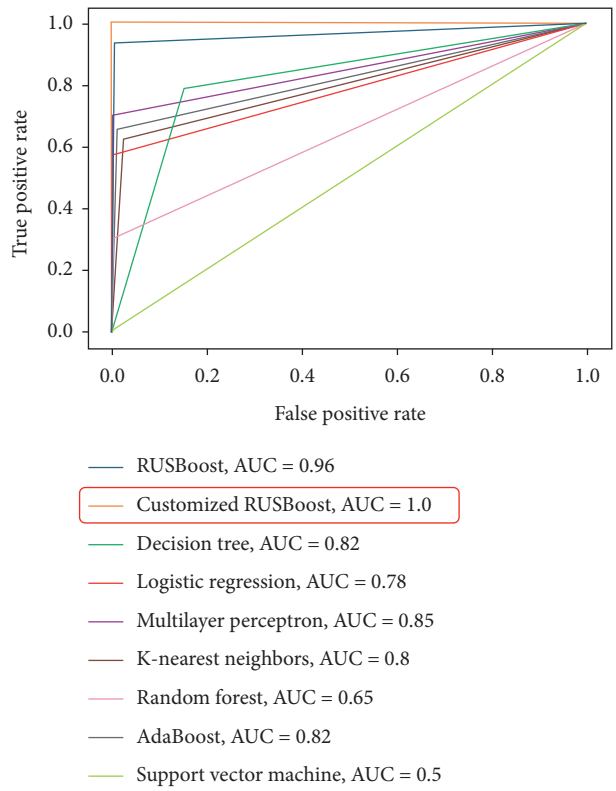



Figure 7: ROC curve obtained on the Abstract dataset for Credit Card Fraud Detection after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

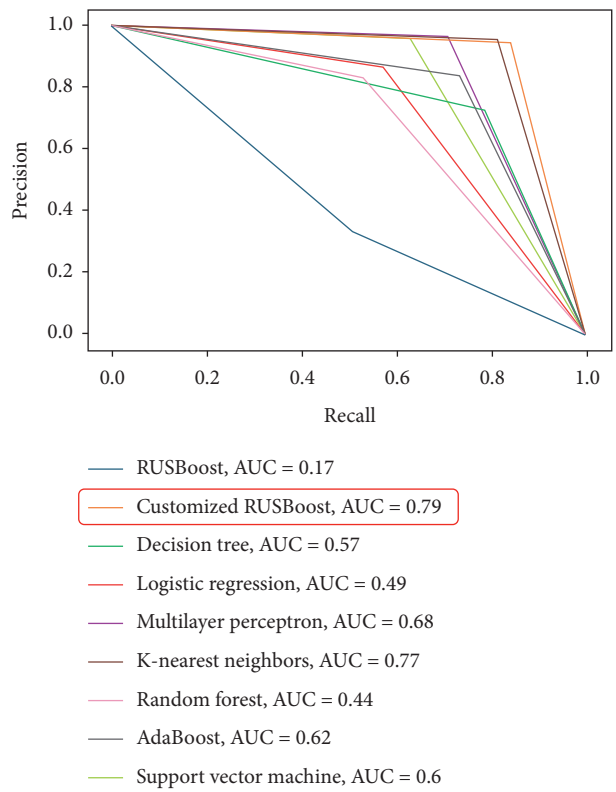



Figure 8: AUPR curve obtained on the Dataset A after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

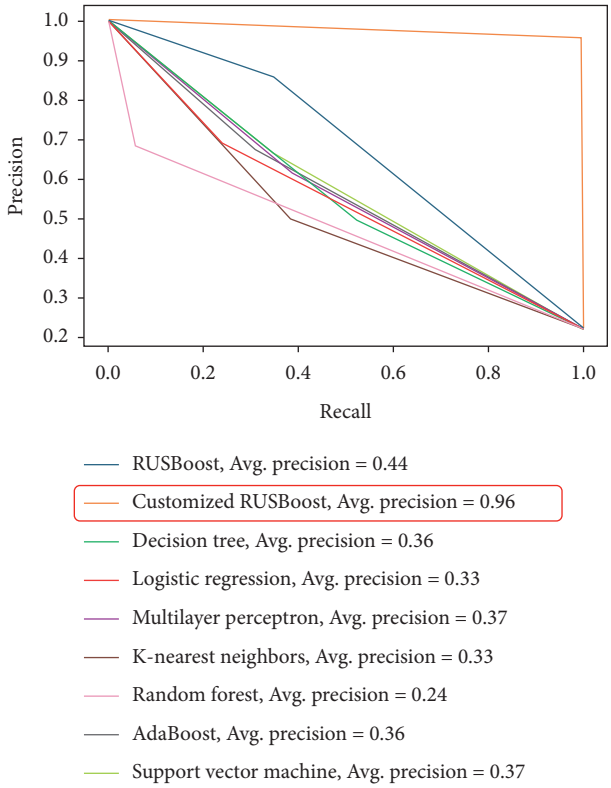



Figure 9: AUPR curve obtained on the Dataset B after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

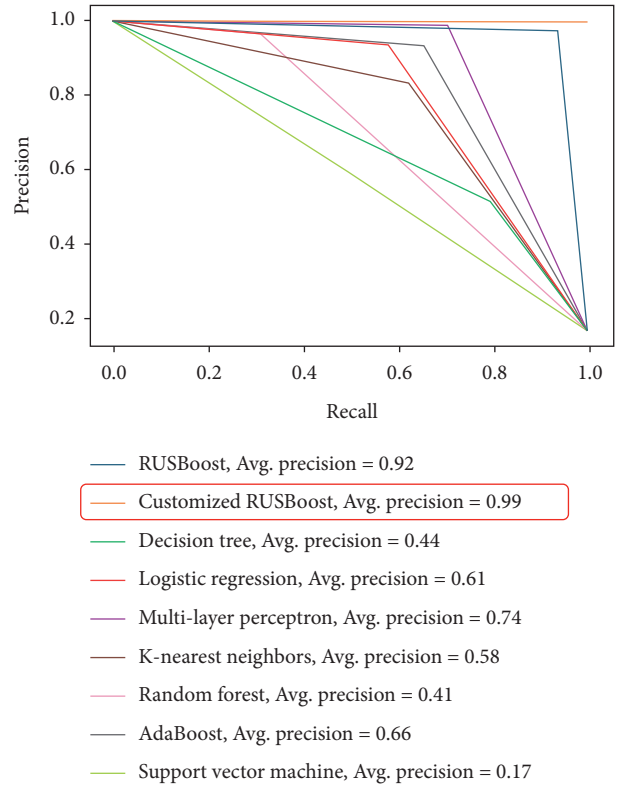



Figure 10: AUPR curve obtained on the Dataset C after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

## 5. Conclusion

In this research work, the existing RUSBoost algorithm has been customized by using a combination of bagging and boosting. The results obtained after customizing the RUSBoost in the proposed methodology are more reliable and authentic when compared with simple/normal RUSBoost, DT, RF, AdaBoost, SVM, LR, KNN, and MLP. The scores obtained for the CtRUSBoost algorithm on three benchmark datasets A, B, and C taken from kaggle.com are 96.30, 99.60, and 100, respectively, for sensitivity; 85.60, 98.70, and 99.80, respectively, for specificity; 94.20, 95.70, and 99.30, respectively, for precision; and 88.60, 97.60, and 99.60, respectively, for $F1$ score. The results obtained from CtRUSBoost have outperformed all the peer approaches used in this study by a large margin, which means it can detect fraudster transactions more robustly. In the future, the work proposed here can be customized further by adding weak classifiers to the process such as $K$-nearest neighbors, linear regression, and multilayer perceptron.

## Data Availability

The datasets used during the current study are available at kaggle.com, and web links to the datasets are as follows: kaggle small-sized dataset, https://www.kaggle.com/shubhamjoshi2130of/abstract-data-set-for-credit-card-fraud-detection, kaggle medium-sized dataset, https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset, and kaggle large-sized dataset, https://www.kaggle.com/mlg-ulb/creditcardfraud. The datasets used to support the findings of this study are included within the article at reference numbers [50–52].

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (no. 2018R1A4A1025632).

## References

[1] L. Delamaire, H. Abdou, and J. Pointon, "Credit card fraud and detection techniques: a review," *Banks and Bank Systems*, vol. 4, no. 2, pp. 57–68, 2009.

[2] S. Benson Edwin Raj and A. Annie Portia, "Analysis on credit card fraud detection methods," in *Proceedings of the 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*, pp. 152–156, Tamil Nadu, India, March 2011.

[3] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: a survey," *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.

[4] M. Arya and G. Hanumant Sastry, "DEAL–"deep ensemble algorithm" framework for credit card fraud detection in real-time data stream with Google TensorFlow," *Smart Science*, vol. 8, no. 2, pp. 71–83, 2020.

[5] K. K. Sherly and R. Nedunchezhian, "BOAT adaptive credit card fraud detection system," in *Proceedings of the 2010 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–7, Coimbatore, India, December 2010.

[6] N. Khare, P. Devan, C. Lal Chowdhary et al., "Spider monkey optimization and deep neural network hybrid classifier model for intrusion detection," *Electronics*, vol. 9, no. 4, p. 692, 2020.

[7] "InterWeave payment gateway, CreatioMarketplace," 2020, https://marketplace.creatio.com/app/interweave-payment-gateway.

[8] S. P. Mishra and P. Kumari, "Analysis of techniques for credit card fraud detection: a data mining perspective," in *New Paradigm, in Decision Science and Management*, I. A. S. Patnaik, M. Tavana, and V. Jain, Eds., vol. 1005, pp. 89–98, Springer, Singapore, Asia, 2020.

[9] J. West and M. Bhattacharya, "Intelligent financial fraud detection: a comprehensive review," *Computers & Security*, vol. 57, pp. 47–66, 2016.

[10] J. Johannes, M. Granitzer, K. Ziegler et al., "Sequence classification for credit-card fraud detection," *Expert Systems with Applications*, vol. 100, pp. 234–245, 2018.

[11] V. Sharma, R. Kumar, W.-H. Cheng, M. Atiquzzaman, K. Srinivasan, and A. Y. Zomaya, "Neuro-fuzzy based horizontal anomaly detection in online social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 11, pp. 2171–2184, 2018.

[12] D. Yue, X. Wu, Y. Wang, Li Yue, and C.-H. Chu, "A review of data mining-based financial fraud detection research," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 5519–5522, Shanghai, China, September 2007.

[13] Z. Li, V. Sharma, and S. P. Mohanty, "Preserving data privacy via federated learning: challenges and solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 3, pp. 8–16, 2020.

[14] N. Laleh and M. A. Azgomi, "A taxonomy of frauds and fraud detection techniques," in *Proceedings of the International Conference on Information Systems, Technology and Management*, pp. 256–267, Ghaziabad, India, March 2009.

[15] S. Zhang and J.-H Lee, "A group signature and authentication scheme for blockchain-based mobile-edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4557–4567, 2019.

[16] S. Singh, P. K. Sharma, B. Yoon, M. Shojafar, G. H. Cho, and I.-H. Ra, "Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city," *Sustainable Cities and Society*, vol. 63, Article ID 102364, 2020.

[17] M. Behdad, L. Barone, M. Bennamoun, and T. French, "Nature-inspired techniques in the context of fraud detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1273–1290, 2012.

[18] S. Tennyson and P. Salsas-Forn, "Claims auditing in automobile insurance: fraud detection and deterrence objectives," *Journal of Risk & Insurance*, vol. 69, no. 3, pp. 289–308, 2002.

[19] J. Veeramreddy, V. V. Rama Prasad, and K. Munivara Prasad, "A review of anomaly based intrusion detection systems," *International Journal of Computer Applications*, vol. 28, no. 7, pp. 26–35, 2011.

[20] S. Ghosh and D. L. Reilly, "Credit card fraud detection with a neural-network," in *Proceedings of the System Sciences, Proceedings of the Twenty-Seventh Hawaii International Conference*, pp. 621–630, Wailea, HI, USA, January 1994.

[21] J. R. Dorronsoro, F. Ginel, C. Sanchez, and C. Santa Cruz, "Neural fraud detection in credit card operations," *IEEE Transactions on Neural Networks*, vol. 8, no. 4, pp. 827–834, 1997.

[22] M. Taniguchi, M. Haft, J. Hollmén, and V. Tresp, "Fraud detection in communication networks using neural and probabilistic methods," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, pp. 1241–1244, Seattle, WA, USA, May 1998.

[23] R. Brause, T. Langsdorf, and M. Hepp, "Neural data mining for credit card fraud detection," in *Proceedings of the 11th International Conference on Tools with Artificial Intelligence*, pp. 103–106, Chicago, IL, USA, November 1999.

[24] E. Michael and P. R. Falcone Sampaio, "The design of FFML: a rule-based policy modelling language for proactive fraud management in financial data streams," *Expert Systems with Applications*, vol. 39, no. 11, pp. 9966–9985, 2012.

[25] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using Bayesian and neural networks," in *Proceedings of the 1st International Naiso Congress on Neuro Fuzzy Technologies*, pp. 261–270, Havana, Cuba, January 2002.

[26] S. J. Stolfo, D. W. Fan, W. Lee, and A. L. Prodromidi, "Credit card fraud detection using meta-learning," in *Proceedings of the AAAI Workshop on Fraud Detection and Risk Management*, pp. 83–90, Providence, RI, USA, July 1997.

[27] V. López, A. Fernández, G. Jose, Moreno-Torres, and F. Herrera, "Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6585–6608, 2012.

[28] R.-C. Chen, T. Chen, and C.-C. Lin, "A new binary support vector system for increasing detection rate of credit card fraud," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 20, no. 2, pp. 227–239, 2006.

[29] P. Brennan, "A comprehensive survey of methods for overcoming the class imbalance problem in fraud detection," M.Sc. in Computing Thesis, Institute of Technology, Blanchardstown, Dublin, Ireland, 2012.

[30] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[31] A. Dal Pozzolo, C. Olivier, Yann-Aël Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4915–4928, 2014.

[32] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Proceedings of the 3rd International Conference on Data Mining*, pp. 435–442, Melbourne, FL, USA, November 2003.

[33] P. K. Chan, W. Fan, A. Prodromidir, and S. Stalfo, "Distributed data mining in credit card fraud detection," *IEEE Intelligent Systems and Their Applications*, vol. 14, no. 6, pp. 67–74, 1999.

[34] F. Nick, R. Tubb, and P. Krause, "Neural network rule extraction to detect credit card fraud," in *Proceedings of the Engineering Applications of Neural Networks*, pp. 101–110, Corfu, Greece, September 2011.

[35] E. Duman and Y. Sahin, "Detecting credit card fraud by decision trees and support vector machines," in *Proceedings of*

*the International Multi Conference of Engineers and Computer Scientists (IMECS)*, vol. 1, Hong-Kong, China, March 2011.

[36] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: a comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.

[37] C. Phua, K. Smith-Miles, V. C.-S. Lee, and R. Gayler, "Resilient identity crime detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 533–546, 2010.

[38] E. Duman, A. Buyukkaya, and I. Elikucuk, "A novel and successful credit card fraud detection system implemented in a Turkish bank," in *Proceedings of the 13th International Conference on Data Mining Workshops*, pp. 162–171, Dallas, TX, USA, December 2013.

[39] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916–5923, 2013.

[40] Q. Lu and C. Ju, "Research on credit card fraud detection model based on class weighted support vector machine," *Journal of Convergence Information Technology*, vol. 6, no. 1, pp. 62–68, 2011.

[41] P. J. Bentley, J. Kim, G.-H. Jung, and J.-U. Choi, "Fuzzy darwinian detection of credit card fraud," in *Proceedings of the 14th Annual Fall Symposium of the Korean Information Processing Society*, vol. 14, Seoul, Korea, Ootober 2000.

[42] A. D. Pozzolo, O. Caelen, S. Waterschoot, and G. Bontempi, "Racing for unbalanced methods selection," in *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*, pp. 24–31, Hefei, China, October 2013.

[43] T. Minegishi and A. Niimi, "Proposal of credit card fraudulent use detection by online-type decision tree construction and verification of generality," *International Journal for Information Security Research (IJISR)*, vol. 1, no. 4, pp. 229–235, 2011.

[44] K. R. Seeja and M. Zareapoo, "FraudMiner: a novel credit card fraud detection model based on frequent itemset mining," *The Scientific World Journal*, vol. 2014, Article ID 252797, 2014.

[45] A. G. C. De S´a, A. C. M. Pereira, and G. L. Pappa, "A customized classification algorithm for credit card fraud detection," *Engineering Applications of Artificial Intelligence*, vol. 72, pp. 21–29, 2018.

[46] A. Husejinovic, "Credit card fraud detection using naive Bayesian and C4.5 decision tree classifiers," *Periodicals of Engineering and Natural Sciences*, vol. 8, no. 1, pp. 1–5, 2020.

[47] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "A novel noise-resistant boosting algorithm for class-skewed data," in *Proceedings of the 11th International Conference on Machine Learning and Applications*, pp. 551–557, Boca Raton, FL, USA, December 2012.

[48] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: a hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2009.

[49] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: improving classification performance when training data is skewed," in *Proceedings of the 19th International Conference on Pattern Recognition*, pp. 1–4, Tampa, FL, USA, December 2008.

[50] S. Joshi, "Abstract data set for credit card fraud detection," 2020, https://www.kaggle.com/shubhamjoshi2130of/abstract-data-set-for-credit-card-fraud-detection.

[51] U. M. Learning, "Default of credit card clients dataset," 2016, https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset.

[52] M. L. G. ULB, "Credit card fraud detection," 2018, https://www.kaggle.com/mlg-ulb/creditcardfraud.

[53] J. Chen, "Default," 2020, https://www.investopedia.com/terms/d/default2.asp.

[54] J. Akosa, "Predictive accuracy: a misleading performance measure for highly imbalanced data," in *Proceedings of the SAS Global Forum*, pp. 2–5, Orlando, FL, USA, April 2017.

[55] V. Arora, R. Leekha, R. Singh, and I. Chana, "Heart sound classification using machine learning and phonocardiogram," *Modern Physics Letters B*, vol. 22, no. 26, Article ID 1950321, 2019.

Hindawi
Mobile Information Systems
Volume 2020, Article ID 8831976, 13 pages
https://doi.org/10.1155/2020/8831976

*Research Article*

# Rogue Device Mitigation in the Internet of Things: A Blockchain-Based Access Control Approach

**Uzair Javaid,[1] Furqan Jameel,[2] Umair Javaid,[3] Muhammad Toaha Raza Khan,[4] and Riku Jäntti[2]**

[1]*Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117583*
[2]*Department of Communications and Networking, Aalto University, Espoo 02150, Finland*
[3]*IREC/MIRO and ICTEAM UCLouvain, Avenue Hippocrate 54, Brussels 1200, Belgium*
[4]*Kyungpook National University, Daegu 41566, Republic of Korea*

Correspondence should be addressed to Furqan Jameel; furqan.jameel@aalto.fi

Received 16 July 2020; Revised 12 August 2020; Accepted 9 October 2020; Published 28 October 2020

Academic Editor: Zengpeng Li



Recent technological developments in wireless and sensor networks have led to a paradigm shift in interacting with everyday objects, which nurtured the concept of Internet of Things (IoT). However, low-powered nature of IoT devices generally becomes a hindrance that makes them vulnerable to a wide array of attacks. Among these, the emergence of rogue devices is quickly becoming a major security concern. Rogue devices are malicious in nature which typically execute different kinds of cyberattacks by exploiting the weaknesses of access control schemes in IoT environments. Therefore, access control is one of the crucial aspects of an IoT ecosystem that defines an entry point for a device or a user in the network. This paper investigates this issue and presents an access control scheme by integrating an IoT network with blockchain technology, thereby arguing to replace the traditional centralized IoT-server architecture with a decentralized one. The blockchain is used with smart contracts to establish a secure platform for device registration. Due to this reason, the IoT devices are first required to register themselves and access the network via contracts thereafter. Moreover, the contracts host a device registry, the access control list, to grant or deny access to devices. This allows the proposed scheme to authorize registered devices only and block unregistered ones, which facilitates the mitigation of rogue devices. To demonstrate the feasibility and improvements of the proposed scheme, security analysis along with in-depth performance evaluation are conducted, where the obtained results indicate its applicability. A case study is also formulated with a comparative analysis that confirms the superior performance of the proposed scheme for low-powered IoT systems.

## 1. Introduction

In recent years, Internet of Things (IoT) has gathered substantial popularity and wide acceptance for low-powered communication among devices [1, 2]. The IoT networks enable connectivity of physical devices via the Internet that can operate, communicate, and actuate autonomously to provide innovative services in a wide array of applications [3]. It is expected that, by the end of the year 2020, almost 50–100 billion devices will be connected to the Internet [4]. These devices would require unconventional and dynamic methodologies to support ultrareliable low-latency communication (URLLC) and enhanced mobile broadband (eMBB) services [5, 6]. Furthermore, there would be a need for novel security mechanisms to ensure the integrity and authenticity of the data.

The interconnection of such a sheer number of devices will inevitably introduce security issues into an IoT-based system as IoT devices are generally resource-constrained in memory, energy, and computational resources, which exacerbate the architectural and security challenges of IoT [7, 8]. To cope with the security issues of IoT networks and prevent future network breaches, several approaches and solutions have been proposed. For instance, some key exchange schemes have been proposed to provide resilience against different kinds of attacks, where key management is

concerned with the generation, storing, and exchange of the keys. Moreover, mechanisms like authentication provide resistance against man-in-the-middle (MITM) and impersonation attacks [9, 10].

With the rise of Bitcoin and cryptocurrency in general, the concept of distributed blockchain databases has received significantly wider attention. This is because a wide range of distributed applications can be built based on the distributed infrastructure of blockchain. One unique variant in this regard is the Ethereum blockchain platform, which includes a Turing-complete programming framework with system state information to realize the so-called smart contracts [11]. Furthermore, the blockchain facilitates a resilient and highly distributed ledger for recording transactions, attributing them to a specific node in a network, and ordering them relative to time. This phenomenon is made possible through a process known as mining, whereby a large number of dedicated high-powered computers running application-specific integrated circuits (ASICs) process the transactions in real time. The miners compete with each other for a small fee in addition to a subsidy in the form of a cryptocurrency or token. Moreover, data is permanently recorded in the blockchain network through a data structure called blocks. Thus, a ledger of past transactions is called the blockchain as it is a chain of blocks that serves to confirm the transactions to the rest of the network [12].

Security protocols in IoT networks are still in a primitive stage and only make use of HTTP, MQTT, and XMPP protocols for routing the messages [13]. With blockchain technology, the issues of key distribution and management are completely solved due to the global unique identifier (GUID) of each IoT device. This would eliminate the handshake procedures and exchange of PKI certificates for communication among IoT devices, thus, leading to a smoother communication experience. Blockchain technology in IoT networks acts as a tool to execute a system of contracts focused on the application of value exchange [14]. Furthermore, there is a multitude of applications that can be run alongside, or in conjunction with, the blockchain-enabled IoT networks, which takes advantage of the large amount of computing power or computational effort generated by the dedicated mining machines. In the next section, we review some of the recent literature in the domain of blockchain-enabled IoT networks.

*1.1. Literature Review.* Research in IoT has recently received worldwide attention such that [7] highlights various challenges in IoT environments and identifies the following avenues for future research directions: architecture and dependencies, creating knowledge and big data, robustness, scaling, privacy, human-in-the-loop, and security in particular. This is because dealing with security attacks is one of the major problems that are prevalent on the Internet [13]. This is deeply problematic for IoT since its operation depends on the Internet connectivity. Moreover, we can define a blockchain as an online and distributed ledger that primarily consists of a list of blocks. Each block is an ordered record of application relevant data and a hash of the preceding block. This enables a system to achieve transparency in its operation and makes a blockchain highly resistant to data tampering. To achieve synchronization of the ledger, different consensus algorithms are used for sharing control across the blockchain network. This contributes to overall increased robustness. Therefore, many applications have adopted it to provide trust-free and decentralized solutions.

The authors of [14] provide a survey of existing blockchain-enabled IoT solutions for permission-less trading in the network. In another work [15], the authors propose a secure signing mechanism for ensuring the integrity of data. The proposed solution makes use of hash-based signing which is more efficient when compared to the existing approaches. The study in [16] proposes SMACS, which is a smart contract access control service. SMACS offloads the burden of expensive access control validation and management operations to an off-chain infrastructure, while only implementing the lightweight token-based access control on a blockchain. Moreover, healthcare is quickly adopting new technologies like artificial intelligence to automate the different modules in a standard clinical workflow for radiation oncology [17, 18]. However, machine learning models are data demanding meaning that abundant data is required for optimal learning, where well-annotated medical data is scarce [19]. In a typical setting, data is collected at a single/different institute(s) and subsequently shared with others as per collaboration agreement. This traditional approach of data sharing is time consuming as it normally requires a centralized database, which is created and maintained by the host institute. Data sharing using blockchain can address this problem. The authors of [20] perform similar studies for private blockchain networks. More specifically, a practical byzantine fault tolerance protocol is proposed. This is an efficient protocol that allows devices to operate even if 33% of the nodes are honest while the rest 66% of nodes become rogue or malicious. The authors of [21] propose a novel approach called Enigma. It is a decentralized platform for guaranteeing the integrity and security of the collected data. The sensitive information in Enigma is stored in an off-chain database with strong encryption that mitigates the impact of cyberattacks. Similarly, a blockchain-based consent model for health data sharing platforms is also discussed in [22].

For smart home applications, the authors of [23] propose a new IoT authorization stack protocol in which the devices are connected to the cloud for exchanging commands with a mobile user. The proposed solution addresses the security leakage issues in an untrusted cloud communication architecture. In a similar work [24], the authors focus on the centrality of blockchain nodes to manage and monitor the IoT devices. Some interesting proposals for private blockchain networks are also provided in [25, 26], wherein the authors created a threat model for evaluating the security protocols. They demonstrated that the intrusion detection systems based on techniques like anomaly behavior analysis can prove quite useful against cyberattacks in IoT networks. Following the same approach, the authors of [27] point out different vulnerabilities and provided solutions for IoT networks.

To ensure the security and integrity of IoT networks, the authors of [28] provide a proof-of-concept implementation of a distributed ledger technology. This was done on multiple Raspberry Pi devices connected to the network in a realistic communication environment. An unclonable solution (used in key management and generation) for low-powered IoT devices and vehicular networks is proposed by [29–31]. Later, an extension of the same was provided in [32, 33], which eliminated the high-cost process of key generation. Quantum security solution for distributed ledger technologies has also been explored in [34]. They propose a one-time signature for reducing the signature time cost and size by 75% and 76%, respectively. The security issues of blockchain-enabled IoT networks for industry 4.0 have also been considered by many studies [35–37], in which different integration challenges and recommendations were highlighted by the authors.

*1.2. Motivation and Contribution.* To help solve and address the aforementioned limitations, we propose a blockchain-based access control scheme for IoT that works in conjunction with smart contracts and achieves distributed and trustworthy access control in an IoT system. Blockchain is used to provide a device registration mechanism via its Public Key Infrastructure (PKI) framework as well as for distributing the control within the network, while smart contracts are used to implement the access control functions with Access Control List (ACL). Moreover, a higher computing capability with lower computation cost for establishing the access control methods is achieved by using smart contracts as opposed to [38–40]. In this backdrop, our work employs blockchain technology for providing access control in IoT networks. More specifically, this paper introduces a scheme for decentralized IoT access control. This is established by integrating the traditional device-to-server communication infrastructure with blockchain and smart contracts. To summarize, the blockchain offers a safe and secure device registration mechanism with its PKI, while the smart contracts enforce the access control functions by using an ACL mechanism. Thus, this paper makes the following contributions to the state of the art:

(i) A novel blockchain-based decentralized IoT access control scheme is proposed. The proposed scheme makes use of the registration platform to register or remove a device in the network.

(ii) The ACL mechanism is designed to authorize registered devices only. The integration of ACL mechanism with the proposed scheme mitigates the impact of rogue devices in an IoT network.

(iii) A comprehensive analysis with a state-of-the-art blockchain-based IoT access control scheme is provided. The results demonstrate the feasibility and superior performance of the proposed scheme.

*1.3. Paper Organization.* The remainder of the paper is organized in the following way. Section 2 describes the IoT-blockchain model while Section 3 explains its operation. Section 4 presents the security analysis of the proposed scheme. Section 5 details its performance evaluation along with its relevant discussion and a comparative analysis. Finally, Section 6 presents the concluding remarks with potential directions for future research.

## 2. IoT-Blockchain Model

This paper presents a blockchain-based access control scheme for IoT that operates in conjunction with smart contracts. The scheme is based on Ethereum [41], a variant of blockchain technology that allows decentralized applications (DApps) to be built atop blockchain along with their corresponding states, which is composed of objects called accounts that have the following fields [41]:

(i) A 20-byte address (i.e., ID)

(ii) A smart contract code that may be empty

(iii) A balance of Ether used to pay transaction fees

(iv) A nonce so that each transaction is processed only once

Furthermore, a state in Ethereum refers to the current data present in the blockchain, whereas a state transition occurs whenever a transaction occurs. Additionally, there are two types of accounts in Ethereum:

(a) Externally owned account (EOA): these are user accounts managed with PKI

(b) Contract: this is a computer program, and its corresponding account has its code and is controlled by the same

Furthermore, by sharing data across the blockchain and committing transactions, the smart contracts can be executed in a decentralized manner. This adheres to their integrity and enables their transparent execution. Besides, there exists a gas limit for each transaction and process within Ethereum, where gas is an analogous word for "resource," i.e., a certain amount of gas for a function means that its execution has that much of resource to use. Therefore, IoT devices have to use very negligible amounts of gas for their operation. It can be interpreted as a cost factor for the IoT devices but it also ensures security by limiting the devices to generate only as many requests as the amount of gas that they have [42].

*2.1. Network Model.* As shown in Figure 1, the IoT-blockchain model consists of seven core components. Thus, the details of these components are provided herewith:

(1) *Server*. It represents a device or a set of devices that is responsible for providing different kinds of services to users and devices of the IoT-blockchain network. Moreover, the server is the host of the IoT-blockchain network; i.e., it initiates a blockchain with the first block but instead of being centralized, servers are decentralized here. This way, the servers act as the trusted hosts since they hold the genesis block that is trusted by all users and devices in the network.

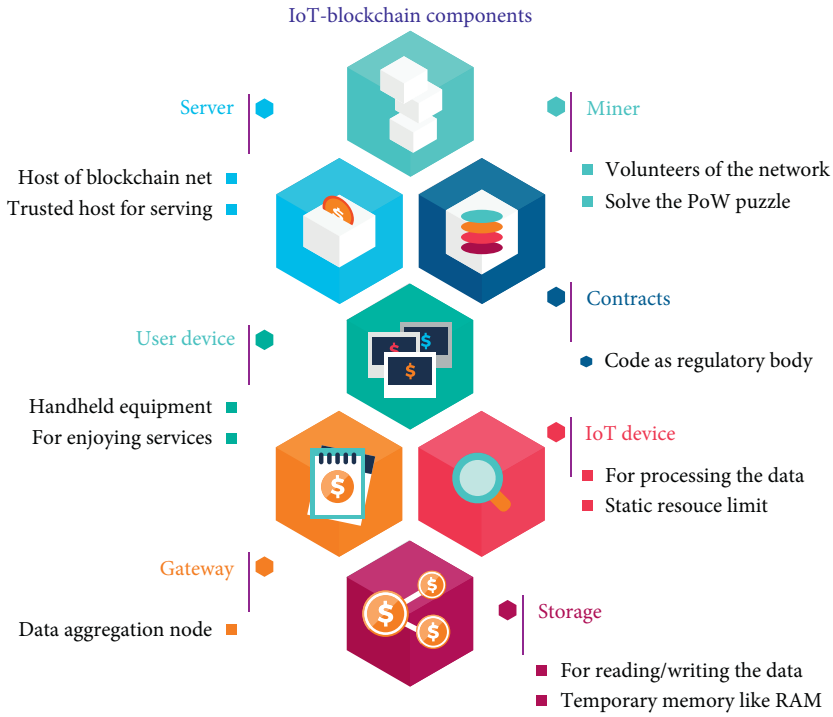



Figure 1: An overview of the IoT-blockchain model and its core components.

Moreover, they may employ permissionless or permissioned consensus protocols to enable interactions between them and the network constituents that include and are not limited to collecting data, processing, querying data from, and/or writing data to storage devices.

(2) *Miner*. It represents the volunteers of the IoT-blockchain network, i.e., miners. They are mainly responsible for solving the PoW puzzles and mine new blocks. Thus, they provide the computing power required by the proposed scheme to operate.

(3) *Smart Contracts*. It represents the computer programs or codes that act as the regulatory bodies in the proposed scheme, i.e., the smart contracts. They enforce the access control functions and host the ACL. Thus, they are responsible for registering and removing devices as well as authorizing them. This way, they can block rogue devices and mitigate their impact.

(4) *User Device*. This represents user setups that include and are not limited to PCs, laptops, and smartphones. A user can conveniently check and enjoy the services provided by the servers in the network using these devices, as well as read data from or write to the storage devices of the network.

(5) *IoT Device*. This represents the things, i.e., devices that are responsible for sensing, processing, and communicating data to the server via gateways. They may also read data from or write to a storage device as well as send control signals to actuators which in turn may operate another device.

(6) *Gateway*. This represents the service agent for IoT devices in the network. The devices can use the gateways for communication; i.e., it provides network connectivity to them via short-range communication technologies and protocols such as Bluetooth, Wi-Fi, and Zigbee. Moreover, a gateway may also provide additional functionalities such as data aggregation and specific security features. Thus, different gateways may be used for different types of devices or a single gateway can also be used for a range of devices, thereby, forming a device cluster.

(7) *Storage*. It represents the reading and/or writing processes of data to storage devices, which may be permanent like read-only memory (ROM) or temporary like random-access memory (RAM). Thus, different data types (e.g., JSON, XML, CSV, etc.) can be written on them such that they can be used by other devices in the network.

*2.2. System Assumptions.* The proposed scheme uses the following system configurations and assumptions:

(i) The scheme uses a proof-of-work (PoW) consensus algorithm for its operation.

(ii) All peers (servers/miners) have a blockchain account that allows them to claim a deployment instance of a smart contract during system initialization and, subsequently, identify themselves as the trusted hosts.

(iii) An adversary/a group of adversaries cannot compromise the blockchain such that peers are not

resource-constrained and control more than 50% of the total computing power.

(iv) Elliptic Curve Cryptography (ECC) with the Elliptic Curve Digital Signature Algorithm (ECDSA) is used to generate the account addresses (IDs) for both IoT devices and peer nodes.

(v) Gateways act as the agents of IoT devices and are responsible for storing their accounts. It is assumed that gateways are physically accessible as well as secure, which makes them unlikely to be compromised. Thus, they can be trusted as agents.

(vi) All peer nodes are assumed to be synchronized on the same blockchain block.

*2.3. Threat Model.* We consider a threat model where the objective of an adversary is to compromise the proposed access control scheme by exploiting a security loophole and gain unauthorized access into the system with his/her rogue device(s), which are just plain malicious in nature by definition. Note that the loophole can include endpoint vulnerabilities, malfunctioning hardware, and "bring your own device" (BYOD). Thus, by compromising the system with weak access control setup, the adversary intends to execute different kinds of cyberattacks on the system, which may include impersonation, resource depletion, sinkhole, denial of service (DoS), distributed DoS (DDoS), birthday, and spoofing. This presents serious security implications: if an adversary successfully enters into a system, he/she can target its specific components to steal information or disrupt the network operations, or in rare cases, permanently damage the whole system. Therefore, effects of rogue devices and devices exhibiting rogue behavior must be mitigated.

## 3. Blockchain-Based IoT Access Control Scheme

The proposed scheme uses ECDSA for generating distinctive IDs for IoT devices and the peer nodes. The smart contracts maintain the ACL and can differentiate between registered and rogue devices. Thus, with the ACL mechanism, each device is required to first register itself with the network using its ID, which is handled through gateways. The registration process will generate a unique ID for each device, which can be used to interact with other devices or peers. These interactions are enabled by the contracts by using ACL. Note that the contracts are hosted by the nodes that deployed them, i.e., peers. Thus, the smart contracts act as the regulatory bodies of the scheme and are responsible for facilitating secure communication between devices and peers. For this purpose, the contracts provide the following ABIs:

deviceAdd: it functions to register a new device using its ID and store it in the ACL. Note that the ID here represents the 20-byte address of the IoT device which is used by this ABI to list the device name in ACL.

deviceDelete: it functions to rescind the access of a device by removing it from the ACL. Similar to deviceAdd ABI, it also requires the 20-byte ID of the device to match against the ACL and remove it thereafter.

sendMessage: this is the enabler of communication with smart contracts. It functions to fetch and return the address of a contract to a device; i.e., if a device wants to send a message, it needs to interact with a contract instance in the network via this ABI.

accessControl: it is the core ABI that is responsible for authorizing and blocking devices with the application of ACL. For this purpose, it first checks if a device is registered in the ACL. Thus, whenever a device calls this ABI to authorize its current access request, it will start the validation process to check the validity of the request according to Algorithm 1, where access$(d_s[n])$ is the access control routine of contracts, request$(d_s[n]$.node$)$ represents a message generated by an IoT device (subject), $d_s$ represents a set of subjects, and ACL is the access control list hosted by the contracts. Thus, this ABI allows the requests of registered devices only and blocks rogue ones, thereby, limiting their impact.

It is worth noting here that only the smart contract creator can add, delete, or update the definitions of these ABIs. Therefore, access control permissions must be carefully considered while designing them.

*3.1. Mining Operation.* To handle the requests (we refer to them as transactions) generated in the IoT-blockchain network, miners (block producers) need to generate blocks efficiently with the optimal time cost. Therefore, they need to complete the following steps: (i) collect, verify, and combine the transactions into a block and mine it; (ii) broadcast the mined block to reach a consensus in the network and append it to the blockchain as the latest block.

We now formulate the miners in our proposed scheme. Let us assume that there are $N$ peer nodes and $M$ miner nodes in the network, where peer nodes represent both miners and servers. Moreover, the set of peer nodes is represented by $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$, where the computing power of node $n_n$, $n = 1, \ldots, N$ is represented by $\Upsilon_n$, respectively. Note that $\Upsilon = \{\Upsilon_1, \Upsilon_2, \ldots, \Upsilon_n\}$ is used here to represent the set of computing power of the network. Thus, $M$ miners represented by $\mathcal{M} = \{m_1, \ldots, m_m, \ldots, m_M\}$, $\mathcal{M} \subseteq \mathcal{N}$, are selected out of $\mathcal{N}$ nodes.

*3.2. Degree of Decentralization.* This paper introduces a novel way to measure the degree of decentralization in the proposed scheme by using Gini coefficient ($G$); it is well studied as a measurement for inequality of wealth or income [43]. Due to its accuracy in evaluating inequality, $G$ has been employed in many fields that include and are not limited to capturing contrast intensity [44], system fairness [45], and resource difference degree [46]. For further details on $G$, we direct the reader to Appendix A. Thus, we measure the decentralization of our scheme by considering the distribution of computing power among the miners. To formulate

**Function**: access($d_s[n]$)
**Input**: request($d_s[n]$.node)
**Output**: allow, block
(1) **while** Input **do**
(2) **for** $n$ in $d_s, d_s \in S \forall n = 1, \ldots, s$ **do**
(3) **if** $d_s[n]$.node is in ACL **then**
(4) allow
(5) **else**
(6) block

Algorithm 1: Establishing access control policies with smart contracts.

this, $G$ for miners with respect to (w.r.t.) computing power distribution can be calculated by in the following way:

$$G(\Upsilon) = \frac{\sum_{m_i \in \mathcal{M}} \sum_{m_j \in \mathcal{M}} \left| \Upsilon_i - \Upsilon_j \right|}{2 \sum_{m_i \in \mathcal{M}} \sum_{m_j \in \mathcal{M}} \Upsilon_i} = \frac{\sum_{m_i \in \mathcal{M}} \sum_{m_j \in \mathcal{M}} \left| \Upsilon_i - \Upsilon_j \right|}{2M \sum_{m_i \in \mathcal{M}} \Upsilon_i}, \tag{1}$$

The values of $G$ are within the range $[0, 1]$, where 0 denotes full decentralization while 1 denotes the opposite (full centralization), respectively. Using this formulation, we can observe that the more uniform or decentralized the distribution of computing power is, the closer $G$ is to 0. Figure 2 describes the decentralization performance of the proposed IoT-blockchain scheme. Different from [47], where decentralization performance of a blockchain is measured by the number of miners, a more general metric, $G$, is used here to capture the degree of decentralization w.r.t. the computing power distribution among miners. It can be seen from the figure that as the threshold of $G$ decreases, the Lorenz curve gradually approaches the line of ideal decentralization, thereby, making the blockchain more decentralized. Note that Lorenz curve details are given in Appendix A. This demonstrates that $G$ is an effective metric that can be used to measure the decentralization degree of blockchain-based systems. Similarly, $G$ can also be calculated for other aspects of a blockchain quantitatively.

## 4. Security Analysis

This section presents the security analyses of the proposed scheme by discussing the following factors.

*4.1. Distributed Servers.* Traditional IoT systems primarily rely on a centralized cloud server that is responsible for managing IoT devices and handling the majority of the computation and decision operations. Although a cloud server may in reality be replicated for authentication and decision processes, the system can still be considered as a single entity. This presents us with a serious security concern; i.e., the whole system can be potentially compromised if an adversary gains access to the server. Thus, the proposed scheme eliminates this concern by distributing the computation resources among miners. This results in high-security fidelity and enables a system to continue operation even if one or more of its peers cease to operate. Moreover, by distributing the computation in this manner, the resources required by a server can be relaxed. This will likely result in a situation where adversaries will consume mainly their resources to perform any malicious activity or attack.

*4.2. Trust-Free System Operation.* A typical IoT system operates on trust which is normally established via third parties that work as the middlemen between the devices and the server. These third parties have their associated costs in terms of labor and latency, where centralized IoT systems have to pay as trust a key security requirement for reliable network operation. The proposed scheme eliminates this reliance since it does not require any intermediary to guarantee its operation [48]. Moreover, a PoW distributed consensus protocol is used instead, which allows the network to reach consensus, and, thus, trust-free system operation is realized.

*4.3. Rogue Device Mitigation.* A conventional IoT system usually lacks a device registration mechanism for effectively handling the devices. By using the ACL mechanism in the proposed scheme, the smart contracts authorize each device whenever they generate a request. Thus, when a device sends a message, it is checked against the ACL and granted access only if it is registered in it. This way, the proposed scheme can establish a defense mechanism against rogue devices and, therefore, mitigate their impact on the IoT-blockchain system.

*4.4. Shorter Key Lengths.* The authenticity of messages in the proposed scheme is guaranteed via digital signatures by using ECDSA [49]. This ensures data integrity; i.e., data can be sent by registered devices only. For its feasibility, we present a comparison between ECC, Rivest–Shamir–Adleman (RSA), Digital Signature Algorithm (DSA), and Diffie–Hellman (DH) in Table 1 [49, 50]. We can observe that for an 80-bit strength of a system's security, ECC needs only 160 bits while all of the other algorithms need 1024 bits. Similarly, for a 256-bit strength, ECC needs 521 bits compared to 15360 bits needed by the others. This proves that ECC needs shorter key lengths when compared with the other cryptographic algorithms to achieve similar security strength levels. This helps reduce the overhead in our scheme as smaller key lengths translate into lower computational overhead [51].

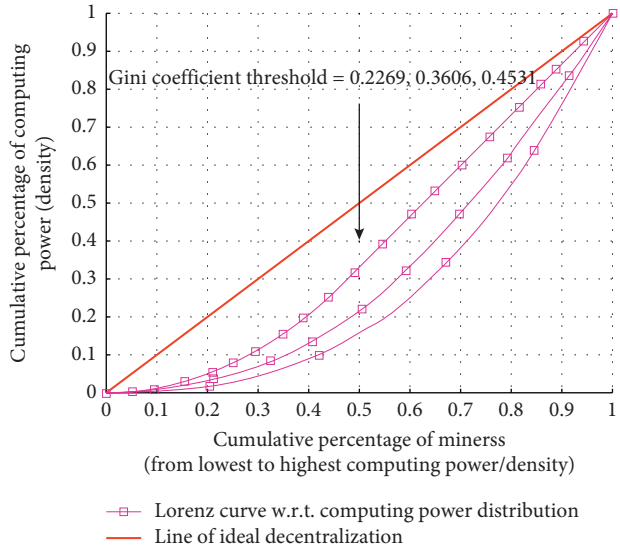



Figure 2: Quantifying the degree of decentralization performance of miners in a blockchain-enabled IoT network. The Gini coefficient here signifies how the distribution of computing power is made among the miners, i.e., whether it is in a centralized or decentralized manner. It can be seen that as the coefficient increases, the performance worsens and is more inclined towards centralization.

Table 1: Security strength comparison of key size combinations for various cryptographic algorithms.

| Security | Key size (bits) | | | Ratio |
|---|---|---|---|---|
| | Symmetric encryption algorithm | ECC | RSA/DH/DSA | |
| 80 | Skipjack | 160–223 | 1024 | |
| 112 | 3DES | 224–255 | 2048 | |
| 128 | AES-128 | 256–383 | 3072 | 1:6–30 |
| 192 | AES-192 | 384–511 | 7680 | |
| 256 | AES-256 | 512–more | 15360 | |

*4.5. Blockchain Robustness.* The quintessential factor of our scheme is the employment of blockchain technology in it. Therefore, it is of paramount importance to guarantee its security. For this purpose, let us consider a case where an adversary $A$ tries to create a dishonest chain faster than the honest chain. Note that the honest chain is hosted by the honest miners in the proposed scheme and we assume that they always control more than 50% of the total computational resources. Moreover, we say that $A$ wants to catch up with the honest chain (we say $i$ blocks behind) and, therefore, be able to invalidate it with his/her dishonest chain. Thus, the probability that $A$ catches up from $i$ blocks behind the honest chain is analogous to a Gambler's Ruin problem. Let us consider a player who starts to play with unlimited credit at a given deficit. The player potentially plays an infinite number of trials and tries to reach a breakeven point. Then, the probability that $A$ ever reaches breakeven, or in other words, that $A$ ever catches up with the honest chain can be represented as [49]

$$Q_i = \begin{cases} 1 & \text{if } p \le q \\ \left(\frac{q}{p}\right)^i & \text{if } p > q \end{cases}, \tag{2}$$

where $q$ represents the probability that $A$ finds the next block, $p$ represents the probability that an honest miner in the IoT-blockchain network finds the next block, and $Q_i$ is the probability that $A$ will catch up with the honest chain from $i$ blocks behind. This is visually illustrated in Figure 3 that confirms the infeasibility of this attack as long as the honest miners have more than 50% of the total computing power. It can be seen that for values $p = 1, 0.9, 0.8, 0.7, 0.6$, $Q_i$exponentially decreases with the increasing number of blocks of deficit. To elaborate, immediately after just 10 blocks, $Q_i$ reduces to 0. Moreover, for the average value $p = 0.5$, $Q_i$ increases to 1, which signifies again that whoever in the IoT-blockchain network controls more than 50% of the total computational capacity, controls the blockchain. However, given our assumption $p > q$, $Q_i$ exponentially drops with the increasing number of blocks of deficit $A$ has to catch up.

## 5. Performance Evaluation

For evaluating our scheme, we realized its implementation by designing a smart contract in Solidity which is the programming language for writing smart contracts. Subsequently, simulations were conducted to validate the

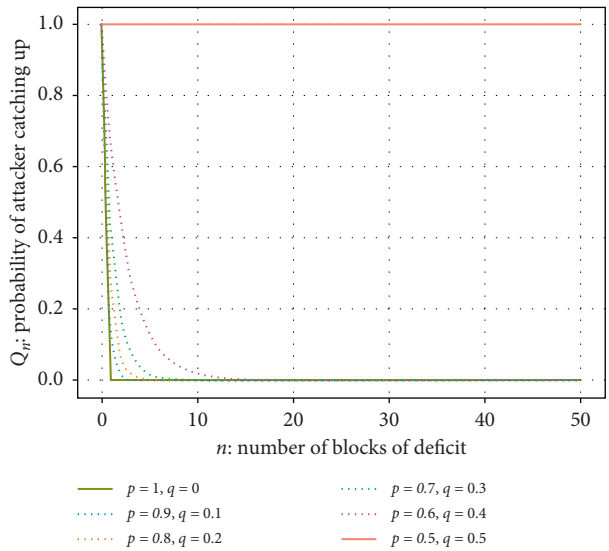



Figure 3: The probability of an adversary ($Q_i$) trying to reach a breakeven point, i.e., mine an alternate and dishonest chain in a blockchain by competing against an honest chain that has the computing power of at least 51% of honest miners.

interactions and access control functions between subject–object pair nodes.

*5.1. Setup.* We conducted the simulations using a PC setup with Ubuntu OS on virtual machine client, Oracle VM VirtualBox. Subsequently, the shell scripting environment of Terminal was used for verifying the access control functions. The specifications of the PC were Intel® Core™ i7-7700HQ CPU @ 2.80 GHz (8 CPUs), 16384 MB RAM, NVIDIA GeForce GTX 1060 with 6052 MB memory and 1024 GB HDD + 128 GB SSD of storage. Moreover, the nodes were instantiated using the Ethereum Go client (Geth) according to Algorithm 2. Note that Geth is a command-line interface (CLI) implemented in the Go language for Ethereum development purposes. Thus, separate nodes were used to simulate the subject–object pair interactions with the distributed contracts. Furthermore, the contracts were written and compiled using the Remix integrated development environment (IDE), a browser-based IDE for Solidity, where an outlook of Remix IDE console can be seen in Figure 4. Note that the contracts are deployed at the object side as the blockchain is hosted by the objects.

*5.2. Deployment Cost.* The cost of performing a task in the Ethereum platform is measured in terms of gas, i.e., for every operation executed in Ethereum, there exists a specified gas cost. Gas is measured in wei and is equal to $1\,\text{wei} = 10^{-18}$ ether. Thus, we can observe that the more complex a task is, the more gas it will require. The gas consumption estimates for the proposed scheme are as follows: the amount required for deploying the contract is 985200 while that for executing it is 21128.

*5.3. Experiments.* Once the subject–object pair nodes are initialized and the contracts are deployed at the object nodes (we refer to them as server), interaction is now possible with the contract from the subject nodes to simulate IoT-server interactions. Thus, the access control results of the proposed scheme are summarized in Figure 5 as follows: the mining of the contract instance for its address by an object node can be seen in Figure 5(a), whereas the functions used in the proposed scheme are demonstrated in Figure 5(b). It can be seen that a subject node with address 0x c7d9 2270 5023 924b 2073 16bc 7fec f794 f608 020a is first registered in the ACL by the contract and then authorized for a message it sends as well as it is subsequently removed and unauthorized. Finally, Figure 5(c) shows the interactions between a subject–object pair node. This demonstrates and confirms the functions of the proposed scheme.

*5.4. Comparative Analysis.* This paper compares its scheme with the state-of-the-art scheme [39] that presents a similar contract-based access control mechanism for IoT. A summary of the comparison results is documented in Table 2. The authors in [39] design their scheme with three smart contracts that include multiple access control contracts (ACC), judge contract (JC), and register contract (RC). The operation of their scheme is defined in the following way:

(i) ACC is responsible for enforcing one access control method at a time for a subject–object pair. It also checks and keeps into account the behavior exhibited by a subject.

(ii) JC is responsible for subject behavior management based on the reports of ACC. It also provides functions (e.g., register, update, and delete) to manage the subjects.

(iii) RC offers a storage hub for the scheme; i.e., it is responsible for storing ACC and JC contracts together with the methods associated with them (access control and subject behavior monitoring).

```
while simulation do
  for i in d_o, d_o ∈ O ∀ i = 1, ..., o do
    genesis (.json) ⟵ define
    d_o[i] ⟵ create node
    for j ⟵ 1, i do
      d_o^i[j].node ⟵ deploy contract
  for n in d_s, d_s ∈ S ∀ n = 1, ..., s do
    genesis (.json) ⟵ define
    d_s[n] ⟵ create node
  while d_o &d_s do
    contract ⟵(message())⟵ d_s[n].node
    d_o^i[j].node ⟵ contract
    if request (d_s[n].node) then
      Algorithm 1 ⟵ call
```

Algorithm 2: Initializing the subject–object pair nodes.

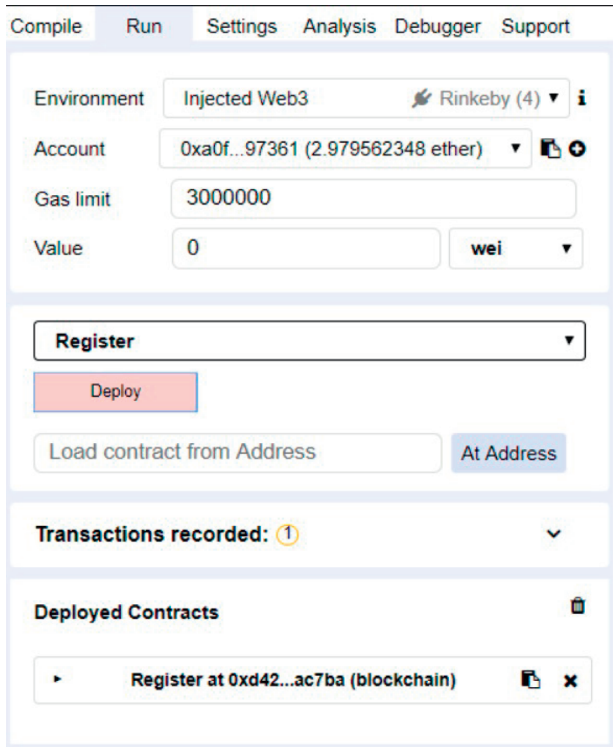



Figure 4: The user interface console of Remix, which is an IDE that is predominantly used in Ethereum to design and compile a smart contract. It offers different settings for analysing and debugging a contract as well as study the execution costs associated with it. The account field represents the address of the contract while the gas limit represents its execution limit among other parameters.

Moreover, [39] does not particularly emphasize on rogue device mitigation, which limits its application and feasibility. It also fails to explain the decentralization degree of miners in a blockchain-enabled IoT network.

In contrast, the proposed scheme establishes the same access control methods by using only one contract with a significantly lesser cost of execution. It manages the malicious behavior of devices via an access control list, which blocks and mitigates the impact of rogue devices. This way, the scheme ensures network reliability by only allowing registered devices to communicate. Furthermore, it discusses the decentralization degree of miners in an

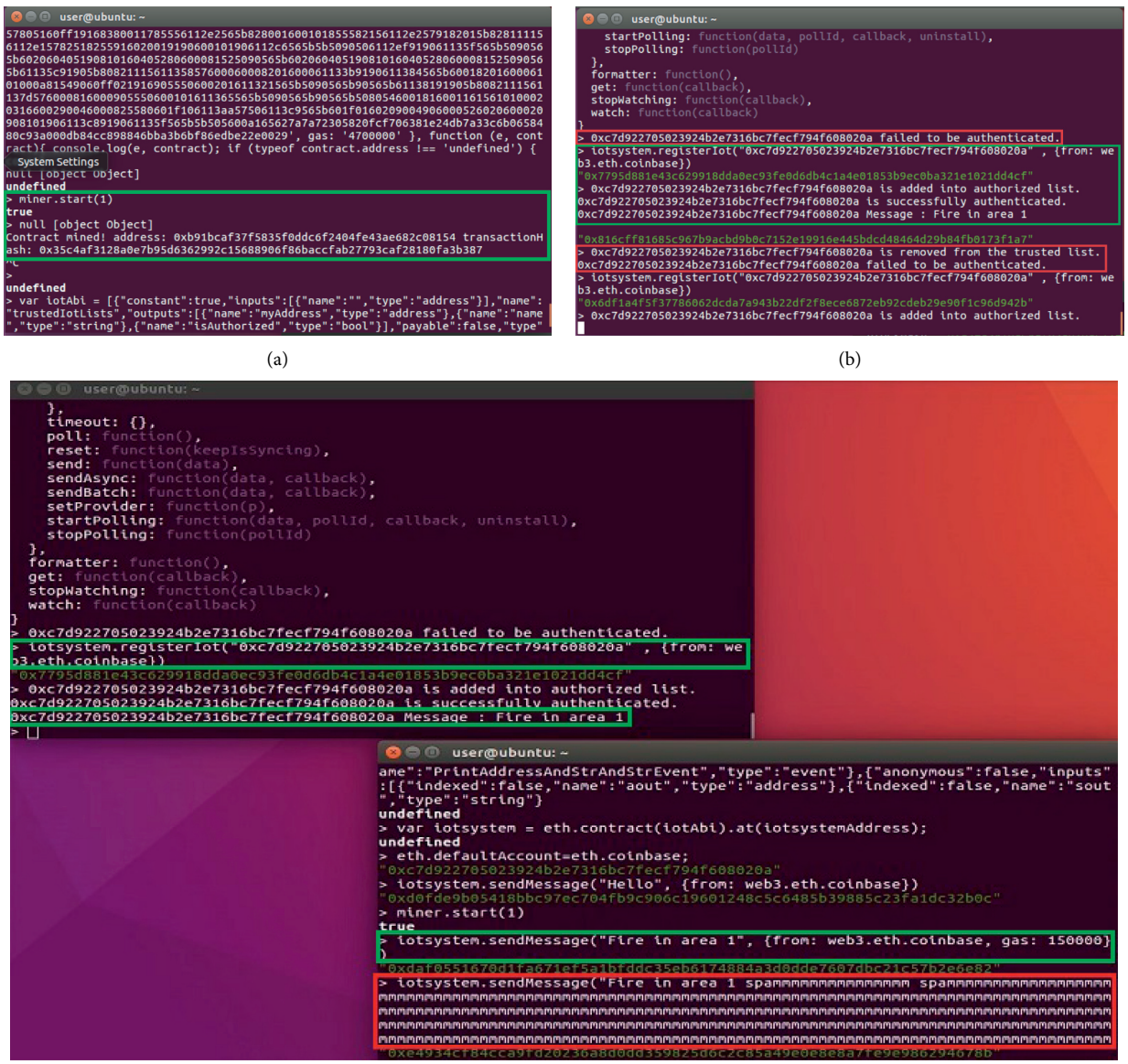



Figure 5: Illustrating the proposed blockchain-based decentralized IoT access control scheme through its implementation in Geth client of Ethereum. Note that the green-colored frames represent a successful operation while the red-colored ones represent a failed operation. (a) Deploying a contract instance on a server (object) node in Geth client. (b) Testing the access control functions on a IoT device node. (c) Demonstrating the interactions between a subject–object pair via a smart contract with a set of message requests.

Table 2: A comparison summary of the proposed scheme with [39].

| Attributes | [39] | Proposed | Improvement (%) |
|---|---|---|---|
| No. of contracts | 3 | 1 | 66 |
| Deploying cost (gas) | 5484074 | 985200 | 82 |
| Execution cost (gas) | 90000 | 21128 | 76 |
| Platform | Ethereum | Ethereum | — |
| Data access | Contract-based | Contract-based | — |

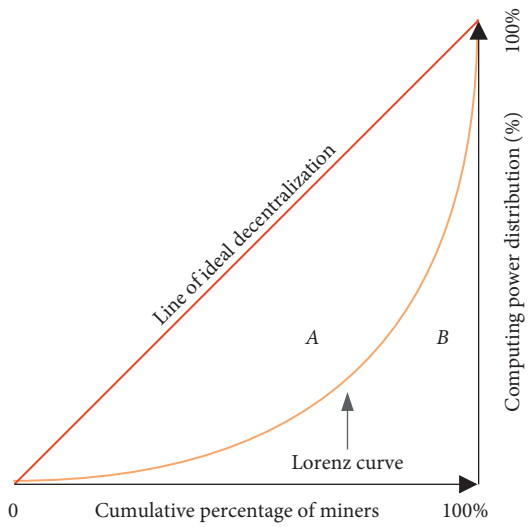



Figure 6: An illustration of a Lorenz curve-based Gini coefficient for quantifying the degree of decentralization of miners in a blockchain. The $x$-axis represents the increasing number of miners while the $y$-axis represents the increasing computing power. The Lorenz curve here represents how well the computing power is divided among miners. The line of ideal decentralization is realized when all of the miners have an equal share of computing resources.

IoT-blockchain model using the Gini coefficient. Thus, it can be seen from Table 2 that our scheme outperforms [39] by offering superior performance with low execution cost.

## 6. Conclusion

This paper investigated the shortcomings of providing access control to devices in a traditional IoT-server communication-based model and presented a blockchain-based access control scheme to mitigate the impact of rogue devices in IoT environments. The proposed scheme uses blockchain in conjunction with smart contracts to provide a secure registration platform for IoT devices. It is also able to distinguish between registered and rogue devices via the application of access control list. To demonstrate the feasibility of the proposed scheme, a security analysis was presented. Additionally, a performance evaluation along with a comparative analysis was also performed for providing access control in a blockchain-based IoT network, which confirms the improvement of the scheme in achieving decentralized IoT access control.

It is noteworthy here that although the results provided in this paper demonstrate the feasibility of the proposed scheme, it can be improved and extended in a number of ways.

Future studies can focus on integrating machine/deep learning techniques to further mitigate the impact of rogue devices in IoT networks. For instance, neural networks can be trained on real data to better identify the attributes of rogue devices and facilitate in providing safeguarding measures together with decentralized access control. The obtained results can also be improved by adopting data sharing and power-domain nonorthogonal multiple access techniques for applications in 5G and beyond. The proposed scheme can be used with resource allocation in cyberphysical systems. For instance, a device can be considered as a subject, which is registered with the network, that requires resource assignment for application-specific purposes, e.g., edge computation offloading. Transaction fees in traditional blockchain platforms remain an open issue that needs to be addressed. Therefore, the proposed implementation can be extended to such platforms where transaction fees are not required. The applicability and feasibility of the proposed scheme can be studied under different consensus protocols of blockchain technology, which will subsequently help identify the applicability of such protocols for providing decentralized and trust-free access control in IoT.

These interesting yet challenging approaches to access control are some of the potential future research avenues that will eventually be discussed and addressed in future studies.

## Appendix

## A Gini Coefficient

The Gini index or Gini coefficient is a statistical measure of distribution which was first introduced in 1912 by the Italian statistician Corrado Gini [43]. It is primarily used as a gauge of economic inequality and measuring income or wealth distribution among a population. The index ranges from 0 to 1 (or 0–100%), with 0 representing perfect equality and 1 representing perfect inequality. Moreover, there are two commonly accepted definitions of the Gini coefficient.

The first definition is based on Lorenz curve which plots the proportion of the total income of population ($y$-axis) against the cumulative share of income earned by the population ($x$-axis). Therefore, the Gini index can be defined as a ratio of the areas area$(A)$/area$(A + B)$ [45, 46] (an illustration for reference is presented in Figure 6). Using this

argument, we can deduce from the figure that the Gini index can be interpreted as the degree of deviation from the line of ideal decentralization.

The second definition is defined as "half of the relative mean absolute difference," which is mathematically equivalent to the definition of Lorenz curve [52]. The mean absolute difference can be calculated by the average absolute difference of all pairs of people in a population, while the relative mean absolute difference is simply the mean absolute difference divided by the relative average. Therefore, the expression of Gini coefficient can be given as [44, 52]

$$\mathcal{G} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|x_i - x_j\right|}{2\sum_{i=1}^{n}\sum_{j=1}^{n}x_i} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|x_i - x_j\right|}{2n\sum_{i=1}^{n}x_i}, \tag{A.1}$$

where $x_i$ is the wealth or income of person $i$ while $n$ is the total number of persons. Thus, this paper uses this definition to calculate the Gini coefficient for measuring the degree of decentralization in a blockchain-enabled IoT network.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

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

## References


[1] M. N. Aman, K. C. Chua, and B. Sikdar, "Hardware primitives-based security protocols for the internet of things," in *Cryptographic Security Solutions for the Internet of Things*, M. T. Banday, Ed., pp. 117–141, IGI Global, Hershey, PA, USA, 2019.

[2] M. Rehan and M. Rehmani, *Blockchain-enabled Fog and Edge Computing: Concepts, Architectures and Applications: Concepts, Architectures and Applications*, CRC Press, Boca Raton, FL, USA, 2020.

[3] F. Jameel, M. A. Javed, S. Zeadally, and R. Jantti, "Efficient mining cluster selection for Blockchain-based cellular V2X communications," 2020, http://arxiv.org/abs/2007.01052.

[4] C. Perera, R. Ranjan, L. Wang, S. U. Khan, and A. Y. Zomaya, "Privacy of big data in the internet of things era," 2014, http://arxiv.org/abs/1412.8339.

[5] A. Chaer, K. Salah, C. Lima, P. P. Ray, and T. Sheltami, "Blockchain for 5G: opportunities and challenges," in *Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, IEEE, Waikoloa, HI, USA, December 2019.

[6] F. Jameel and S. A. Hassan, *Wireless-Powered Backscatter Communications for Internet of Things*, Piscataway, NJ, USA, 2020.

[7] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[8] F. Jameel, U. Javaid, B. Sikdar, I. Khan, G. Mastorakis, and C. X. Mavromoustakis, "Optimizing Blockchain networks with artificial intelligence: towards efficient and reliable IoT applications," in *Convergence Of Artificial Intelligence And the Internet of Things*, pp. 299–321, Springer, Berlin, Germany, 2020.

[9] M. N. Aman, U. Javaid, and B. Sikdar, "A privacy-preserving and scalable authentication protocol for the internet of vehicles," *IEEE Internet of Things Journal*, p. 1, 2020.

[10] T. Xu, J. B. Wendt, and M. Potkonjak, "Security of IoT systems: design challenges and opportunities," in *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 417–423, San Jose, CA, USA, November 2014.

[11] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the internet of things: a review," in *Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering*, vol. 3, pp. 648–651, Hangzhou, China, March 2012.

[12] M. N. Aman, M. H. Basheer, and B. Sikdar, "Data provenance for iot with light weight authentication and privacy preservation," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 441–510 457, 2019.

[13] S. Ravi, A. Raghunathan, and S. Chakradhar, "Tamper resistance mechanisms for secure embedded systems," in *Proceedings of the 17th International Conference on VLSI Design Proceedings*, pp. 605–611, Mumbai, India, January 2004.

[14] A. Sedrati, M. A. Abdelraheem, and S. Raza, "Blockchain and IoT: mind the gap," in *Interoperability, Safety And Security in IoT*, pp. 113–122, Springer, Berlin, Germany, 2017.

[15] A. Malik, S. Gautam, S. Abidin, and B. Bhushan, "Blockchain technology-future of IoT: including structure, limitations and various possible attacks," vol. 1, pp. 1100–1104, in *Proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, vol. 1, IEEE, Kannur, India, July 2019.

[16] B. Liu, S. Sun, and P. Szalachowski, "Smacs: smart contract access control service," in *Proceedings of the 2020 50th Annual IEEE/IFIP International Conference On Dependable Systems And Networks (DSN)*, pp. 221–232, Valencia, Spain, 2020.

[17] U. Javaid, D. Dasnoy, and J. A. Lee, "Multi-organ segmentation of chest ct images in radiation oncology: comparison of standard and dilated unet," in *International Conference on Advanced Concepts for intelligent Vision Systems*, pp. 188–199, Springer, Berlin, Germany, 2018.

[18] U. Javaid, K. Souris, D. Dasnoy, S. Huang, and J. A. Lee, "Mitigating inherent noise in Monte Carlo dose distributions using dilated U-Net," *Medical Physics*, vol. 46, no. 12, pp. 5790–5798, 2019.

[19] U. Javaid, D. Dasnoy, and J. A. Lee, "Semantic segmentation of computed tomography for radiotherapy with deep learning: compensating insufficient annotation quality using contour augmentation," in *Medical Imaging 2019: Image Processing*, vol. 10949, p. 109492P, SPIE Press, Washington, DC, USA, 2019.

[20] S. Gao, T. Yu, J. Zhu, and W. Cai, "T-pbft: an eigentrust-based practical byzantine fault tolerance consensus algorithm," *China Communications*, vol. 16, no. 12, pp. 111–123, 2019.

[21] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: decentralized computation platform with guaranteed privacy," 2015, http://arxiv.org/abs/1506.03471.

[22] V. Jaiman and V. Urovi, "A consent model for Blockchain-based health data sharing platforms," *IEEE Access*, vol. 8, pp. 143734–143745, 2020.

[23] B.-C. Chifor, I. Bica, V.-V. Patriciu, and F. Pop, "A security authorization scheme for smart home Internet of Things devices," *Future Generation Computer Systems*, vol. 86, pp. 740–749, 2018.

[24] M. von Maltitz, S. Smarzly, H. Kinkelin, and G. Carle, "A management framework for secure multiparty computation in dynamic environments," in *Proceedings of the NOMS 2018-*

*2018 IEEE/IFIP Network Operations And Management Symposium*, pp. 1–7, IEEE, Taipei, Taiwan, April 2018.

[25] K. Košťál, P. Helebrandt, M. Belluš, M. Ries, and I. Kotuliak, "Management and monitoring of IoT devices using blockchain," *Sensors*, vol. 19, no. 4, p. 856, 2019.

[26] J. Pacheco and S. Hariri, "IoT security framework for smart cyber infrastructures," in *Proceedings of the 2016 IEEE 1st International Workshops on Foundations and Applications of Self∗ Systems (FAS∗ W)*, pp. 242–247, IEEE, Augsburg, Germany, September 2016.

[27] H. Lin and N. Bergmann, "IoT privacy and security challenges for smart home environments," *Information*, vol. 7, no. 3, p. 44, 2016.

[28] L. Hang and D.-H. Kim, "Design and implementation of an integrated iot blockchain platform for sensing data integrity," *Sensors*, vol. 19, no. 10, p. 2228, 2019.

[29] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, no. 8, p. 352, 2018.

[30] U. Javaid, M. N. Aman, and B. Sikdar, "BlockPro: blockchain based data provenance and integrity for secure IoT environments," in *Proceedings of the 1st Workshop On Blockchain-Enabled Networked Sensor Systems*, pp. 13–18, ACM, New York, NY, USA, 2018.

[31] U. Javaid, M. N. Aman, and B. Sikdar, "Drivman: driving trust management and data sharing in vanets with blockchain and smart contracts," in *Proceedings of the 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1–5, Kuala Lumpur, Malaysia, May 2019.

[32] S. S. Arslan, R. Jurdak, J. Jelitto, and B. Krishnamachari, "Advancements in distributed ledger technology for internet of things," *Internet of Things*, vol. 9, Article ID 100114, 2020.

[33] M. Á. Prada-Delgado, I. Baturone, G. Dittmann, J. Jelitto, and A. Kind, "PUF-derived IoT identities in a zero-knowledge protocol for blockchain," *Internet of Things*, vol. 9, Article ID 100057, 2020.

[34] F. Shahid, A. Khan, and G. Jeon, "Post-quantum distributed ledger for internet of things," *Computers & Electrical Engineering*, vol. 83, Article ID 106581, 2020.

[35] Q. Wang, X. Zhu, Y. Ni, L. Gu, and H. Zhu, "Blockchain for the IoT and industrial IoT: a review," *Internet of Things*, vol. 10, Article ID 100081, 2019.

[36] H. F. Atlam and G. B. Wills, "Intersections between IoT and distributed ledger," *Advances in Computers, Role of Blockchain Technology in IoT Applications*, vol. 115, pp. 73–113, 2019.

[37] F. Jameel, U. Javaid, W. U. Khan, M. N. Aman, H. Pervaiz, and R. Jäntti, "Reinforcement learning in blockchain-enabled IIoT networks: a survey of recent advances and open challenges," *Sustainability*, vol. 12, no. 12, p. 5161, 2020.

[38] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "Fairaccess: a new blockchain-based access control framework for the internet of things," *Security and Communication Networks*, vol. 9, no. 18, pp. 5943–5964, 2016.

[39] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2019.

[40] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: the case study of a smart home," in *Proceedings of 2017 IEEE Interence Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 618–623, Kona, HI, USA, March 2017.

[41] V. Buterin, "Ethereum: a next-generation smart contract and decentralized application platform," 2014, https://github.com/ethereum/wiki/wiki/White-Paper.

[42] U. Javaid, A. K. Siang, M. N. Aman, and B. Sikdar, "Mitigating loT device based DDoS attacks using blockchain," in *Proceedings Of the 1st Workshop On Cryptocurrencies And Blockchains for Distributed Systems, ser. CryBlock'18*, pp. 71–76, ACM, New York, NY, USA, 2018.

[43] C. Gini, "Variability and mutability," *Journal of the Royal Statistical Society*, vol. 76, pp. 619–622, 1913.

[44] Z. Lin, F. Wen, Y. Ding, and Y. Xue, "Data-driven coherency identification for generators based on spectral clustering," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1275–1285, 2018.

[45] L. Dai, Y. Jia, L. Liang, and Z. Chang, "Metric and control of system fairness in heterogeneous networks," in *Proceedings of the 2017 23rd Asia-Pacific Conference on Communication (APCC)*, pp. 1–5, Perth, Australia, December 2017.

[46] D. Wu, G. Zeng, L. Meng, W. Zhou, and L. Li, "Gini coefficient-based task allocation for multi-robot systems with limited energy resources," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 155–168, 2018.

[47] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile Blockchain meets edge computing: challenges and applications," 2017, http://arxiv.org/abs/1711.05938.

[48] R. Beck, J. Stenum Czepluch, N. Lollike, and S. Malone, "Blockchain -the gateway to trust-free cryptographic transactions," in *Proceedings of the Twenty-Fourth European Conf. On Information Systems (ECIS)*, pp. 1–14, Springer Publishing Company, Istanbul, Turkey, 2016.

[49] U. Javaid, M. N. Aman, and B. Sikdar, "A scalable protocol for driving trust management in internet of vehicles with blockchain," *IEEE Internet of Things Journal*, p. 1, 2020.

[50] K. Maletsky, "Rsa vs ECC comparison for embedded systems," 2015, http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8951-CryptoAuth-RSA-ECC-Comparison-Embedded-Systems-WhitePaper.pdf.

[51] M. N. Aman, S. Taneja, B. Sikdar, K. C. Chua, and M. Alioto, "Token-based security for the internet of things with dynamic energy-quality tradeoff," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2843–2859, 2018.

[52] A. Sen, *On Economic inequality*, Oxford University Press, Oxford, UK, 1977.

Hindawi
Mobile Information Systems
Volume 2020, Article ID 8862920, 10 pages
https://doi.org/10.1155/2020/8862920

*Research Article*

# Achieving Message-Encapsulated Leveled FHE for IoT Privacy Protection

**Weiping Ouyang**,[1] **Chunguang Ma**,[1,2] **Guoyin Zhang**,[1] **and Keming Diao**[1]

[1]*Harbin Engineering University, Harbin, China*
[2]*Shandong University of Science and Technology, Qingdao, China*

Correspondence should be addressed to Weiping Ouyang; ouyangweiping@hrbeu.edu.cn

Received 15 April 2020; Revised 15 July 2020; Accepted 16 September 2020; Published 14 October 2020

Academic Editor: Kathiravan Srinivasan



The rapid development of the Internet of Things has made the issue of privacy protection even more concerning. Privacy protection has affected the large-scale application of the Internet of Things. Fully Homomorphic Encryption (FHE) is a newly emerging public key encryption scheme, which can be used to prevent information leakage. It allows performing arbitrary algebraic operations on data which are encrypted, such that the operation performed on the ciphertext is directly transformed into the corresponding plaintext. Recently, overwhelming majority of FHE schemes are confined to single-bit encryption, whereas how to achieve a multibit FHE scheme is still an open problem. This problem is partially (rather than fully) solved by Hiromasa-Abe-Okamoto (PKC′15), who proposed a packed message FHE scheme which only supports decryption in a bit-by-bit manner. Followed by that, Li-Ma-Morais-Du (Inscrypt′16) proposed a multibit FHE scheme which can decrypt the ciphertext at one time, but their scheme is based on dual LWE assumption. Armed with the abovementioned two schemes, in this paper, we propose an efficient packed message FHE that supports the decryption in two ways: single-bit decryption and one-time decryption.

## 1. Introduction

In recent years, the Internet of Things (IoT) has become an attractive system paradigm to drive a substantive leap on goods and services and has been widely used in intelligent transportation, intelligent power grid, environmental monitoring and perception, intelligent home appliances, and other fields. It covers traditional equipment to general household equipment, which brings more efficiency and convenience to the users. Because many of the data transmitted in the Internet of Things are confidential information or personal privacy information, it usually needs to be encrypted first. With more and more encrypted data stored on the server, it is very frequent for us to retrieve and process these data. Although there are some algorithms for retrieving encrypted data, they are only suitable for small-scale data, and the cost is too high. The encrypted data retrieval method based on the Fully Homomorphic Encryption (FHE) can solve this problem. By directly retrieving the encrypted data, it not only ensures that the retrieved data will not be analyzed, but also carries out homomorphic operation on the retrieved data without changing the sequence of the corresponding plaintext. It can not only protect the user's data security but also improve the retrieval efficiency. Since the first introduction of Gentry in 2009, the construction and optimization of the Fully Homomorphic Encryption scheme have been paid special attention by researchers. However, most of the existing Fully Homomorphic Encryption schemes only allow cryptographic calculations for a single bit, and the efficiency is not satisfactory. Although the cascading (or simple combination) approach can be used to implement message-encapsulated calculations, the performance of such a simple message-encapsulated FHE is not ideal.

In an application scenario, in many cases, it is necessary to calculate data of multiple bits at a time, and thus, constructing an efficient Message-encapsulation Fully Homomorphic encryption becomes an urgent requirement. At

present, the research in this area has made initial progress [1, 2], which has increased the efficiency of FHE to a certain extent, but comprehensively, its efficiency still needs to be improved. Specifically, the following are considered:

(1) Brakerski's scheme [1] is constructed on the basis of the Brakerski's [3] scheme and is a typical representative of the second generation of FHE. But, the latter scheme needs to implement homomorphic calculations by calculating the evaluation key, which increases the computational cost.

(2) Hiromasa-Abe-Okamoto (HAO) [2] is based on the GSW [4] scheme and is a typical representative of the third generation of FHE. HAO constructs a message-encapsulation FHE scheme in the form of encapsulated messages, but it cannot implement one-time decryption and only decrypts the ciphertext bit-by-bit, so the scheme is still very inefficient.

An important question arises: Besides those mentioned above, is it possible to design an efficient method to decrypt the ciphertext of the message-encapsulation GSW-FHE scheme at one time?

Li et al. [5] used dual Regev [6] to construct a public key with multiple instances of the small short integer solution (SIS). Inspired by this work, we will construct public keys with multiple instances of LWEs (Learning with errors), and this constructs a Message-Encapsulation FHE scheme that can be decrypted at one time.

*1.1. Our Contribution.* Firstly, the public key of the Message-encapsulation Fully Homomorphic Encryption scheme of Hiromasa et al. [2] is as follows:

$$(\mathbf{B} \coloneqq \mathbf{A} \cdot \mathbf{T} + \mathbf{E} \pmod q \mid \mathbf{A}) \in \mathbb{Z}_q^{m\times t} \times \mathbb{Z}_q^{m\times n}. \tag{1}$$

Among them are the secret matrix $\mathbf{T} \longleftarrow \mathbb{Z}_q^{n\times t}$ and the noise matrix $\mathbf{E} \longleftarrow \chi^{m\times t}$. Then, the plaintext message is encapsulated in a matrix, and the public key of the above-mentioned form is used to encrypt the message. However, the obtained ciphertext matrix cannot recover all the plaintext bits at one time, but can only be decrypted bit-by-bit.

Secondly, we notice that the public key matrix of the message-encapsulated fully homomorphic encryption scheme constructed by Li et al. [5] is as follows:

$$(\mathbf{A} \cdot \mathbf{e}_1, \ldots, \mathbf{A} \cdot \mathbf{e}_t \mid \mathbf{A}) \in \mathbb{Z}_q^{m\times t} \times \mathbb{Z}_q^{m\times n}. \tag{2}$$

Among them, there is $\mathbf{e}_1, \ldots, \mathbf{A} \cdot \mathbf{e}_t \longleftarrow \chi^{n\times 1}$. Although Li et al.'s scheme [5] supports bit-by-bit encryption and one-time decryption, the scheme relies on the minimum integer solution hypothesis (see detailed analysis in [7]), and its parameter size depends on $m\,(m \geq n \log q)$ instead of causing the size of the evaluation key and the ciphertext to be too large.

Based on the abovementioned observations, in this paper, we construct a public key matrix first with multiple LWE instances. Different from the typical FHE scheme [3, 4, 8] and follow-up works [9–13], its public key matrix contains only one LWE instance. Then, using the new public key, we construct a message-encapsulation GSW-class FEH scheme (MFHE). We give an overview of the scheme in the following:

(1) Firstly, we use a new public key matrix with multiple LWE instances as follows:

$$\mathbf{A}' = [\mathbf{b}_1, \ldots, \mathbf{b}_t \mid \mathbf{A}] \in \mathbb{Z}_q^{m\times(n+t)}. \tag{3}$$

Among them, $\mathbf{b}_1 = \mathbf{A} \cdot \mathbf{t}_i + e_i \pmod q$ and $i \in [t]$ is an LWE instance. This is significantly different from existing message-encapsulation PKE schemes (for example, [14, 15]) and message-encapsulation FHE schemes (for example, [1, 2]) and is also the fundamental difference between other schemes and the FHE scheme constructed in this paper. Private keys corresponding to the public key $[\mathbf{b}_1, \ldots, \mathbf{b}_t] | \mathbf{A}$ is shaped as follows:

$$\mathbf{sk}_i \coloneqq \left[0, \ldots, 1, \ldots, 0 \,\middle|\, \mathbf{t}_i\right] \in \mathbb{Z}_q^{1\times(n+t)},\ i \in [t]. \tag{4}$$

(2) Next, we use the public key matrix $\mathbf{A}'$ we constructed to encrypt multibit messages. The difference is that we use the message-encapsulation method of Li et al. [5] and Hiromasa et al. [2] to embed multibit messages into the plaintext of a diagonal matrix. That is,

$$\mathbf{M} \coloneqq \operatorname{diag}\left(m_1, \ldots, m_t \,\middle|\, 1, \ldots, 1\right) \in \mathbb{Z}_q^{(n+t)\times(n+t)}, \tag{5}$$

and while constructing a private key matrix with private keys,

$$\mathbf{S} \coloneqq \left[\mathbf{E} \mid \begin{pmatrix} t_1 \\ \vdots \\ t_t \end{pmatrix}\right] \in \mathbb{Z}_q^{(n+t)\times(n+t)}. \tag{6}$$

$\mathbf{E}(n \times n)$ is the identity matrix, and we can get

$$\mathbf{S} \cdot \mathbf{M} \coloneqq \left[\operatorname{diag}(m_1, \ldots, m_t) \,\middle|\, \begin{pmatrix} t_1 \\ \vdots \\ t_t \end{pmatrix}\right]. \tag{7}$$

Finally, using the matrix $\mathbf{W} \coloneqq [\operatorname{diag}(\lfloor (q/2) \rfloor, \ldots, \lfloor (q/2) \rfloor) \mid 0]$ we constructed, calculation of $\mathbf{SM} \cdot GG - 1\,(W)$ can directly recover the message vector $(m_1, \ldots, m_t)$ . See Section 4 for a detailed analysis.

*1.2. Organization and Structure of the Paper.* The rest of this paper is organized as follows. In Section 2, the definitions and symbols used in this paper are introduced. In Section 3, we review the scheme of Gentry-Sahai-Waters et al. In Section 4, we introduce the Message-encapsulation FHE (MFHE) scheme we constructed. Finally, we give a summary of the full paper in Chapter 5.

## 2. Preliminaries

In this section, we give the preparatory knowledge needed, including definitions and lemmas.

*2.1. Symbols.* For $n \in \mathbb{N}$, we use $[n]$ to represent aggregation $\{1, \ldots, n\}$. For a real number $x \in \mathbb{R}$, we use $\lfloor x \rfloor$ to represent the largest integer that is not greater than $x$, $\lfloor x \rceil \coloneqq \lfloor x + (1/2) \rfloor$ to represent the nearest integer to $x$. We represent vectors in bold lowercase letters, for example, $\mathbf{x}$, and the matrix in bold uppercase letters, for example, $\mathbf{A}$. In addition, we use $\mathbf{A}_{i,j}$ to represent elements in $\mathbf{A}_{i,j}$ from row $i$ and column $j$. We use "$\coloneqq$" to indicate the assignment. It is worth noting that we use the definition of computationally indistinguishable and statistics indistinguishable and they are represented by $\approx_c$ and $\approx_s$. In addition to this, we also define $\|\mathbf{v}\|_\infty = \max\{|v_1|, \ldots, |v_n|\}$ and $\|\mathbf{R}\| = \max_i \|\mathbf{r}_i\|$. For convenience, we use $\|v\|$ to represent its $l_2$ norm.

We need to use the following variant of the Left-over Hash Lemma (LHL) [16].

**Lemma 1** *(Matrix-Vector LHL). Let* $\lambda \in \mathbb{Z}, n, q \in \mathbb{N}, m \geq n \log q + 2\lambda, \mathbf{r} \xleftarrow{R} \{0,1\}^m$ *and* $\mathbf{y} \xleftarrow{R} \mathbb{Z}_q^n$. *We select a uniform random matrix* $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{m \times n}$, *and then, the statistical distance of the distribution* $(\mathbf{A}, \mathbf{A}^T\mathbf{r})$ *and* $(\mathbf{A}, \mathbf{y})$ *is as follows:*

$$\Delta\left(\left(\mathbf{A}, \mathbf{A}^T \cdot \mathbf{r}\right), (\mathbf{A}, \mathbf{y})\right) \leq 2^{-\lambda}. \tag{8}$$

*2.2. Learning with Errors (LWEs).* LWEs is the main computational assumption that cryptosystems and our variants rely on.

*Definition 1* (LWE Distribution). For safety parameters, let $n = n(\lambda)$ and $m = m(\lambda)$ be integers, let $\chi = \chi(\lambda)$ be the $\mathbb{Z}$ error distribution with the bound of $B = B(\lambda)$, and let $q = q(\lambda) \geq 2$ be an integer modulo of any polynomial $p = p(\lambda)$ that meets $q \geq 2^p \cdot B$. Then, we select a vector $\mathbf{s} \in \mathbb{Z}_q^{n \times 1}$ and call it a secret, the LWE distribution $\mathcal{A}_{s,\chi}$ in $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is selected uniformly and randomly, and we select $\mathbf{e} \xleftarrow{} \chi^{m \times 1}$ and output $(\mathbf{A}, \mathbf{b} = A \cdot s + e \pmod q)$.

There are two kinds of the LWE hypothesis: the search-LWE and the decision-LWE. The decision-LWE is defined as follows:

*Definition 2* (Decision-LWE$_{n,q,\chi,m}$). Assume an independent selected $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times 1}$, which is selected according to one of the following distributions: (1) for $\mathcal{A}_{s,\chi}$ from a uniform and random $\mathbf{s} \in \mathbb{Z}_q^n$ (i.e., $\{(\mathbf{A}, \mathbf{b})$: $\mathbf{A} \xleftarrow{} \mathbb{Z}_q^{m \times n}, \mathbf{s} \xleftarrow{} \mathbb{Z}_q^{n \times 1}, \mathbf{e} \xleftarrow{} \chi^{m \times 1}, \mathbf{b} = \mathbf{A} \cdot s + e \pmod q\}$) or (2) uniform distribution (i.e., $\{(\mathbf{A}, \mathbf{b})$: $\mathbf{A} \xleftarrow{} \mathbb{Z}_q^{m \times n}$, $\mathbf{b} \xleftarrow{} \mathbb{Z}_q^{m \times 1}\}$). The two distributions mentioned above are computable indistinguishable.

*Note 1.* Regev and others [6, 17–19] introduce the convention between the approximate shortest vector problem (for appropriate parameters) in the LWE hypothesis. We have omitted the lemma of the results of these schemes; see [6, 17–19] for details.

*2.3. Discrete Gauss.* In our structure, we need to analyze the behavior of choosing the wrong element from the Gaussian distribution.

*Definition 3* (B Bounded [3]). A distribution $\chi = \chi(\lambda)$ on an integer if the following exists:

$$\Pr_{x \xleftarrow{\$} \chi}\left[|x| \geq B\right] \leq 2^{-\widetilde{\Omega}(n)}, \tag{9}$$

and then, it is called $B$-bound (represented as $|\chi| \leq B$).

For the analysis of our scheme, the vector selected from the Gaussian distribution needs to have a certain bound on its norm.

**Lemma 2** *(See [20]). 1. For* $\forall k > 0, \Pr[|e| > k \cdot \sigma, e \xleftarrow{} D_\sigma^1] \leq 2 \cdot \exp(-(k^2/2))$; *2. for* $\forall k > 0$, *there is* $\Pr[\|\mathbf{e}\| > k \cdot \sigma \cdot \sqrt{m}\, \mathbf{e} \xleftarrow{} D_\sigma^m] \leq k^m \cdot \exp((m/2) \cdot (1 - k^2))$ *Therefore, in this paper, we set* $|e| \leq B$ *and* $\|e\| \leq 2\sqrt{m}B$.

*In this paper, we assume* $\sigma \geq 2\sqrt{n}$. *So, if* $\mathbf{e} \xleftarrow{} D_\sigma^m$, *then on average,* $\|e\| \approx \sqrt{m} \cdot \sigma$. *It can be known from Lemma 2.2 (2) that there is a high possibility that* $\|\mathbf{e}\| \leq 2\sigma\sqrt{m}$. *Therefore, in this paper, we set* $|e| \leq B$ *and* $\|\mathbf{e}\| \leq 2\sqrt{m}B$.

*2.4. Leveled Fully Homomorphic Encryption.* In public-key cryptography, the cipher keeps a public key and encrypts the message in order that the corresponding private key holder can recover the original plaintext message.

*Definition 4* (See [21]). Let a fixed function $L = L(\lambda)$ be the level of Fully Homomorphic Encryption. For a kind of circuit $\{\mathcal{C}_\lambda\}_{\lambda \in N}$, the L-FHE scheme includes four Probabilistic Polynomial Times (PPTs), and the algorithm is as follows:

$$(\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval}). \tag{10}$$

The key generation algorithm (KeyGen) is a randomization algorithm that inputs security parameters $1^\lambda$ and outputs public keys (pk) and private keys (sk)

The encryption algorithm Enc is a randomization algorithm that inputs a public key (pk) and a message $m \in \{0,1\}^*$ and outputs a ciphertext $c$

The decryption algorithm Dec is a deterministic algorithm that inputs the private key sk and ciphertext and outputs the decrypted message $m \in \{0,1\}^*$

The homomorphic algorithm Eval inputs a public key pk, a circuit $C \in \mathcal{C}_\lambda$, and a sequence of ciphertexts $c_1, \ldots, c_{\ell(\lambda)}$, here let $\ell(\lambda)$ be a polynomial related to $\lambda$ the and outputs the computed ciphertext $c^*$

The correctness requirements are as follows:

For arbitrary $\lambda, m \in \{0,1\}^*$ and $(\text{pk}, \text{sk})$ output by $\text{KeyGen}(1^\lambda)$, we have

$$m = \mathrm{Dec}(\mathrm{sk}, (\mathrm{Enc}(\mathrm{pk}, m))). \tag{11}$$

For arbitrary $\lambda$, arbitrary $m_1, \ldots, m_l \in \{0,1\}^*$, and $C \in \mathscr{C}_\lambda$, we have

$$\mathscr{C}(m_1, \ldots, m_\ell) = \mathrm{Dec}(\mathrm{sk}, (\mathrm{Eval}(\mathrm{pk}, (C, \mathrm{Enc}(pk, m_1), \ldots, \mathrm{Enc}(pk, m_\ell))))). \tag{12}$$

*Definition 5* (CPA Security [21]). One FHE scheme is indistinguishable from the choice of plaintext attack (IND − CPA): the condition that security needs to be satisfied is that for any PPT adversary $\mathscr{A}$, the following probabilities related to are negligible:

$$\begin{aligned} &|\Pr[\mathscr{A}(\mathrm{pk}, \mathrm{Enc}(\mathrm{pk}, m_0)) = 1], \\ &-\Pr[\mathscr{A}(\mathrm{pk}, \mathrm{Enc}(\mathrm{pk}, m_1)) = 1]| = \mathrm{negl}(\lambda). \end{aligned} \tag{13}$$

Among them, $(\mathrm{pk}, \mathrm{sk}) \longleftarrow \mathrm{KeyGen}(1^\lambda)$ and $m_0 \cdot m_1$ is arbitrarily selected from the plaintext space by the adversary.

The security definition of a message-encapsulation GSW (MFHE) is the same as GSW for a single bit. Because in public key settings, the security of single message encryption implies the security of multiple message encryption. See section 11 in [22] for more details.

*Definition 6* (Compactness [21]). For a class of loops $\{\mathbb{C}_k\}_{k\in\mathbb{N}}$, if there is a polynomial $\alpha = \alpha(\lambda)$ such that the length of output ciphertext of Eval is at most $\alpha$, then an $L$ Fully Homomorphic Encryption is compact (if it is nontrivial, then for all $\lambda$, some $C \in \{\mathbb{C}\}_\lambda$, and we have $\alpha(\lambda) \le |C|$).

*2.5. Basic Tools.* Let us review some of the basic tools proposed by Brakerski and Vaikuntanathan [23] and Gentry et al. [4]. We fix $q, m \in \mathbb{N}$. Let $l = \lfloor \log(q) \rfloor + 1$, and therefore, $2^{l-1} \le q < 2^l$ and $N = m \cdot l$.

*Definition 7* (See [24, 25]). The algorithm BitComp enters a vector $\mathbf{v} \in \mathbb{Z}_q^m$ and outputs an $N$-dimensional vector $(v_{1,0}, \ldots, v_{1,l-1}, \ldots, v_{m,0}, \ldots, v_{m,l-1})^T \in \{0,1\}^N$ where $v_{i,j}$ is the $j$ bit in the binary representation of $v_i$ (sorted by minimum impact to maximum impact). In other words,

$$v_i = \sum_{j=0}^{l-1} 2^j v_{i,j}. \tag{14}$$

*Definition 8* (See [24, 25]). Algorithm enters a vector

$$\mathbf{v} = \left(v_{1,0}, \ldots, v_{1,l-1}, \ldots, v_{m,0}, \ldots, v_{m,l-1}\right)^T \epsilon_q^N \tag{15}$$

and output $(\sum_{j=0}^{l-1} 2^j, \ldots, v_{1,j}, \ldots, \sum_{j=0}^{l-1} 2^{j v_{m,j}})^T \epsilon_q^m$.

Note that the input vector $\mathbf{v}$ does not need to be binary and any of the input vector algorithms in $\mathbb{Z}^N$ are already defined.

*Definition 9* (See [24, 25]). The algorithm Flatten enters a vector $\mathbf{v} \in \mathbb{Z}_q^N$ and outputs an $N$-dimension binary vector (i.e., an element from $0, 1^N$) defined as

$$\mathrm{Flatten}(\mathbf{v}) = \mathrm{BitDecomp}\left(\mathrm{BitDecomp}^{-1}(\mathbf{v})\right). \tag{16}$$

*Definition 10* (See [24, 25]). The algorithm PoweOftwo enters an $m$-dimension vector $\mathbf{v} \in \mathbb{Z}_q^N$ and outputs an $N$-dimension vector in $\mathbb{Z}_q^N$. The output is as follows:

$$\left(v_1, 2v_1, \ldots, 2^{l-1}v_1, \ldots, v_m, 2v_m, \ldots, 2^{l-1}v_m\right)^T. \tag{17}$$

**Lemma 3** *(See [26]). For any $N \ge m\lfloor \log q \rfloor$, there is a fixed effective computable matrix $\mathbf{G} \in \mathbb{Z}_q^{m\times N}$ and a valid computable deterministic "short-image" function $\mathbf{G}^{-1}(\cdot)$ that meets the following conditions. For arbitrary $m'$, we enter a matrix $\mathbf{M} \in \mathbb{Z}_q^{m\times m'}$ and the inverse function $\mathbf{G}^{-1}(\mathbf{M})$ outputs a matrix $\mathbf{G}^{-1}(\mathbf{M}) \in \{0,1\}^{N\times m'}$ so that $\mathbf{G}\mathbf{G}^{-1}(\mathbf{M}) = \mathbf{M}$.*

*Note 2.* In fact, we can also express the abovementioned definitions and results as follows using the language of $\mathbf{G}$ and $\mathbf{G}^{-1}$. Micciancio and Peikert's [26] matrix $\mathbf{G}$ can be expressed as $\mathbf{G} = \mathbf{I}_m \otimes \in \mathbb{Z}_q^{m\times N}$, where $\mathbf{g} = (1, 2, 4, \ldots, 2^{l-1})^T$. For $\mathbf{v} \in \mathbb{Z}_q^m$, there is $(\mathbf{v}) = \mathbf{v}^T\mathbf{G}$. For $\mathbf{v} \in \mathbb{Z}_q^N$, there is $\mathrm{BitDecomp}^{-1}(\mathbf{v}) = \mathbf{G}\mathbf{v}$. For $\mathbf{a} \in \mathbb{Z}_q^m$, the algorithm BitDecomp $(\mathbf{a})$ is renamed as $\mathbf{G}^{-1}(\mathbf{a})$. For $\mathbf{v} \in \mathbb{Z}_q^m$, there is $\mathrm{Power\,Of\,two}(\mathbf{v}) = \mathbf{v}^T\mathbf{G}$. For $\mathbf{v} \in \mathbb{Z}_q^N$, there is $\mathrm{BitDecomp}^{-1}(\mathbf{v}) = \mathbf{G}\mathbf{v}$. For $\mathbf{a} \in \mathbb{Z}_q^m$, the algorithm BitDecomp $(a)$ is renamed as $\mathbf{G}^{-1}(\mathbf{a})$.

## 3. Gentry–Sahai–Waters (GSW) Scheme

Before our work, we first review the GSW scheme and, then, summarize the safety of the scheme of Gentry et al. [4].

We review the algorithms which make up the GSW scheme [4]. These algorithms were originally defined based on functions BitDecomp, $\mathrm{BitDecomp}^{-1}$, and Flatten, but the ideas from [19, 27] borrowed into this paper are defined using tool matrix $\mathbf{G}$. Let $\lambda$ be the security parameter and $L$ be the number of levels of homomorphic encryption.

GSW.Setup $(1^\lambda, 1^L)$:

(1) Select a module $q$ of bit $\mathscr{K}$ = mathcal$K(\lambda, L)$, error distribution $\chi = \chi(\lambda, L)$ on the parameter $n = n(\lambda, L) \in \mathbb{N}$ and $\mathbb{Z}$, so that the $(q, n, \chi)$ − LWE problem is at least $2^\lambda$ secure for known attacks. Choose a parameter $m = m(\lambda, L) = O(n\log(q))$.
(2) Output: params $= (n, q, \chi, m)$. We express $l = \lfloor \log(q) \rfloor + 1$ and $N = (n+1)\cdot l$.

GSW.KeyGen (params):

(1) Select $\mathbf{t} = (t_1, \ldots, t_n)^T \longleftarrow \mathbb{Z}_q^n$ and calculate

$$\mathbf{s} \longleftarrow \left(1, -\mathbf{t}^T\right)^T = \left(1, -t_1, \ldots, -t_n\right)^T \in \mathbb{Z}_q^{(n+1)\times 1}. \quad (18)$$

(2) Generate a matrix $\mathbf{B} \longleftarrow \mathbb{Z}_q^{mm\times n}$ and a vector $\mathbf{e} \longleftarrow \chi^m$.

(3) Calculate $\mathbf{b} = \mathbf{Bt} + \mathbf{e} \in \mathbb{Z}_q^m$ and construct matrix $\mathbf{A} = (b \mid B) \in \mathbb{Z}_q^{m\times(n+1)}$. Obviously, we observed.

(4) Return to $\mathrm{sk} \longleftarrow \mathbf{s}$ and $\mathrm{pk} \longleftarrow \mathbf{A}$.

GSW.Enc(params, pk, $\mu$): in order to encrypt a single-bit message $\mu \in \{0, 1\}$,

(1) Let $\mathbf{G}$ be the abovementioned matrix $(n+1) \times N$

(2) Select a matrix $\mathbf{R} \longleftarrow \{0,1\}^{m\times N}$ evenly

(3) Calculate

$$\mathbf{C} = \mu G + A^T \mathbf{R} \,(\mathrm{mod}\, q) \in \mathbb{Z}_q^{(n+1)\times N} \quad (19)$$

In the original GSW scheme,

Flatten($\mu\mathbf{I}$ + BitDecomp($\mathbf{RA}$)) $\in \{0,1\}^{N\times N}$, where $\mathbf{I}$ is an identity matrix.

GSW.Dec(params, sk, $C$):

(1) We have $\mathrm{sk} = \mathbf{s} \in \mathbb{Z}_q^{n+1}$.

(2) Let $I$ meet $(q/4) < 2^{I-1} \le (q/2)$. Let $\mathbf{C}_I$ be column $I$ of $\mathbf{C}$.

(3) Calculate $x \longleftarrow \langle \mathbf{C}_I, \mathbf{s} \rangle \,(\mathrm{mod}\, q)$ within the scope of $(-(q/2), (q/2)]$; note $\langle \mathbf{C}_I, \mathbf{s} \rangle = \mathbf{C}_I^T \mathbf{s}$ and

$$\mathbf{C}^T \mathbf{s} = \mu \mathbf{G}^T \mathbf{s} + \mathbf{R}^T \mathbf{A}\mathbf{s} = \mu\,(1, 2, 4, \ldots)^T + \mathbf{R}^T \mathbf{e}. \quad (20)$$

From that mentioned above, it can be seen that column $I$ of the ciphertext matrix $\mathbf{C}$ selected in the calculation corresponds to coordinate $I$ of the vector $\langle \mathbf{C}_I, \mathbf{s} \rangle$, i.e. $\mu 2^{I-1} + \mathbf{R}_I^T \mathbf{e}$.

(4) Output $\mu' = \left|\left\lfloor (x/2)^{I-1} \right\rfloor\right|$.

So, if it is $|x| < 2^{I-2} \le (q/4)$, then it returns to 0, and if it is $|x| > 2^{I-2}$, then it returns to 1.

GSW.Eval(params, $C_1, \ldots, C_l$):

GSW.Mult($\mathbf{C}_1, \mathbf{C}_2$): calculate and output

$$\begin{aligned} \mathbf{C}_1 \mathbf{G}^{-1}(\mathbf{C}_2) &= \left(\mu_1 \mathbf{G} + \mathbf{A}^T \mathbf{R}_1\right) \mathbf{G}^{-1}(\mathbf{C}_2) \\ &= \mu_1 \mathbf{C}_2 + \mathbf{A}^T \mathbf{R}_1 \mathbf{G}^{-1}(\mathbf{C}_2) \\ &= \mathbf{A}^T \left(\mathbf{R}_1 \mathbf{G}^{-1}(\mathbf{C}_2) + \mu_1 \mathbf{R}_2 + \mu_1 \mu_2 \mathbf{G} (\mathrm{mod} q)\right). \end{aligned} \quad (21)$$

GSW.Add($\mathbf{C}_1, \mathbf{C}_2$) $\in \mathbb{Z}_q^{(n+1)\times N}$: output

$$\mathbf{C}_1 + \mathbf{C}_2 = (\mu_1 + \mu_2)\mathbf{G} + \mathbf{A}^T (\mathbf{R}_1 + \mathbf{R}_2). \quad (22)$$

Note that $\mathbf{C}_1 \mathbf{G}^{-1}(\mathbf{C}_2) \in \mathbb{Z}_q^{(n+1)\times N}$. In addition, use $\mathbf{G} - \mathbf{C}_1 \mathbf{G}^{-1}(\mathbf{C}_2)$ to calculate homomorphic NAND gates.

*Note 3.* Note that, in [19], the decryption algorithm is to select a suitable vector $\mathbf{w}$ and calculate $\mathbf{sCG}^{-1}(\mathbf{w}^T)$. It is much less efficient than the original one (all about calculation time and error item size). So, we used the GSW decryption algorithm in our scheme.

When $q$ is a power of 2, there is also a variant of the message in $\mathbb{Z}_q$. See more details in [4].

*3.1. Security.* A brief proof of the following theorem is given in [4].

**Theorem 1.** *Let $(n, q, \chi)$ be public parameter so that the $LWE_{(n,q,\chi)}$ hypothesis is true, and let $m = O(n \log(q))$. Then, we can say that the GSW scheme is IND − CPA safe.*

*The most important step of the proof is to prove that $(\mathbf{A}, \mathbf{RA})$ and the uniform distribution is computational indistinguishable.*

*Note 4.* The correctness of the GSW scheme is obtained by analyzing the scale of the noise during encryption, decryption, and homomorphism. Always ensure that the maximum noise level in the abovementioned process is still less than 1/4, which can be decrypted correctly. This work is not the focus of this paper, so it will not be repeated. See more details [4].

## 4. Message-Encapsulation FHE

*4.1. Message-Encapsulation FHE (MFHE Scheme).* Now, we introduce our MFHE scheme as follows: a message-encapsulation public-key encryption scheme based on the difficulty of the LWE hypothesis. We give the security parameter $\lambda$, set $t$ to be the private keys number, and then, can encrypt the $t$-bit messages at one time.

Let $q = q(\lambda)$ be an integer, and let $\chi = \chi(\lambda)$ be a distribution set on $\mathbb{Z}$. The definition of the variant of the GSW scheme is similar to the cryptosystem proposed in [19, 27, 28]. More specifically,

params $\longleftarrow$ MFHE.Setup($1^\lambda, 1^L$):

(1) In particular, we first select the modulo $q = q(\lambda)$, and the dimension of lattice $n = n(\lambda, L)$. We appropriately select the error distribution for $\chi = \chi(\lambda, L)$ for $2^\lambda$ security against known LWE attacks, Finally, we select the parameter $m = m(\lambda, L) = O(n \log q)$ and a parameter $t = O(\log(n))$.

(2) Let $l = \lfloor \log q \rfloor + 1$ and $N = (n + t) \cdots l$, and then, output params $= (n, q, \chi, m, t)$.

(pk, sk) $\longleftarrow$ MFHE.KeyGen(params):

(1) For $i \in [t]$, select $\mathbf{t}_i^T = (t_{i,1}, \ldots, t_{i,n})$ from $\mathbb{Z}_q^{1\times n}$ and output

$$\begin{aligned} \mathbf{sk}_i \coloneqq \mathbf{s}_i &= \left(\mathbf{I}_i \mid -\mathbf{t}_i^T\right)^T \\ &= \left(0, \ldots, 1, \ldots, 0 \mid -t_{i,1}, \ldots, -t_{i,n}\right)^T \in \mathbb{Z}_q^{(n+t)\times 1}, \end{aligned} \quad (23)$$

the $i$ position of which is 1.

(2) Select a matrix $\mathbf{B} \longleftarrow \mathbb{Z}_q^{m\times n}$ and $t$ vectors $\mathbf{e}_i \longleftarrow \chi^{m\times 1}$, $i \in [t]$ evenly, and then, calculate $\mathbf{b}_i = \mathbf{B} \cdot t_i + \mathbf{e}_i \pmod q$ and output

$$\text{pk} = \mathbf{P} = \left[\mathbf{b}_1 | \cdots | \mathbf{b}_t | \mathbf{B}\right] \in \mathbb{Z}_q^{m\times(n+t)}, \tag{24}$$

where the size of pk is $O(nm \cdot \log^2 q)$. In addition, we observed that $\mathbf{P} \cdot s_i = \mathbf{e}_i \pmod q$.

(3) Output $\text{pk} \longleftarrow \mathbf{P}$ and $\text{sk} \longleftarrow \mathbf{S} \coloneqq \{\mathbf{s}_1, \ldots, \mathbf{s}_t\}$. It is worth noting that $\mathbf{P} \cdot S = [\mathbf{e}_1, \ldots, \mathbf{e}_t] \pmod q$.

$\mathbf{C} \longleftarrow$ MFHE.Enc (params, pk, $\mathbf{M}$):

(1) To encrypt $t$-bit $\mu_i \in 0, 1$, $\mu_i \in 0, 1$, embed the $t$ bits into a $(t \times t)$-dimension matrix first, $\mathbf{U} = \text{diag}(\mu_{1,1}, \ldots, \mu_{t,t}) \in 0, 1^{t\times t}$, where $\mu_{i,j} = 0$, $i \neq j$, and $j \in [t]$. Later, for simplicity, $\mu_{i,j}$ will be abbreviated as $\mu_i$, and the message matrix is constructed using a plaintext matrix $\mathbf{U}$.

$$\mathbf{M} = \begin{pmatrix} \mathbf{U}_{t\times t} & 0_{t\times n} \\ 0_{n\times t} & \mathbf{E}_{n\times n} \end{pmatrix} \in \{0, 1\}^{(n+t)\times(n+t)}, \tag{25}$$

where $\mathbf{U}$ is a random diagonal matrix, and note that $\mathbf{E}$ is a $(n \times n)$-dimensional matrix.

(2) Then, select a uniform matrix $\mathbf{R} \longleftarrow 0, 1^{m\times N}$. Calculate and output cipher text:

$$\mathbf{C} = M \cdot G + \mathbf{P}^T \cdot \mathbf{R} \pmod q \in \mathbb{Z}_q^{(n+t)\times N}. \tag{26}$$

Now, we propose a decryption algorithm for the MFHE scheme which allows us to recover all the message bits at the one time.

$\mathbf{U} \longleftarrow$ MFHE.Dec (params, pk, $\mathbf{C}$):

(1) First, assume that the user has a private key matrix $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_t) \in \mathbb{Z}_q^{(n+t)\times t}$ as follows:

$$\mathbf{S} \coloneqq (\mathbf{s}_1, \ldots, \mathbf{s}_t) = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \\ -t_{1,1} & \cdots & -t_{t,1} \\ \vdots & \ddots & \vdots \\ -t_{1,n} & \cdots & -t_{t,n} \end{pmatrix}. \tag{27}$$

What needs to be noted here is

$$\mathbf{P} \cdot S = \left[\mathbf{b}_1 - \mathbf{B}\mathbf{t}_1, \ldots, \mathbf{b}_t - \mathbf{B}\mathbf{b}_t\right] = \left[\mathbf{e}_1, \ldots, \mathbf{e}_t\right]\left(\text{mod} \in \mathbb{Z}_q^{m\times t}\right). \tag{28}$$

Therefore, it is easy for us to get the bound of $\mathbf{P} \cdot S$ which is less than or equal to $t|\mathbf{e}|$, i.e. $\|\mathbf{P} \cdot S\| \leq t|\mathbf{e}|$.

(2) Define the matrix $\mathbf{W}\mathbb{Z}_q^{t\times((+t)}$ as follows:

$$\mathbf{W}^T \coloneqq \begin{pmatrix} \lceil \frac{q}{2} \rceil & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lceil \frac{q}{2} \rceil \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix}. \tag{29}$$

(3) Calculate and output

$$\mathbf{V}_{i,j} = \langle \mathbf{S}, \mathbf{C} \rangle \cdot \mathbf{G}^{-1}\left(\mathbf{W}^T\right) \pmod q \in \mathbb{Z}_q^{t\times t}. \tag{30}$$

Among them, we have $\langle \mathbf{S}, \mathbf{C} \rangle \in \mathbb{Z}_q^{t\times t}$, i.e.,

$$\langle \mathbf{S}, \mathbf{C} \rangle = \mathbf{S}^T\mathbf{P}^T\mathbf{R} + \mathbf{S}^T\mathbf{M}\mathbf{G} = \left[\mathbf{e}_1, \ldots, \mathbf{e}_t\right]^T\mathbf{R} + \mathbf{S}^T\mathbf{M}\mathbf{G} \pmod q. \tag{31}$$

(4) Finally, use the results mentioned above to output the complete message $\mathbf{U} = \| \lfloor (\mathbf{V}_{i,j}/(q/2)) \rfloor \| \in \{0, 1\}^{t\times t}$.

MFHE.Eval (params, $\mathbf{C}_1, \ldots, \mathbf{C}_l$):there are two algorithms, which are, homomorphic addition and homomorphic multiplication. For any two plaintext matrices $\mathbf{U}_1, \mathbf{U}_2 \in \{0, 1\}^{t\times t}$, we get the ciphertext separately.

$$\begin{aligned} \mathbf{C}_1 &= \mathbf{M}_1 \cdot \mathbf{G} + \mathbf{P}^T \cdot \mathbf{R}_1, \\ \mathbf{C}_2 &= \mathbf{M}_2 \cdot \mathbf{G} + \mathbf{P}^T \cdot \mathbf{R}_2. \end{aligned} \tag{32}$$

Therefore, the homomorphic addition and multiplication are as follows:

MFHE.Mult $(\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{(n+t)\times N}$: output

$$\begin{aligned} \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) = \left(\mathbf{M}_1\mathbf{G} + \mathbf{P}^T\mathbf{R}_1\right) \cdot \mathbf{G}^{-1}(\mathbf{C}_2) = \mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) \\ + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2 + \mathbf{M}_1\mathbf{M}_2\mathbf{G} \pmod q. \end{aligned} \tag{33}$$

MFHE.Add $(\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{(n+t)\times N}$: output $\mathbf{C}_1 + \mathbf{C}_2 = (\mathbf{M}_1 + \mathbf{M}_2)\mathbf{G} + \mathbf{P}^T(\mathbf{R}_1 + \mathbf{R}_2)$.

Here, we can calculate a homomorphic NAND gate from the output.

*Note 5.* Generally, we can choose different private keys $\text{sk}_i$ to decrypt column $j$ of the ciphertext $\mathbf{C}_j$ bit-by-bit and get the $i$

bit message of $C_j$, that is, we can get the bit in row $i$ and column $j$ under the $i$ private key. However, it is actually possible to recover the entire message using the private key matrix $\mathbf{S}$ based on the abovementioned decryption algorithm. We calculate $\mathbf{V}_{i,j} = \mathbf{S}^T\mathbf{C} \cdot G^{-1}(\mathbf{W}^T)$ as follows:

$$\mathbf{V}_{i,j} = \lceil \frac{q}{2} \rceil \cdot \mathbf{U} + \begin{pmatrix} \mathbf{e}_1^T\mathbf{R} \\ \vdots \\ \mathbf{e}_t^T\mathbf{R} \end{pmatrix} \cdot \mathbf{G}^{-1}\left(\mathbf{W}^T\right) \in \mathbb{Z}_q^{t\times t}. \tag{34}$$

The magnitude of the noise can be simply calculated and verified to grow linearly compared to single-bit decryption algorithm.

$\mu_{i,j} \longleftarrow$ MFHE.bitDec(params, $\mathrm{sk}_i$, $\mathbf{C}$, $\mathbf{w}_j$):

(1) Suppose we want to decrypt the bit $\mu_{i,j}$ of row $i$ and column $j$, so let $\mathrm{sk}_i = \mathbf{s}_i :=$ , then define a vector so that the position is, and the other positions are 0, $j \in [t]$.

$$\mathbf{w}_j^T = \left[ \underbrace{0, \ldots, \lceil \frac{q}{2} \rceil_j, \ldots, 0}_{t} \middle| \underbrace{0, \ldots, 0}_{n} \right]. \tag{35}$$

(2) For $i$, $j$ to $t$, calculate

$$v_{i,j} = \mathbf{s}_i^T\mathbf{C} \cdot \mathbf{G}^{-1}\left(\mathbf{w}_j^T\right) (\mathrm{mod}\, q) \in \mathbb{Z}_q. \tag{36}$$

The inner product of $\langle \mathbf{s}_i, \mathbf{C} \rangle$ equals to

$$\mathbf{s}_i^T\mathbf{P}^T\mathbf{R} + \mathbf{s}_i^T\mathbf{M}\mathbf{G} = \mathbf{e}_i^T\mathbf{R} + \mathbf{s}_i^T\mathbf{M}\mathbf{G} (\mathrm{mod}\, q)\mathbb{Z}_q^{1\times N}. \tag{37}$$

(3) Output a message $\mu_{i,j} = \| \lfloor (\mathbf{V}_{i,j}/(q/2)) \rfloor \| \in \{0, 1\}$, in which $\lfloor \cdot \rfloor$ represents the operation that rounds to the nearest integer. Therefore the value belongs to $\{0, 1\}$. 4. Finally, by repeating it $t^2$ times, the entire message can be recovered. The bitDec algorithm here is similar to the algorithm in [2], which is achieved by recovering each element separately.

*Note 6.* It should be noted here that due to the structural characteristics of the public key in our scheme, accurate decryption is achieved by dynamically adjusting the position of $\lceil (q/2) \rceil$ in the vector $\mathbf{w}$. That is, dot-multiply $\mathbf{s}_i^T\mathbf{C}$ and $\mathbf{G}^{-1}(\mathbf{w}_j)$ to obtain the bits of the row and column of the plaintext matrix.

We can get all the bits of the message by using the bitDec decryption algorithm and appropriate private key.

*Note 7.* It can be seen that our message-encapsulation GSW scheme is to implement $t \times t$-bit homomorphic addition. However, since the $(i, j)$ element of $\mathbf{U}_1 \times \mathbf{U}_2$ is not a product of $\mu_{1_{i,j}} \times \mu_{2_{i,j}}$, only $t$-bit homomorphic multiplication is supported.

*4.2. Correctness Analysis.* Next, we analyze the correctness of the MFHE scheme.

*Definition 11.* We call the message matrix $\mathbf{U} \in \mathbb{Z}_q^{t\times t}$ which is obtained by decrypting the ciphertext under $t$ different private keys $\mathbf{s}_i, i \in [t]$ (see (2)). The noise of a single-bit message is as follows:

$$\mathrm{noise}_{(\mathbf{s}_i,\mathbf{M})} = \mathbf{s}_i^T\mathbf{C} - \mathbf{s}_i^T\mathbf{M}\mathbf{G} = \mathbf{s}_i^T\mathbf{P}^T\mathbf{R} = \mathbf{e}_i^T\mathbf{R}. \tag{38}$$

For flexible single-bit decryption algorithm bitDec, we represent the noise vector as noise $\in \mathbb{Z}_q^{1\times N}$. For simplicity, we abbreviate $\mathrm{noise}_{(\mathbf{s}_i,\mathbf{M})}(\mathbf{C})$ to $\mathrm{noise}_{\mathbf{s}_i}$ when $\mathbf{M}$ and $\mathbf{C}$ do not affect the contextual understanding.

Note that, in our setup, due to the structure of the new public key, $\mathrm{noise}_{\mathbf{s}_i}$ is the noise of row $i$ of the plaintext matrix $\mathbf{U}$, not the single-bit noise.

**Lemma 4.** *Obviously, using Definition 4.1, for convenience, for a decryption algorithm Dec, if the noise meets*

$$\mathrm{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{C}) = \mathbf{S}^T \cdot \mathbf{P}^T \cdot \mathbf{R} = \begin{pmatrix} \mathrm{noise}_{\mathbf{s}_1} \\ \vdots \\ \mathrm{noise}_{\mathbf{s}_t} \end{pmatrix} (\mathrm{mod} q), \tag{39}$$

*where* $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_t]$ *is a one-time private key matrix, we can represent the entire noise matrix as*

$$\mathrm{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{C}) = \left(\mathrm{noise}_{s_1}, \ldots, \mathrm{noise}_{s_t}\right)^T \in \mathbb{Z}_q^{t\times N}. \tag{40}$$

*For convenience, we will abbreviate* $Noise_{(\mathbf{S},\mathbf{M})}(\mathbf{C})$ *as* $Noise_{\mathbf{S}}$ *when* $\mathbf{M}$ *and* $\mathbf{C}$ *do not affect the contextual understanding.*

*In order to analyze the correctness, for convenience, we first define the following noise ciphertext concept.*

*Definition 12* ($E$-Noise Ciphertext). A ciphertext matrix $\mathbf{C} \in \mathbb{Z}_q^{(m+1)\times N}$ with $E$ noise, which makes in a private key $\mathbf{s}_i \in \mathbb{Z}_q^{(n+t)\times 1}$, for a corresponding message $\mathbf{M}$, $\langle \mathbf{s}_i, \mathbf{C} \rangle = \mathbf{s}_i^T \cdot \mathbf{M} \cdot G + \mathbf{e}_i^T \cdot \mathbf{R}$. Then, let the norm of $\mathrm{noise}_{\mathbf{s}_i}$ be

$$\left\|\mathrm{noise}_{\mathbf{s}_i}\right\| \le \left\|\mathbf{e}_i^T\mathbf{R}\right\| \le \left\|\mathbf{e}_i^T\right\|_2 \cdot \|\mathbf{R}\|_\infty \le \sqrt{N} \cdot 2\sqrt{m}B \le E. \tag{41}$$

**Lemma 5.** *For a one-time private key matrix* $\mathbf{S} \in \mathbb{Z}_q^{(n+t)\times t}$, *we can get* $Noise_{\mathbf{S}} = [\mathbf{e}_1, \ldots, \mathbf{e}_t]^T \cdot \mathbf{R}$ *when we run the Dec algorithm. So, in this case, we get*

$$\left\|\mathrm{Noise}_{\mathbf{S}}\right\| \le t \cdot \left\|\mathrm{noise}_{\mathbf{s}_i}\right\| \le t \cdot E. \tag{42}$$

**Lemma 6.** *For a plaintext matrix* $\mathbf{U}$ *(a combination of* $\mathbf{M}$*) and a private key* $\mathbf{s}_i, i \in [t]$, *the noise vector of the ciphertext* $\mathbf{C}$ *meets*

$$t\left\|\mathrm{noise}_{\mathbf{s}_i}\right\| = \left\|\mathrm{Noise}_{\mathbf{S}}\right\|. \tag{43}$$

*In the following, we analyze the correctness of the decryption.*

**Lemma 7.** *Let* $\mathbf{C}$ *be an* $E$ *noise encryption of* $\mathbf{M}$. *If we can recover* $\mu_{i,j}$ *(an element of* $\mathbf{U}$*) from the ciphertext* $\mathbf{C}$ *under the private key* $\mathbf{s}_i$, *then there is*

$$\mu_{i,j} \coloneqq \langle \mathbf{s}_i, \mathbf{C} \rangle \cdot \mathbf{G}^{-1}\left(\mathbf{w}_j^T\right) = \left(\text{noise}_{\mathbf{s}_i} + \mathbf{s}_i^T \mathbf{M}\mathbf{G}\right) \cdot \mathbf{G}^{-1}\left(\mathbf{w}_j^T\right), \tag{44}$$

*so that*

$$\left\|\text{noise}_{\mathbf{s}_i} \cdot \mathbf{G}^{-1}\left(\mathbf{w}_j^T\right)\right\|_\infty \le \left\|\text{noise}_{\mathbf{s}_i}\right\| \cdot \left\|\mathbf{G}^{-1}\left(\mathbf{w}_j^T\right)\right\| \le N \cdot E < \frac{q}{8}. \tag{45}$$

*Proof.* Obviously, by using Lemma 4.2 we can simply prove Lemma 4.7, and we will not go into details here. □

**Lemma 8.** *Let* $\mathbf{C}$ *be an* $E$ *noise encryption in* $\mathbf{M}$. *If we can recover all* $\mathbf{U}$ *from the ciphertext* $\mathbf{C}$, *then there is a private key matrix* $\mathbf{S}$ *such that*

$$\mathbf{V} = \langle \mathbf{S}, \mathbf{C} \rangle \cdot \mathbf{G}^{-1}\left(\mathbf{W}^T\right) = \left(\text{Noise}_{\mathbf{S}} + \mathbf{S}^T \mathbf{M}\mathbf{G}\right) \cdot \mathbf{G}^{-1}\left(\mathbf{W}^T\right), \tag{46}$$

*where* $\|Noise_{SS} \cdot \mathbf{G}^{-1}(\mathbf{W}^T)\|_\infty \le N \cdot tE < (q/8)$.

*Proof.* This proof can be obtained directly from Lemma 4.2 and Lemma 4.7. Now, we know that as long as $\|\text{Noise}_{\mathbf{S}} \cdot \mathbf{G}^{-1}(\mathbf{W}^T)\|_\infty \le (q/8)$, the decryption runs correctly, i.e., $E < (q/4tN)$. Therefore, we call the value $E = (q/4tN)$ as the bound of noise.

The analysis of the homomorphic operation is given in the following. Before introducing the boundary of noise, the following notes are given. □

*Note 8.* For the convenience of reading, let $\Upsilon_{\mathbf{C}_1} \coloneqq \text{Noise}_{(\mathbf{S},\mathbf{M}_1)}(\mathbf{C}_1)$ and $\Upsilon_{\mathbf{C}_2} \coloneqq \text{Noise}_{(SS,\mathbf{M}_2)}(\mathbf{C}_2)$.

**Lemma 9** *(See [8]). The boundary of the noise of homomorphic addition, homomorphic multiplication, and homomorphic negative is as follows:*

*Addition: for* $\mathbf{M}_1, \mathbf{M}_2 \in \{0,1\}^{(n+t)\times(n+t)}$, *the following condition is met:*

$$\left\|\text{Noise}_{(\mathbf{S},\mathbf{M}_1+\mathbf{M}_2)}(\mathbf{C}_1 + \mathbf{C}_2)\right\| \le \left\|\Upsilon_{\mathbf{C}_1}\right\| + \left\|\Upsilon_{\mathbf{C}_2}\right\|. \tag{47}$$

*Multiplication: for* $\mathbf{M}_1, \mathbf{M}_2$, *the following condition is met:*

$$\left\|\text{Noise}_{(\mathbf{S},(\mathbf{M}_1\cdot\mathbf{M}_2))}\left(\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)\right)\right\| \le \|\mathbf{U}_1\|_2 \cdot \left\|\Upsilon_{\mathbf{C}_2}\right\|_\infty + \left\|\mathbf{G}^{-1}(\mathbf{C}_2)\right\|_\infty \cdot \left\|\Upsilon_{\mathbf{C}_1}\right\|_\infty. \tag{48}$$

*NAND: for* $\mathbf{M}$, *the following condition is met:*

$$\left\|\text{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{G} - \mathbf{C})\right\| = \left\|\text{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{C})\right\|. \tag{49}$$

*Proof.* Let $\mathbf{S} \in \mathbb{Z}^{(n+t)\times t}$ be a private key matrix. Let $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{Z}_q^{(m+1)\times N}$ be the ciphertext of the encrypted message $\mathbf{M}_1, \mathbf{M}_2 \in \{0,1\}^{(n+t)\times(n+t)}$ separately. Then,

Homomorphic addition, that is, add ciphertext and ciphertext $\mathbf{C}^{\text{Add}} = \mathbf{C}_1 + \mathbf{C}_2 \pmod q$, so that

$$\langle \mathbf{S}, \mathbf{C}^{\text{Add}} \rangle = \text{Noise}_{(SS,\mathbf{M}_1+\mathbf{M}_2)} + \mathbf{S}^T \cdot \mathbf{M}^{\text{Add}} \cdot \mathbf{G}. \tag{50}$$

Where $\mathbf{M}^{\text{Add}} = \mathbf{M}_1 + \mathbf{M}_2$ and the noise is

$$\text{Noise}_{(\mathbf{S},\mathbf{M}_1+\mathbf{M}_2)} = \text{Noise}_{(\mathbf{S},\mathbf{M}_1)} + \text{Noise}_{(\mathbf{S},\mathbf{M}_2)}. \tag{51}$$

Obviously, the noise is $t \cdot (E_1 + E_2)$.

Homomorphic multiplication: that is, multiply the ciphertext and ciphertext $\mathbf{C}^{\text{Mult}} = \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) \in \mathbb{Z}_q^{(n+t)\times N}$, so that

$$\mathbf{C}^{\text{Mult}} = \mathbf{M}_1\mathbf{M}_2\mathbf{G} + \left(\mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2\right). \tag{52}$$

Then, we have $\langle \mathbf{S}, \mathbf{C}^{\text{Mult}} \rangle$ which equals to

$$\mathbf{S}^T\left(\mathbf{M}_1\mathbf{M}_2\mathbf{G} + \left(\mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2\right)\right). \tag{53}$$

For convenience, we first set the noise to

$$\text{Noise}_{(\mathbf{S},\mathbf{M}_1\mathbf{M}_2)} = \mathbf{S}^T\left(\mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2\right). \tag{54}$$

Obviously, according to Lemma 4.2, there is

$$\left\|\Upsilon_{\mathbf{C}_1}\right\| = \left\|\mathbf{S}^T\mathbf{P}^T\mathbf{R}_1\right\| \le \left\|[\mathbf{e}_1, \ldots, \mathbf{e}_t]^T \cdot \mathbf{R}_1\right\| \le tE_1, \tag{55}$$

and $\mathbf{C}_2$ is a $(n+t) \times N$ binary matrix ($\mathbf{G}^{-1} \in \mathbb{Z}_q^{N\times(n+t)}$). Therefore, in this case,

$$\left\|\mathbf{S}^T\mathbf{P}^T\mathbf{R}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2)\right\| \le tE_2 \cdot \left\|\mathbf{G}^{-1}(\mathbf{C}_2)\right\| \le N \cdot tE_2 \tag{56}$$

exists. Also, pay attention to that

$$\mathbf{S}^T \cdot \left(\mathbf{M}_1\mathbf{P}^T\right) = \begin{pmatrix} \left(u_i\mathbf{b}_1^T - \mathbf{t}_i^T\mathbf{B}^T\right) \\ \vdots \\ \left(u_i\mathbf{b}_t^T - \mathbf{t}_i^T\mathbf{B}^T\right) \end{pmatrix}. \tag{57}$$

The boundary of $(u_i \cdot \mathbf{b}_i^T - \mathbf{t}_i^T \cdot \mathbf{B}^T)$ is $|\mathbf{e}_i^T|$. Therefore,

$$\left\|\mathbf{S}^T \cdot \left(\mathbf{M}_1\mathbf{P}^T\right)\right\| \le \left\|\left[\mathbf{e}_1^T, \ldots, \mathbf{e}_t^T\right]^T\right\| \le \max_i\left\|\mathbf{e}_i^T\right\|. \tag{58}$$

In this case, we can easily get the boundary $\|\Upsilon_{\mathbf{C}_2}\| \coloneqq \|\mathbf{S}^T \cdot (\mathbf{M}_1\mathbf{P}^T\mathbf{R}_2)\| \le \|\mathbf{e}_i^T\mathbf{R}\| \le E_2$. In other words, $\|\mathbf{U}_1\|_2 \cdot \|\Upsilon_{\mathbf{C}_2}\|_\infty \le \sqrt{t} \cdot E_2$. Therefore, we have $\|\text{Noise}_{(SS,(\mathbf{M}_1\cdot\mathbf{M}_2))}(\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2))\| \le NtE_2 + \sqrt{t}E_2$, and the ciphertext $\mathbf{C}^{\text{Mult}}$ is $((Nt + \sqrt{r}) \cdot E)$ noisy.

Table 1: Comparison of the related works.

| Underlying Assumption | HAO LWE | Ours-MFHE LWE | LMDO[ dual-LWE |
|---|---|---|---|
| $\|\text{msg}\|$ | t | t | t |
| $\|\mathbf{pk}\|$ | $\mathcal{O}(mn\log q)$ | $\mathcal{O}(nm\log q)$ | $\mathcal{O}(mn\log q)$ |
| $\|\mathbf{sk}\|$ | $\mathcal{O}(nt\log q)$ | $\mathcal{O}(nt\log q)$ | $\mathcal{O}(nt\log q)$ |
| $\|\text{ct}\|$ | $\mathcal{O}(nN\log q)$ | $\mathcal{O}(nN\log q)$ | $\mathcal{O}(nN'\log q)$ |
| flexibledec | √ | √ | √ |
| one − timedec | × | √ | √ |
| _$\|\text{msg}\|$: message length | | _$N$: $(n+t)l$ | |
| _$\|\text{ct}\|$: cipher text length | | _$N'$: $(m+t)l$ | |

NAND gate: the same operation is true for the NAND gate, and output matrix product is $\mathbf{G} - \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$. Consider a Boolean circuit whose computational depth is $L$ while containing NAND gates. It takes the new ciphertext as input, that is, the $E$ noise ciphertext, the noise multiplied by a factor which is at most $(Nt + \sqrt{t})$ at each level, that is, the norm of the error element increases by a factor which is, at most, $(Nt + \sqrt{t})$. Therefore, the wrong element norm of the final ciphertext is bounded as $E_{\text{final}} = (Nt + \sqrt{t})^L \cdot E$.

In order to ensure the correctness of the decryption, $E_{\text{final}} \le (\lfloor (q/2) \rfloor/4)$ needs to be true. That is to say, the inequality $(Nt + \sqrt{t})^L \cdot E \le (\lfloor (q/2) \rfloor/4)$ must be true, which is guaranteed by the parameters we choose. The proof is completed. □

*4.3. IND − CPA Security Analysis.* In the following, we use Theorem 4.1 to prove that the message-encapsulation GSW scheme based on the LWE assumption that it is IND − CPA safe and that the scheme is indistinguishable from the original GSW scheme [4].

**Theorem 2.** *Let $m > n \in \mathbb{N}, q \in \mathbb{N}$ and $\chi$ be a discrete Gaussian distribution on $\mathbb{Z}$, which makes the $(n, q, \chi, m)$ − LWE problem difficult. Let $t$ be an integer that makes $t = O(\log(n))$ true. Define two distributions $\mathcal{X}$ and $\mathcal{Y}$ as follows:*

*$\mathcal{X}$ is a distribution on the $m \times (t+n)$ matrix $[\mathbf{b}_1|\cdots|\mathbf{b}_t|\mathbf{B}]$. Among them, $\mathbf{B} \in \mathbb{Z}_q^{m\times n}$ is uniformly selected, for all $1 \le i \le t$, $\mathbf{b}_i = \mathbf{B}\mathbf{t}_i + \mathbf{e}_i \pmod q$, in which $\mathbf{t}_i$ are uniformly selected from $\mathbb{Z}_q^n$, and $\mathbf{e}_i$ is selected from a discrete Gaussian distribution $\chi$.*

*$\mathcal{Y}$ is evenly distributed on $\mathbb{Z}_q^{m\times(t+n)}$.*

*Therefore, the distribution $\mathcal{X}$ and $\mathcal{Y}$ is computational indistinguishable.*

**Theorem 3.** *Let params $= (n, q, \chi, m, t)$ so that the assumption $LWE_{n,q,\chi,m}$ is true and $m = O(n\log q)$. Then, the MFHE scheme is IND − CPA safe.*

*Proof.* The proof of security contains two steps:

First, we use Theorem 4.11 to prove that, under the LWE assumption, the matrix $\mathbf{P} = [\mathbf{b}_1, \ldots, \mathbf{b}_t, \mathbf{B}] \in \mathbb{Z}_q^{m\times(n+t)}$ and the randomly chosen matrix are computationally indistinguishable

Then, using the Left-over Hash Lemma, a uniform random value $\mathbf{C}'$ can be used to replace the ciphertext $\mathbf{C} = \mathbf{MG} + \mathbf{P}^T\mathbf{R}$, that is, $\mathbf{P}^T \cdot \mathbf{R}$ is indistinguishable from the uniform distribution

The brief proof is over. See more details in [4]. □

## 5. Conclusions

In this paper, we construct an efficient message-encapsulation FHE scheme. The scheme can achieve the decryption at one time and can also flexibly decrypt bit-by-bit. In Table 1, we give a comparison of the parameters of this scheme with the existing schemes. It can be seen from the comparison that compared with the previous ones, the scheme keeps the key length substantially, and this scheme is based on more conventional assumptions and, meanwhile, reduces the ciphertext length to some extent. The proposal of this scheme makes the full homomorphic encryption take a big step from theoretical research to large-scale application. It is conducive to greatly improving the efficiency of encrypted data processing (such as retrieval and operation) in the Internet of things, saving the energy consumption of nodes in the Internet of Things, and ensuring that the data are not statistically analyzed, which has a better application scenario [29–31].

In addition, there are many interesting open issues that may be resolved in the future. For example, our thinking has certain reference value for enhancing big data security and constructing a message-encapsulated casual transmission protocol, but it also has certain challenges.

## Data Availability

No data were used in this study.

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

[1] Z. Brakerski, C. Gentry, and S. Halevi, "Packed ciphertexts in LWE-based homomorphic encryption," in *Proceedings of the Public-Key Cryptography—PKC 2013—16th International Conference on Practice and Theory in Public-Key Cryptography*, pp. 1–13, Nara, Japan, February 2013.

[2] R. Hiromasa, M. Abe, and T. Okamoto, "Packing messages and optimizing bootstrapping in GSW-FHE," in *Proceedings of the Public-Key Cryptography—PKC 2015—18th IACR International Conference on Practice and Theory in Public-Key Cryptography*, pp. 699–715, Gaithersburg, MD, USA, March 2015.

[3] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," in *Proceedings of the Advances in Cryptology—CRYPTO 2012—32nd Annual Cryptology Conference*, pp. 868–886, Santa Barbara, CA, USA, August 2012.

[4] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based," in *Proceedings of the Part I Advances in Cryptology—CRYPTO 2013—33rd Annual Cryptology Conference*, pp. 75–92, Santa Barbara, CA, USA, August 2013.

[5] Z. Li, C. Ma, E. Morais, and G. Du, "Multi-bit leveled homomorphic encryption via dual LWE-based," in *Proceedings of the Revised Selected Papers Information Security and Cryptology—12th International Conference*, pp. 221–242, Beijing, China, November 2016.

[6] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, pp. 1–34, 2009.

[7] Z. Li, S. D. Galbraith, and C. Ma, "Preventing adaptive key recovery attacks on the gentry-sahai-waters leveled homomorphic encryption scheme," *IACR Cryptology ePrint Archive*, p. 1146, 2016.

[8] Z. Brakerski and R. Perlman, "Lattice-based fully dynamic multi-key FHE with short ciphertexts," in *Proceedings of the Part I Advances in Cryptology—CRYPTO 2016—36th Annual International Cryptology Conference*, pp. 190–213, Santa Barbara, CA, USA, August 2016.

[9] Z. Li, C. Ma, and D. Wang, "Leakage resilient leveled FHE on multiple bit message," *IEEE Transactions on Big Data*, 2017.

[10] Z. Li, C. Ma, and D. Wang, "Towards multi-hop homomorphic identity-based proxy re-encryption via branching program," *IEEE Access*, vol. 5, pp. 16214–16228, 2017.

[11] Z. Li, C. Ma, and D. Wang, "Achieving multi-hop PRE via branching program," *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 45–58, 2020.

[12] Z. Li, C. Ma, and H. Zhou, "Multi-key FHE for multi-bit messages," *Sciece China Information Sciences*, vol. 61, no. 2, Article ID 029101, 2018.

[13] Z. Li, C. Xiang, and C. Wang, "Oblivious transfer via lossy encryption from lattice-based cryptography," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5973285, 11 pages, 2018.

[14] R. Lindner and C. Peikert, "Better key sizes (and attacks) for LWE-based encryption," in *Proceedings of the Topics in Cryptology—CT-RSA 2011—the Cryptographers' Track at the RSA Conference 2011*, pp. 319–339, San Francisco, CA, USA, February 2011.

[15] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," in *Proceedings of the 28th Annual International Cryptology Conference Advances in Cryptology—CRYPTO 2008*, pp. 554–571, Santa Barbara, CA, USA, August 2008.

[16] J. Hastad, R. Impagliazzo, L. A. Levin, and M. Luby, "Pseudo-random generation from one-way functions (extended abstracts)," in *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pp. 12–24, Seattle, Washigton, USA, May 1989.

[17] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract," in *Proceedings of the STOC 2009 41st Annual ACM Symposium on Theory of Computing*, pp. 333–342, Bethesda, MD, USA, May 2009.

[18] C. Peikert and B. Waters, "Lossy trapdoor functions and their applications," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pp. 187–196, British Columbia, Canada, May 2008.

[19] P. Mukherjee and D. Wichs, "Two round multiparty computation via multi-key FHE," in *Proceedings of the Part II Advances in Cryptology—EUROCRYPT 2016—35th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 735–763, Vienna, Austria, May 2016.

[20] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Proceedings of the Advances in Cryptology—EUROCRYPT 2012— 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 738–755, Cambridge, UK, April 2012.

[21] J. Katz, A. Thiruvengadam, and H. Zhou, "Feasibility and infeasibility of adaptively secure fully homomorphic encryption," in *Proceedings of the Public-Key Cryptography—PKC 2013—16th International Conference on Practice and Theory in Public-Key Cryptography*, pp. 14–31, Nara, Japan, February 2013.

[22] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, CRC Press, Boca Raton, FL, USA, Second edition, 2014.

[23] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proceedings of the FOCS 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pp. 97–106, Palm Springs, CA, USA, October 2011.

[24] M. Clear and C. McGoldrick, "Multi-identity and multi-key leveled FHE from learning with errors," in *Proceedings of the Part II Advances in Cryptology—CRYPTO 2015—35th Annual Cryptology Conference*, pp. 630–656, Santa Barbara, CA, USA, August 2015.

[25] Z. Brakerski and V. Vaikuntanathan, "Lattice-based FHE as secure as PKE," in *Proceedings of the ITCS'14 Innovations in Theoretical Computer Science*, pp. 1–12, Princeton, NJ, USA, January 2014.

[26] D. Micciancio and C. Peikert, "Trapdoors for lattices: simpler, tighter, faster, smaller," in *Proceedings of the Advances in Cryptology—EUROCRYPT 2012—31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 700–718, Cambridge, UK, April 2012.

[27] J. Alperin-Sheriff and C. Peikert, "Faster bootstrapping with polynomial error," in *Proceedings of the Part I Advances in Cryptology—CRYPTO 2014—34th Annual Cryptology Conference*, pp. 297–314, Santa Barbara, CA, USA, August 2014.

[28] L. Ducas and D. Micciancio, "FHEW: bootstrapping homomorphic encryption in less than a second," in *Proceedings of the Part I Advances in Cryptology—EUROCRYPT 2015—34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 617–640, Sofia, Bulgaria, April 2015.

[29] Z. Li, V. Sharma, C. Ma, C. Ge, and W. Susilo, "Ciphertext-policy attribute-based proxy re-encryption via constrained PRFS," *Science China Information Sciences*, vol. 64, no. 6, 2020.

[30] Z. Li, J. Wang, C. Choi, and W. Zhang, "Multi-factor password-authenticated key exchange via pythia PRF service," *Computers, Materials & Continua*, vol. 63, no. 2, pp. 663–674, 2020.

[31] V. Sharma, D. N. K. Jayakody, and M. Qaraqe, "Osmotic computing-based service migration and resource scheduling in mobile augmented reality networks (MARN)," *Future Generation Computer Systems*, vol. 102, pp. 723–737, 2020.

Hindawi
Mobile Information Systems
Volume 2020, Article ID 8813243, 11 pages
https://doi.org/10.1155/2020/8813243

*Research Article*

# Self-Controllable Mobile App Protection Scheme Based on Binary Code Splitting

**Sungtae Kim,[1] Taeyong Park,[1] Geochang Jeon,[2] and Jeong Hyun Yi[2]**

[1]*School of Computer Science and Engineering, Soongsil University, Seoul 06978, Republic of Korea*
[2]*School of Software, Soongsil University, Seoul 06978, Republic of Korea*

Correspondence should be addressed to Jeong Hyun Yi; jhyi@ssu.ac.kr

Received 17 July 2020; Revised 27 August 2020; Accepted 23 September 2020; Published 10 October 2020

Academic Editor: Navuday Sharma



Mobile apps are booming with the expansion of mobile devices such as smartphones, tablet PCs, smartwatches, and IoT devices. As the capabilities of mobile apps and the types of personal information required to run apps have diversified, the need for increased security has grown. In particular, Android apps are vulnerable to repackaging attacks, so various code protection techniques such as obfuscation and packing have been applied. However, apps protected with these techniques can also be disabled with static and dynamic analyses. In recent years, instead of using such application level protection techniques, a number of approaches have been adopted to monitor the behavior of apps at the platform level. However, in these cases, not only incompatibility of system software due to platform modification, but also self-control functionality cannot be provided at the user level and is very inconvenient. Therefore, in this paper we propose an app protection scheme that can split a part of the app code, store it in a separate IoT device, and self-control the split code through the partial app. In the proposed scheme, the partial app is executed only when it matches the split code stored in the IoT device. It does not require complicated encryption techniques to protect the code like the existing schemes. It also provides solutions to the parameter dependency and register reallocation issues that must be considered when implementing the proposed code splitting scheme. Finally, we present and analyze the results of experimenting the proposed scheme on real devices.

## 1. Introduction

Since the advent of mobile technologies, mobile apps have expanded very rapidly. According to IDC's smartphone market share report [1], smartphone shipments are expected to increase from 1.3 billion units in 2020 to 1.5 billion units in 2024 due to the launch of new devices and 5G plans. Of these, Android devices are predicted to occupy 87% of the 1.5 billion units. With the increase in the number of apps, their functions and personal information required from users are diversifying. Apps that require a variety of personal information such as smart banking, social network service (SNS), e-mail, and so on generally store users' IDs and passwords for convenience so that they automatically remain logged in. However, if a device is unlocked or infected with a virus due to an Android vulnerability [2], malware can access or steal confidential information and leak it to an attacker.

Currently, various authentication schemes [3–7], such as password, pattern, and biometric information authentication, are provided with Android smartphones. However, once the authentication is made, apps can be run without any restrictions until the smartphone is locked. In other words, unauthorized users can access personal information if they manage to pass the authentication process. In particular, Android apps are vulnerable to repackaging attacks [8], so various code protection techniques such as obfuscation and packing have been applied. However, apps protected with these techniques can also be disabled with static and dynamic analyses.

To deal with these problems, many techniques [9–11] are introduced to protect the app by modifying the platform or using root privileges. Typically, a monitor function is inserted inside an app that contains a lot of sensitive personal information to trace and control the behaviour of the app. However, this approach of modifying the app itself is

very inconvenient to apply directly at the user level. In order to overcome these shortcomings, techniques that allow users to directly protect apps by utilizing a private launcher are recently introduced [12, 13].

Therefore, in this paper, we propose a self-controllable mobile app protection scheme that can freely split binary code and authenticate using the split code to resolve smartphone security issues. The proposed scheme randomly splits the code of the target app through a launcher app, stores it in a separate IoT device, and reinstalls it after reconfiguring it as an executable app with the rest, except for the missing split code. With the proposed scheme, an app can only be run through the proposed private launcher. When the app is executed, the proposed launcher can receive the split code from a separate IoT device and deliver the split code to the app for execution. At this moment, a code-based authentication scheme is used, so only authenticated code can be run in the app and there is no need for a complicated cryptographic authentication. By using this scheme, only the user who has the split code can run the app, thereby improving the security of the smartphone by preventing the unauthorized user from running the app. In addition, the proposed scheme can be applied at the app level, so no platform modification and root privileges are required. The user can simply improve the security of personal information by installing the app.

In order to implement the code splitting function, which is the core part of the proposed scheme, a parameter dependency problem and a register reallocation problem inevitably occur. In this paper, solutions to these problems are presented in detail along with sample codes. It also describes the results of measuring feasibility and performance overhead of the proposed scheme on real Android device and smartwatch.

This paper is organized as follows. Section 2 addresses the related work. Section 3 provides the background and motivation behind the proposed scheme. Section 4 describes the design of the proposed scheme. Section 5 describes issues that arise when implementing the proposed scheme and their solutions. Section 6 demonstrates the experimental results with the proposed scheme. Finally we conclude the paper in Section 7.

## 2. Related Work

Protecting mobile apps by modifying the platform or using root privileges is inconvenient and difficult for users to apply directly. Many techniques [9–11] have been developed to modify and protect the app itself. I-arm-droid [9] identifies security-sensitive API methods and specifies security policies for the app. It also improves security by rewriting bytecodes in policies by monitoring apps. Aurasium [10] does not require modification of the Android OS to provide the security and policy desired by the user. This tool also monitors behaviour for privacy breaches, such as attempts to retrieve sensitive information from users or access malicious IP addresses. However, in such methods, a monitor function should be inserted inside an app that contains a lot of sensitive personal information to control the behaviour of the app. However, this method of modifying the app itself is very inconvenient to apply directly at the user level.

Recently, many protection schemes have been introduced through the launcher app [12, 13] to help users manage the app comfortably. In general, Android launcher refers only to a program that runs a home screen in the user interface (UI) [14–16]. In most cases, it consists of home screens and app drawers, and it can be seen that it is included in the Android UI. In addition, the app is always running while the terminal is running, and additionally, the home screen area can be provided to arrange shortcut icons or widgets of the app so that the developer can execute the desired function, such as executing or deleting other apps. The manufacturer's launcher is designed as the default launcher from booting, but as the new launcher is installed, a selection window pops up from the home button and allows selecting the installed launcher. To change the default launcher that is already specified, we can use the launcher to clear the default task or use a separate app. Boxify [17] is a representative example of a protection technique through a launcher app. It executes the target app through the launcher app, which is an isolated process with minimal privileges, and monitors it through hooking to control untrusted apps from doing actions that cause damage such as personal information leakage.

In addition, an example of applying the code splitting technique to Robot OS (ROS), an embedded software for smart cars, was recently introduced [18]. This study applied the code splitting scheme for the purpose of secure booting to prevent an attacker from remotely controlling the smart car. In addition, while this uses code splitting for native code, the proposed scheme is applied for Android bytecode. Except for the concept, the detailed underlying techniques such as parameter dependency checking and register reallocation are completely different.

## 3. Background

*3.1. Android App.* Android apps are provided in a single file called the Android package (apk) file. The apk file is in zip format and consists of classes.dex, which contains not only the app's code, but also resource files that contain configuration information such as the app's icons, images, and strings. Each apk file contains an AndroidManifest.xml file containing the app's components and permission information. The main language of the Android app is Java. Java code is compiled into the Dalvik bytecode and consists of a file called classes.dex. The generated bytecode is executed on the Dalvik virtual machine. In addition, developers can use native library (.so) written in the C or C++ language. These native library codes run directly on the processor of the device, not on the Dalvik virtual machine.

All apps can be identified by a unique package name and are self-signed by the developer's private key [19]. Android apps consist of different types of components: Activity, Service, Broadcast Receiver, and Content Provider. The Activity represents functions that are performed through a UI. A single app can consist of several activities. In contrast, the Service runs in the background without a UI. For

example, a music player app might require a UI for selecting songs, but no additional UI is required while music is playing. This task can be implemented as a Service. The Broadcast Receiver is a function that can receive the message service and perform the corresponding action when a system event occurs in Android. Finally, the Content Provider is used to provide app data to other apps.

*3.2. Android Repository.* Android has internal storage and external storage [20]. The internal storage primarily stores systems and apps, while data is stored in the primary external storage. The internal storage can read and write data only in the apps, and the external storage is used as a common area. Also, the data in the internal storage is deleted when the app is deleted. The external storage contains photos, videos, and other files. With permission, it is possible to read and write data, which is in the external storage, from other apps. There is a cache area, a database area, and a file area in the internal storage that exists for each app. Since it is troublesome to find the necessary path whenever the path of each area is required, Android provides an API that easily obtains the main path where data is stored.

*3.3. ASMDEX.* ASMDEX [21] is an open-source project which parses the dex file and organizes it into a tree. It allows the user to modify, add, or delete the generated tree and rebuild it as a dex file. The tree structure created by ASMDEX is shown in Figure 1. When constructing a dex file as a tree, the root node is represented by ApplicationNode. ApplicationNode has member variables called classes, which represents a list of ClassNodes classes. ClassNode parses and holds all information, such as name, authority, and method for every class in a dex file. A member variable, method, represents a list of MethodNodes classes. MethodNode is the information of method contained in ClassNode. Similar to ClassNode, MethodNode parses and contains information about method, such as name, descriptors, exceptions, and number of registers used. Unlike ClassNode, MethodNode may have a duplicate name. In such a case, the method is identified through a descriptor. The MethodNode class has a member variable, instruction, which is a class called InsnList that implements a double linked list for AbstractInsnNode. AbstractInsnNode is an abstract class, and a method inherits the corresponding abstract class and executes each instruction.

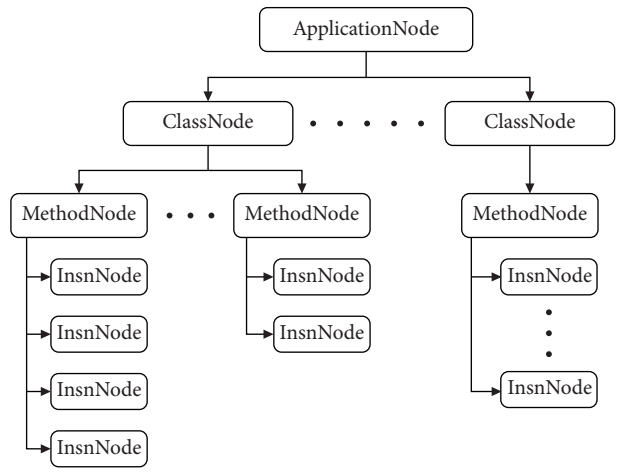



Figure 1: The dex file tree generated from ASMDEX.

## 4. Proposed Scheme

When the user authentication is performed once before use, anyone can run all the apps installed in a smartphone until it is locked, thereby allowing personal information leakage. This section proposes a scheme to protect personal information by implementing app execution environment through the self-controllable private launcher.

*4.1. Concept.* The basic idea of the proposed scheme is to split and manage a part of the binary code of the app safely and separately and to take the split code at runtime and functionally assemble it to operate the same as the original code. More specifically, a part of the binary code of the app is split and stored in the IoT device, and each time the app is executed, the split code stored in the IoT device is taken and assembled functionally through a code-based authentication. We call this launcher app that provides this functionality an AppContainer, which is provided in two modes: Normal or Protected modes. If the target app is given as input to the AppContainer, it will enter the Normal mode by default. As shown in Figure 2, when the target app is executed in the Normal mode, the binary code splitting function is operated first. A part of the binary code of the app is randomly selected, split, and then stored in an IoT device. The rest of the code is incomplete, but apparently reconstructed to take the form of the app and reinstalled on the smartphone. This incomplete-but-normal-looking app is called a partial app in the rest of this paper. At this time, the partial app can be run only in Protected mode. That is, after switching to Protected mode, all apps displayed in the AppContainer are partial apps. When running this partial app in the Protected mode, the corresponding split code is received from the IoT device. Then, it operates in the same way as the original app through the code-based authentication protocol.

The dex file can be decompiled into smali code at any time, so it is possible to parse the method in the executable. There can be up to 65536 methods in one dex file. Currently, methods are randomly selected. Even if the proposed scheme is applied to the same app several times, different pairs of split and remaining codes can be generated each time. Thus, by uniquely creating a split code, only the owner of the IoT device can run the app. In addition, if you select the core logic and then store it in the split code and configure only the less important code in the partial app, the core logic of the original app can still be protected even if the partial app is exposed to reverse engineering.

*4.2. Design Details.* The proposed scheme consists of two main phases: binary code splitting for the target app and applying the code-based authentication to the partial app.

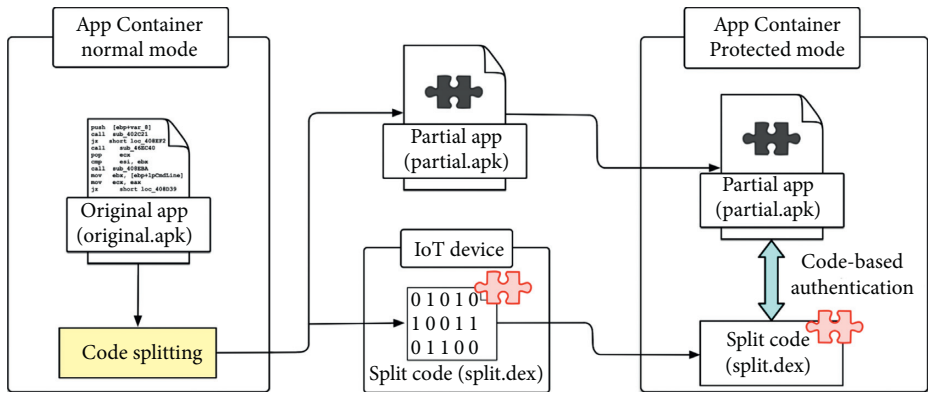



Figure 2: Concept of proposed AppContainer scheme.

*4.2.1. Binary Code Splitting.* Like the general launcher app, except for system apps, the list of apps installed by the user is displayed in the form of icons in the Normal mode. At this time, when a specific app is selected, the code of the app is split, the rest of the app is rebuilt as a partial app, and a code-based authentication function is additionally applied. This rebuilt partial app is only displayed in the Protected mode and no longer visible in the Normal mode. The outline of how the Normal mode operates is shown in Figure 3. A target app (original.apk) can be selected to apply the code splitting scheme. The package name of the target app goes to the internal memory path/data/app/"packagename"/. This path can only read the original.apk files of the target apps because read-only permission exists for other apps. Then, the original.apk file is copied to external storage such as an/sdcard. To use external storage, the AppContainer must have READ_EXTERNAL_STORAGE and WRITE_EXTERNAL_STORAGE permissions. Next, the imported original.apk file is unzipped into the original.dex file. After the code splitting process, the original.dex file is split into the partial.dex and the split.img files. The split.img file is reconstructed into the split.dex file with a wrapping function. When the reconstruction is complete, a hash value on the split.dex file is created with the application name. It will be used later for code-based authentication. Then the split.dex file is sent to the connected IoT device. In the other side, the folder containing the partial.dex file is recompressed to create in form of a partial.apk file. This partial.apk file is resigned with the user's private key. Then, the original.apk file is replaced with the partial.apk file in the internal storage. After completing the code splitting process, all files used in the/sdcard path are deleted.

The detailed process of the code splitting scheme is given in Algorithm 1. Given the original.apk file, the original.dex inside the original.apk file is turned into a tree through ASMDEX. It traverses the created tree and randomly selects (we note that it is impossible to split any part of the Android application. This is always applicable only for user-defined classes and methods. It does not apply to classes or methods with framework code or system dependencies. Also, the onCreate function of the MainActivity class, which is the basis for app running, or a class that is automatically created by the system such as R$ should not be selected. However, there is no problem for practical application because sensitive or secret code logic, which is the main target to be split in this proposed scheme, is all user-defined classes and methods) specific splitClass and splitMethod. When the split node selection is complete, ASMDEX is used to create a new tree. The selected splitMethod is added to the splitClass node to create a new dex tree. Since the selected splitMethod disappears from the existing tree, it is necessary to modify the caller part and splitMethod body part. If the selected splitMethod is Static, the splitMethod body part needs to be modified; otherwise, the caller part needs to be modified. If it is not Static method, the splitMethod and splitClass are changed into Abstract, and all splitMethod bodies are deleted. The reason for changing to Abstract is to allow a splitClass with a splitMethod to inherit the existing class and use the undefined methods and field values of the existing class.

Consequently, it traverses the existing targetTree and finds all caller parts of the splitMethod. If a splitMethod is used as an existing class, it is changed to an Abstract method, and thus a splitMethod cannot be used by creating an existing class. The existing class part should be replaced by the stub code, which finds a splitClass that inherits an existing class. A split.dex file is created including the splitClass and then is returned. In the case of Static method, the caller should be replaced with the contents of calling the stub code without removing the caller part. In the split code, the following actions are performed: find and create a splitClass that inherits the existing class, execute the splitMethod immediately, and return the result value of the splitMethod. Inserting or deleting other codes may cause parameter and register dependency problems that may conflict with existing registers because the Dalvik bytecode is register-based, not stack-based. To solve this problem, we have to deal with a parameter dependency checking and a register reallocation, which are explained in Sections 5.1 and 5.2, respectively.

*4.2.2. Split Code Integrity Checking.* In the Normal mode, when AppContainer separates the split.dex and transmits it to an IoT device, it stores IMEI (International Mobile Equipment Identity) information of the device in the

```
Input: target app (original.apk)
Output: remaining target tree and split code tree
(1)  Tree targetTree, newTree;
(2)  Class splitClass;
(3)  Method splitMethod;
(4)
(5)  targetTree = makeTree (original.dex);
(6)  selectSplittingTarget (targetTree, splitClass, splitMethod);
(7)
(8)  ASMDEX.init (newTree);
(9)  ASMDEX.makeClassNode (newTree, splitClass);
(10) ASMDEX.makeMethodNode (newTree, splitMethod);
(11)
(12) if splitMethod.Type = = STATIC then
(13)    convertMethodCalleeToStub (targetTree, newTree, splitClass, splitMethod);
(14) else
(15)    convertToAbstract (targetTree, splitClass, splitMethod);
(16)    deleteMethodBody (targetTree, splitMethod);
(17)
(18)    for Class class:targetTree.Classes do
(19)       if findMethodCaller (class, splitMethod) = = true then
(20)          convertToStub (class, newTree, splitMethod);
(21)       end if
(22)    end for
(23) end if
```

Algorithm 1: Pseudocode for splitting original app.

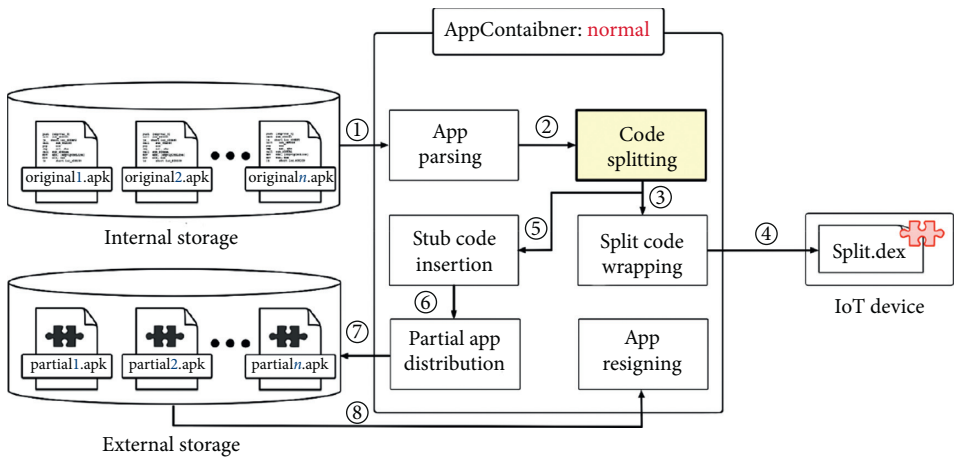



Figure 3: Normal mode operation of the AppContainer.

internal storage. After that, when AppContainer requests split.dex from the IoT device, the IoT device creates a hash value using not only split.dex, but also IMEI and salt. The IoT device transmits the remaining split.dex, salt, and hash value excluding IMEI information to the AppContainer. Then, AppContainer calculates the hash value using split.dex, salt received from the IoT device, and IMEI stored in the internal storage and then checks whether it matches the hash value received from IoT. If the two hash values match, AppContainer proves that it has received the split.dex file from a trusted IoT device and that the integrity of split.dex is verified.

*4.2.3. Code-Based Authentication.* In the Protected mode, only partial apps with code splitting scheme are displayed in the form of icons on the home screen area. As shown in Figure 4, when the partial.apk starts, it requests its corresponding split.dex to the IoT device. The partial.apk remains on standby until the split.dex file is transmitted from the IoT device to AppContainer. Upon downloading the split.dex file to AppContainer, as explained in Section 4.2.2, AppContainer checks the integrity of the split code and stores it in the internal storage (/data/data/"packagename"/). Then, check if split.dex and corresponding original.dex work normally. If it is wrong, the partial.apk does not work

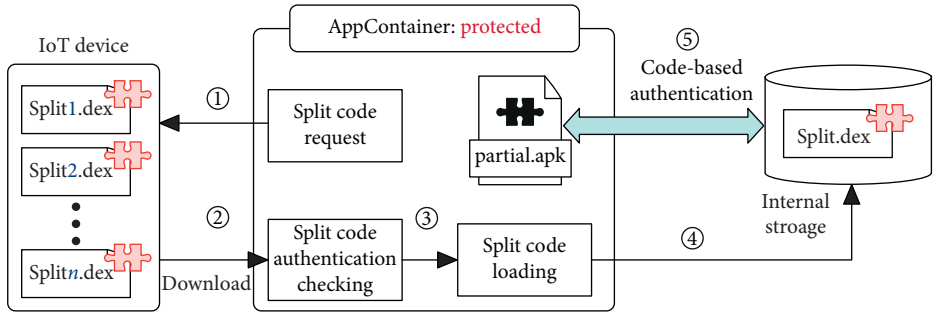



Figure 4: Protected mode operation of the AppContainer.

anymore and is terminated. When the partial.apk terminates abnormally, the split.dex files created during download are deleted. After that, when the partial.apk receives the split.dex file stored in the internal storage through the Intent, the partial.apk can be normally executed. Once again, if you try to run a partial.apk on a general launcher other than AppContainer, the partial.apk does not work because its corresponding split.dex file does not exist.

## 5. Implementation Issues

In this section, we present several issues and solutions to implement the code splitting scheme described above. In regard to code-based authentication, since there are no implementation problems, we focus on the issues for the code splitting scheme.

*5.1. Parameter Dependency Checking.* As shown in Figure 5, given the original.dex file, it is divided into the partial.dex and split.img files. When the partial.dex is transformed to the partial.apk, there are important implementation issues on the parameter dependency checking and the register reallocation.

To perform the same operation as before splitting, the splitClass instead of an existing class should be created since the splitMethod is replaced with an Abstract method. In the stub code, a new class that inherits the existing class is created instead (refer to Figure 6). Since the splitClass has always different shape, the type and number of parameters required for class creation are different, so the number of registers used is different. To generically solve this problem, the parameter dependency should be resolved by adding three registers to the method that contains the caller part. Reusing an existing register can cause conflicts with the other code, so only the new register is used. The first register is a register containing name information of a splitClass that inherits an existing splitClass. The second register is an object array that can hold the constructor parameters. The reason for using an object array is that the number of registers used is different because the number and type of parameters in the splitClass constructor are different each time. Therefore, several parameter registers are managed as one register and sent to the stub code to generically fix the caller part. When creating an array of objects, a register is created using the init() in the original code and moved to the second register. The last register is the index register that controls the object array. In addition, parameters of primitive types such as Integer and Double cannot be put directly into the object array, but they must be converted to Integer and Double types using the valueof() function. Therefore, before putting it into the object array, the type is converted into the array by using the register used as a parameter register.

*5.2. Register Reallocation.* The register dependency problem occurs because it does not match the number of registers previously used. To solve this problem, register reallocation is additionally needed. This task is to solve the index conflict caused by three registers added to resolve the parameter dependency. As shown in Table 1, method registers used in the Dalvik bytecode [22] can be divided into local registers and parameter registers. Local registers are numbered from the beginning, and in the case of parameter registers, the last register is used in all registers. The register before the parameter register is this register that represents the method itself. Thus, adding three registers changes the total number of registers and may cause a malfunction during the execution because the modified register is accessed; thus, the relocation of the registers is necessary.

To solve this problem, at the start of the splitMethod, this register and the parameter registers are returned to the register position before adding the register. As shown in Table 2, when 5 registers are used in the splitMethod and 2 parameters are used, v3 and v4 registers have first and second parameters, and v2 register has this register. If three registers are added, the first parameter goes into the v6 register, the second parameter goes into the v7 register, and this register goes into the v5 register. In this case, if v2, v3, or v4 is used in the original code, an error occurs because the desired value is not included. Therefore, the values of v5, v6, and v7 are put back to v2, v3, and v4. Then the added v5, v6, and v7 registers are used to resolve parameter dependencies.

If Double and Long of the parameter register type are used, two registers are used instead of one. Also, by adding registers, the total number of registers used in the splitMethod may be over 16. In some cases, more than 16 registers of the existing splitMethod may be used. For example, invoke-virtual should be used when using registers less than 16, but invoke-virtual/range should be used when using registers above 16. In addition, the number of registers to be used must

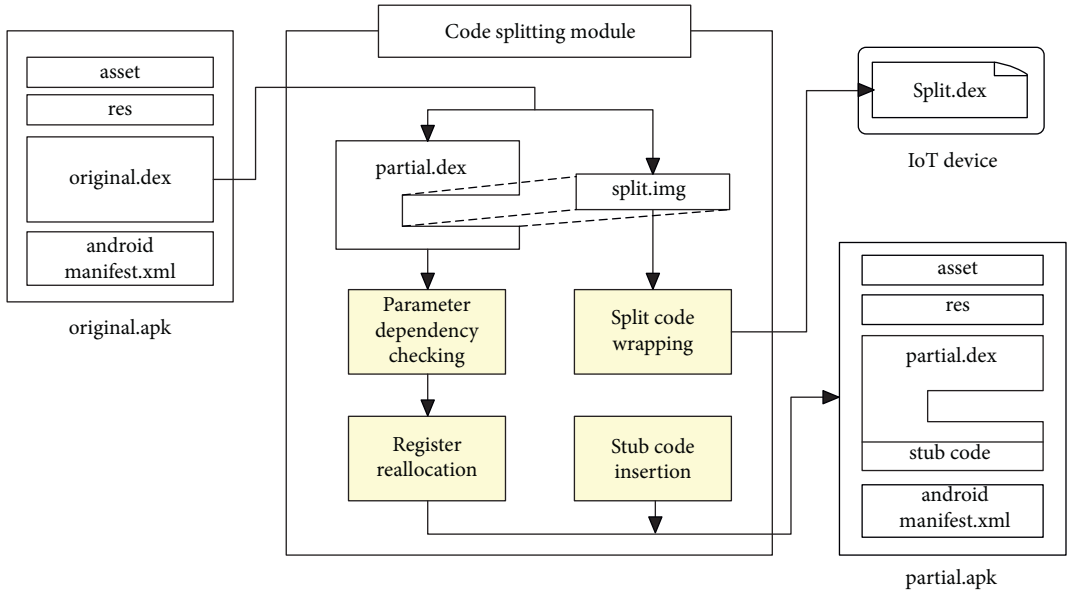



Figure 5: Implementation issues with code splitting.

```
SplitCodeCaller(…)
{
    …
    RandomClass a = new RandomClass(…)
    a.RandomMethod(…)
    …
}
```

⇨

```
SplitCodeCaller(…)
{
    …
    RandomClass a = (RandomClass)StubcodeCall(…)
    a.RandomMethod(…)
    …
}
```

Figure 6: Method caller change for solving parameter dependency problem.

Table 1: Dalvik register allocation (before).

| Variables | Parameters | Description |
|---|---|---|
| v0 | | Local register |
| v1 | | Local register |
| v2 | p0 | This-register |
| v3 | p1 | First parameter register |
| v4 | p2 | Second parameter register |

Table 2: Dalvik register allocation (after).

| Variables | Parameters | Description |
|---|---|---|
| v0 | | Local register |
| v1 | | Local register |
| v2 | | Class name register |
| v3 | | Parameter information register |
| v4 | | Index register |
| v5 | p0 | This-register |
| v6 | p1 | First parameter register |
| v7 | p2 | Second parameter register |

be sequential. When invoke-virtual is available, three registers such as v5, v8, and v3 are available. But when invoke-virtual/range is available, the registers should be v5, v6, and v7. In addition, more than 16 registers cannot new-array and thus cannot create object arrays. The object array is created and relocated using the init() command register, which uses the register below 16 unconditionally, as described above.

*5.3. Stub Code Insertion.* The stub code needs to be injected in two cases. The first is the case that the AppContainer needs to get the split.dex and store it in internal memory when the app first starts. The second is necessary to load the split.dex from the internal memory when the split.dex is called and to execute the splitClass from the split.dex. The first case analyzes the AndroidManifest.xml and inserts stub code into the Activity class that starts first when the app is run. You also need to modify the beginning of the onCreate() function to add a call to the inserted stub code when the app starts. Moreover, we need to check whether the split.dex received from the AppContainer is the corresponding the split.dex to the partial.apk. If the checking is correct, save the split.dex in the internal memory. If not, terminate the program. The code that loads the split.dex is inserted by adding a splitClass node to the tree created by ASMDEX. The stub code is executed when the caller invokes the split.dex. We create a DexClassLoader object and load the split.dex stored in the internal memory into the DexClassLoader object. Find the splitClass in the created DexClassLoader object and execute the splitMethod normally. Therefore, it is an object that has a class name and constructor parameter value. It finds the desired splitClass by using the reflection API provided by Java. In the case of the Static method, the method finds and executes the method through the object that has the class name, method name, and method parameter value.

*5.4. Resigning.* When all the code splitting procedures are done, the folder containing the partial.dex is recompressed to create an partial.apk file. Using ASMDEX, a new tree created with split.img is created as a split.dex file. This split.dex file is distributed to the connected IoT device. The Android app must be digitally signed before distribution. Since the original.apk file was modified during the splitting process, the previous signature is useless, so resigning is required to install the app. Therefore, the user's signing key stored in the AppContainer is used. When all the resigning is done, the existing original.apk is deleted, and the partial.apk is reinstalled.

## 6. Experimental Results

In this section, we describe the results of evaluating performance of the proposed scheme. We implement and measure the performance on an Android version 6 or later and Galaxy Gear for an IoT device.

*6.1. Sample Codes with Code Splitting.* When the code splitting scheme is applied, the target method is randomly chosen from all the methods. As shown in Figure 7, the addFont method of jxL/biff/Fonts class is selected among all methods and converted into an Abstract method, and the method body disappeared and its size became zero.

Figure 8 shows the caller part of a splitMethod. Previously, only 7 registers from v0 to v6 were used. Three registers were added to modify 10 registers from v0 to v9. We also reallocated the parameter register and this-register through the move-object at the start of the method to avoid register conflicts.

Figure 9 is the part that creates class before calling the splitMethod. It creates an object array using the register used to execute the Init() function and stores the object array in the added register, v8. The register v9 was not used because there were no parameters in the constructor of the splitClass, and the object[] array was also created with a size of zero.

The name of the class to create is stored in the register v7. The class name and the object array to be created are sent with the parameters for calling the stub code, and the/range command is used in case the register number becomes 16 or more. The generated class is cast to the original class and stored in the register v0 because the original class uses the register v0 in the code before modification.

Figure 10 shows the splitClass and splitMethod in the split.dex file stored in the IoT device. In the example code above, there is a splitMethod in the newly named class that inherits the selected class.

If the static method is selected for splitting as in Figure 11, the change of the caller part is not necessary and only the body of the splitMethod is changed. The changed code executes the method by sending class name, method name to execute, and parameters of the method to stub code. It then processes the parameter information received by the method, converts the result to the original return format, and delivers it.

```
com/example/secrettest/Secret4 extends java/lang/Object
    method: <init>()V [public constructor] size:4
    method: sum1(I I)I [public] size:3
    method: sum2(I I I)I [public] size:3
```

⇩

```
com/example/secrettest/Secret4 extends java/lang/Object
    method: <init>()V [public constructor] size:4
    method: sum1(I I)I [public] size:3
    method: sum2(I I I)I [public abstract] size:0
```

Figure 7: Method change with code splitting.

```
# Parameters:
# - Local registers:v0…v11
# - v12:android.view.View
#
# - return:void

onClick-BB@0x0 :
    const/4                          v10, 1
    const/4                          v9, 3
    const/4                          v8, 2
```

⇩

```
# Parameters :
# - Local registers :  v0…v14
# - v15 : android.view.View
#
# - return:void

onClick-BB@0x0 :
    move-object                      v11, v14
    move-object                      v12, v15
    const/4                          v10, 1
    const/4                          v9, 3
    const/4                          v8, 2
```

Figure 8: Register rearrangement followed by register addition.

*6.2. Performance Overhead.* We tested whether the proposed scheme is properly applied to real apps in the Google Play Store and evaluated the execution overhead by comparing the launching time of apps with the proposed scheme and apps without it.

Table 3 shows the launching speed of the app before and after applying the proposed scheme to five apps by category in Google Play Store. Experimental results show that the proposed scheme has a delay time of 138 milliseconds on average, although the delay time is different for each app. This delay is caused by the time required to load the split code when the app starts and to check the authenticity of the split code received from the AppContainer. Because each app has different size and functions, its launching time before and after applying the proposed scheme is different. The fastest launching time is 163 milliseconds, and the slowest one is 975 milliseconds. Looking through the experimental results, it can be seen that the overhead due to the proposed scheme increases by about 15% to 2 times. However, the average increase of 138 milliseconds is reasonable, making the launching delay of the proposed scheme acceptable.

```
invoke-virtual      v3, v5, v6, Lcom/example/secrettest/Secret3;->e(Ljava/lang/
new-instance        v4, Lcom/example/secrettest/Secret4;
invoke-direct       v4, Lcom/example/secrettest/Secret4;-><init>()V
const/4             v5, 6
invoke-virtual      v4, v9, v8, v5, Lcom/example/secrettest/Secret4;->sum2(I I
```

⇩

```
invoke-virtual      v3, v5, v6, Lcom/example/secrettest/Secret3;->e(Ljava/lang/
const/4             v4, 0
new-array           v4, v4, [Ljava/lang/Object;
move-object         v14, v4
const-string        v13, 'com.example.secrettest.rStUVNacyAlA'
invoke-static/range v13 … v14, Ledu/ssu/msec/sot/a;->gI(Ljava/lang/String;[java
move-result-object  v13
move-object         v4, v13
check-cast          v4, Lcom/example/secrettest/Secret4;
const/4             v5, 6
invoke-virtual      v4, v9, v8, v5, Lcom/example/secrettest/Secret4;->sum2(I I
```

Figure 9: Caller part of changed method with code splitting.

```
Lcom/example/secrettest
    com/example/secrettest/rStUVNacyALA extends com/example/secrettest/Secret4
        method: <init> () [public constructor] size:4
        method: sum2 (I I I)I [public] size:3
```

Figure 10: Split code stored in IoT devices.

```
sum-BB@0x0 :
    (00000000) add-int          v0, v1, v2
    (00000004) return           v0
```

⇩

```
sum-BB@0x0 :
    const-string        v0, 'com.example.secrettest.FpWcUBZqSvRa'
    const-string        v3, 'sum'
    const/4             v1, 2
    new-array           v1, v1, [Ljava/lang/Object;
    invoke-static       v4, [Ljava/lang/Integer;->valueOf(I)Ljava/lang/Integer;
    move-result-object  v4
    const/4             v2, 0
    aput-object         v4, v1, v2
    invoke-static       v5, Ljava/lang/Integer;->valueOf(I)Ljava/lang/Integer;
    move-result-object  v5
    const/4             v2, 1
    aput-object         v5, v1, v2
    invoke-static       v0, v3, v1, Ledu/ssu/msec/sot/a;->gStatic(Ljava/lang/S
    move-result-object  v0
    check-cast          v0, Ljava/lang/Integer;
    invoke-virtual      v0, Ljava/lang/Integer;->intValue()I
    move-result         v0
    return              v0
```

Figure 11: Static method definition part with the proposed scheme.

Table 3: Comparison of launching time.

| | Lifestyle apps | Banking apps | Finance apps | Education apps | Test apps |
|---|---|---|---|---|---|
| Original app | 0.31 (sec) | 0.98 (sec) | 0.19 (sec) | 1.00 (sec) | 0.14 (sec) |
| App with split code | 0.44 (sec) | 1.14 (sec) | 0.32 (sec) | 1.13 (sec) | 0.28 (sec) |

Table 4: Feature comparison between code protection solutions.

| | DexProtector | DexGuard | AppContainer |
|---|---|---|---|
| Class protection | Encryption | Encryption | Code splitting |
| Method protection | Encryption | Encryption | Code splitting |
| Reversing resistance | Static | Static | Dynamic |
| Side effect | — | — | Device authentication |

*6.3. Feature Comparison.* Table 4 shows the feature comparison of AppContainer with typical commercial tools for software code protection for Android. The existing tools such as DexGuard [23] and DexProtector [24] adopt encryption to protect methods and classes, but the proposed scheme utilizes code splitting to protect the code without

encryption. Since Android bytecode can automatically recover encrypted code by using advanced dynamic analysis tools [25], the existing tools with encryption can prevent static analysis, but have the disadvantage of being exposed to dynamic analysis. On the other hand, the proposed AppContainer does not expose the complete code even when attempting dynamic analysis of the partial app because a part of the code exists in the external device. Therefore, the proposed scheme can resist dynamic analysis as well as static analysis without applying any encryption techniques. Recall that the split code is physically stored on an external device, and the partial app is stored on a smartphone. In the proposed scheme, since the code works normally only when the pair of partial app and split code must match each other, the app runs normally means that the external device that stores the split code can be trusted. In other words, this means that device authentication is obtained as a side effect.

## 7. Conclusion

As many apps require personal information, such as smart banking, SNS, and e-mail, the importance of personal information protection is also increasing. However, most users keep their auto-login status by storing their ID and password even though they are apps with sensitive personal information for convenience. Smartphones are protected by various authentication methods such as the PIN, patterns, and biometric information authentication, but they fall short of providing the utmost security of personal information. Thus, we proposed a scheme that protects the app from unauthorized users by assigning control of app execution by merely installing the app without modifying the platform of the smartphone.

The AppContainer is designed to meet the following design goals. First, an app with the proposed scheme requires user authentication before running the app so that unauthorized users cannot run the app itself. The AppContainer is responsible for receiving the split code from the IoT device and communicating it with the app. Therefore, only authenticated users who have a split code on the IoT device can run the app through the AppContainer. Secondly, it can be applied simply as an app-level protection technique rather than a platform modification. Existing protection techniques have enhanced the security by changing the platform of the smartphone, but since the proposed scheme does not require any platform change, it can be used on any platform by any user.

In addition, the AppContainer shows a list of apps with code splitting so that the user can recognize which apps have code-based authentication. Just in case, if the misbehaving app is reinstalled due to a repackaging attack, it is excluded from the list so that users can easily recognize that it is not an existing app. In conclusion, the proposed AppContainer is expected to prevent personal information leakage by effectively avoiding app execution by unauthorized users. As a future work, we intend to expand and develop the proposed scheme by applying the code splitting technique not only to Android but also to various embedded software such as smart vehicles, robots, and drones.

## Data Availability

All data generated or analysed during this study are included in this published article.

## Disclosure

The authors disclose that this manuscript is an expanded and improved version of the master's thesis [26] by the first author, S. Kim.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea Government (MSIT) (No. 2017-0-00168, Automatic Deep Malware Analysis Technology for Cyber Threat Intelligence) and in part by the Mid-Career Researcher program through the National Research Foundation of Korea (NRF) funded by the MSIT (Ministry of Science and ICT) under Grant NRF-2020R1A2C2014336.

## References

[1] M. Chau and R. Reith, "Smartphone market share," 2020, https://www.idc.com/promo/smartphone-market-share/.

[2] W. Winder, "28 million android phones exposed to "eye-opening" attack risk," 2020, https://www.forbes.com/sites/daveywinder/2019/08/03/28-million-android-phones-exposed-to-eye-opening-attack-risk/#761afc4a7b74.

[3] M. A. Ferrag, L. Maglaras, A. Derhab, and H. Janicke, "Authentication schemes for smart mobile devices: threat models, countermeasures, and open research issues," *Telecommunication Systems*, vol. 73, no. 2, pp. 317–348, 2020.

[4] Z. Lin, W. Meng, W. Li, and D. S. Wong, "Developing cloudbased intelligent touch behavioral authentication on mobile phones," in *Deep Biometrics*, pp. 141–159, Springer, Berlin, Germany, 2020.

[5] A. O. Ekpezu, E. E. Umoh, F. N. Koranteng, and J. A. Abandoh-Sam, "Biometric authentication schemes and methods on mobile devices: a systematic review," in *Modern Theories and Practices for Cyber Ethics and Security Compliance*, W. Yaokumah, M. Rajarajan, J. Abdulai, I. Wiafe, and F. A. Katsriku Eds., IGI Global, Hershey, PA, USA, pp. 172–192, 2020.

[6] Q. Li, P. Dong, and J. Zheng, "Enhancing the security of pattern unlock with surface EMG-based biometrics," *Applied Sciences*, vol. 10, no. 2, p. 541, 2020.

[7] M. Guerar, L. Verderame, A. Merlo, F. Palmieri, M. Migliardi, and L. Vallerini, "CirclePIN: a novel authentication mechanism for smartwatches to prevent unauthorized access to IoT devices," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 3, pp. 1–19, 2020.

[8] J.-H. Jung, J. Y. Kim, H.-C. Lee, and J. H. Yi, "Repackaging attack on android banking applications and its countermeasures," *Wireless Personal Communications*, vol. 73, no. 4, pp. 1421–1437, 2013.

[9] B. Davis, B. Sanders, A. Khodaverdian, and H. Chen, "I-armdroid: a rewriting framework for in-app reference monitors

for android applications," *Mobile Security Technologies*, vol. 2012, no. 2, pp. 1–7, 2012.

[10] R. Xu, H. Saïdi, and R. Anderson, "Aurasium: practical policy enforcement for android applications," in *Proceedings of the 21st USENIX Security Symposium*, Bellevue, WA, USA, 2012.

[11] B. Davis and H. Chen, "RetroSkeleton: retrofitting android apps," in *Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, Taipei, Taiwan, 2013.

[12] M. Backes, S. Gerling, C. Hammer, M. Maffei, and P. von Styp-Rekowsky, "Appguard-real-time policy enforcement for third-party applications," Technical report A/02/2012, Saarland University, Saarbrücken, Germany, 2012.

[13] A. Bianchi, Y. Fratantonio, C. Kruegel, and G. Vigna, "NJAS: sandboxing unmodified applications in non-rooted devices running stock android," in *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, Denver, CO, USA, 2015.

[14] Go Launcher, 2020, http://www.goforandroid.com/.

[15] ADW Launcher, 2020, http://jbthemes.com/anderweb/.

[16] LauncherPro, 2020, http://www.launcherpro.com/.

[17] M. Backes, S. Bugiel, C. Hammer, O. Schranz, and P. von Styp-Rekowsky, "Boxify: full-fledged app sandboxing for stock android," in *Proceedings of 24th USENIX Security Symposium*, Washington, DC, USA, 2015.

[18] J. Yoo and J. H. Yi, "Code-based authentication scheme for lightweight integrity checking of smart vehicles," *IEEE Access*, vol. 6, pp. 46731–46741, 2018.

[19] Oracle, "Understanding signning and veification," 2020, https://docs.oracle.com/javase/tutorial/deployment/jar/intor.html.

[20] H. Kim, N. Agrawal, and C. Ungureanu, "Examining storage performance on mobile devices," in *Proceedings of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds*, Cascais, Portugal, 2011.

[21] ASMDEX, 2020, http://asm.ow2.org/doc/tutorial-asmdex.html.

[22] Android Open Source Project, 2020, https://source.android.com/index.html.

[23] DexGuard, 2020, https://www.guardsquare.com/en/products/dexguard.

[24] DexProtector, 2020, https://dexprotector.com/.

[25] H. Cho, J. H. Yi, and G.-J. Ahn, "DexMonitor: dynamically analyzing and monitoring obfuscated android applications," *IEEE Access*, vol. 6, pp. 71229–71240, 2018.

[26] S. Kim, "Self-controllable mobile app protection scheme based on binary code splitting," Master degree thesis, Soongsil University, Seoul, Republic of Korea, 2017.

Hindawi
Mobile Information Systems
Volume 2020, Article ID 8844116, 9 pages
https://doi.org/10.1155/2020/8844116

*Research Article*

# An Intrusion Detection Scheme Based on Repeated Game in Smart Home

**Rui Zhang,[1] Hui Xia,[2] Shu-shu Shao,[1] Hang Ren,[1] Shuai Xu,[1] and Xiang-guo Cheng[1]**

[1]*The College of Computer Science and Technology, Qingdao University, Qingdao 266100, China*
[2]*The College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China*

Correspondence should be addressed to Hui Xia; xiahui@qdu.edu.cn and Xiang-guo Cheng; 15964252399@163.com

Received 25 June 2020; Revised 17 August 2020; Accepted 28 August 2020; Published 17 September 2020

Academic Editor: Vinod Karar



Smart Home brings a new people-oriented home life experience. However, the edge devices in this system are facing severe threats such as data security and equipment safety. To solve the above problems, this paper proposes an intrusion detection scheme based on repeated game. We first use the K-Nearest Neighbors (KNN) algorithm to classify edge devices and equip the intrusion detection system to cluster heads. Secondly, we use the regret minimization algorithm to determine the mixed strategy Nash equilibrium of the one-order game and then take a severe punishment strategy to domesticate malicious attackers. Thirdly, the intrusion detection system can detect malicious attackers by reduction of payoff. Finally, the detailed experimental results show that the proposed scheme can reduce the loss of attacked intrusion detection system and then achieve the purpose of defending against the attacker.

## 1. Introduction

Internet of things (IoT) is entering people's lives and makes the production and life of human beings more intelligent and convenient. Smart Home is a typical application of the IoT [1]. Smart Home integrates integrated wiring technology and network communication technology and is an effective management system [2]. However, Smart Home is facing severe security threats such as data security and device security [3]. The distribution of edge devices is too scattered to apply security technologies in a Smart Home. Besides, some equipment uses outdated versions that are unable to remotely upgrade weaknesses and vulnerabilities, making Smart Home devices vulnerable to attacks. For instance, equipment such as cameras and smart thermostats collect information about people's daily lives which can be traced directly or indirectly back to the person. Once the data of Smart Home devices is stolen, users' private information will be disclosed. Therefore, it is urgent to design an effective security protection scheme to ensure user data security in the Smart Home.

Intrusion detection technology is a method to resist the attacker invasion, which can monitor, analyze, and deal with a variety of intrusions without affecting network performance as much as possible to improve the ability of networks to deal with external threats. According to the technology used, intrusion detection technology can be divided into three categories: anomaly detection, misuse intrusion detection, and hybrid intrusion detection. The abnormal detection technology can detect the new intrusion, but it is difficult to establish the attacker's behavior model [4]. Misuse detection technology has high detection accuracy, but it is difficult to collect and update intrusion information [5]. Hybrid intrusion detection technology combines misuse detection and anomaly detection, inherits the advantages of both, improves the detection rate, and decreases false positive rate [6]. To sum up, the existing intrusion detection technologies mainly have the following shortcomings: the volume of data is too difficult to process and the data dimension is too high to be reduced.

Inspired by the above schemes, this paper models interactions between attackers and intrusion detection systems

as the repeated game and proposes an intrusion detection scheme based on repeated game to protect the security of Smart Home. The main contributions are as follows:

(1) To reduce the cost of equipping the intrusion detection system, this paper uses the K-Nearest Neighbors (KNN) algorithm to classify edge devices and equips the intrusion detection system for cluster heads to achieve the purpose of protecting Smart Home system.

(2) To defend against attackers, we build interactions between attackers and intrusion detection systems as a repeated game model, use the regret minimization algorithm to determine the mixed strategy Nash equilibrium of this game, and set the severe punishment mechanism to force the attacker to take good action.

(3) For the part of the simulation experiment, we compare the proposed scheme with Winner, ALL-S, ALL-P, and ALL-R with three factors: the intrusion detection rate, the attacker's payoff, and the intrusion detection system's payoff. The experimental results show that the proposed scheme can resist attackers.

The remainder of this paper is organized as follows: Section 2 describes the representative achievements of intrusion detection technology. We propose an intrusion detection scheme based on repeated game in Smart Home in Section 3. Section 4 shows the performance of intrusion detection scheme based on repeated game. Finally, Section 5 summarizes the possible expansion and research directions in the future.

## 2. Related Work

Intrusion detection technology [7] can be divided into three types: anomaly detection, misuse detection, and hybrid intrusion detection. This section mainly summarizes two kinds of techniques of anomaly detection and misuse detection.

The anomaly intrusion detection [8] takes the intrusion activity as a subset of the anomaly activity, which is divided into feature selection-based anomaly detection, Bayesian inference-based anomaly detection, and pattern prediction-based anomaly detection. The feature selection-based anomaly detection is to accurately predict or classify detected intrusions by selecting a subset of metrics that can detect intrusions [9, 10]. However, the metric set cannot encompass all the various intrusion types; and the preidentified specific metric set may miss intrusions in a particular environment alone. The Bayesian inference-based anomaly detection is to judge whether the system has an intrusion event by measuring the variable [11, 12]. However, this method requires correlation analysis of each variable for determining the relationship between each variable and the intrusion event. The pattern prediction-based anomaly detection considers the sequence of intrusion events and their correlation [13, 14], but the unrecognized behavior pattern is judged as an abnormal event in this method.

Misuse intrusion detection [15, 16] detects intrusion events by matching the defined intrusion pattern with the observed intrusion behavior, which can be divided into contingent probability-based misuse intrusion detection, state transition analysis-based misuse intrusion detection, and keyboard monitoring-based misuse intrusion detection. The contingent probability-based misuse intrusion detection maps the intrusion to an event sequence and then infers the intrusion occurrence by observing the event [17, 18]. However, in this method, the prior probability is hard to give, and the event independences are hard to be satisfied. The state transition analysis-based misuse intrusion detection regards an attack as a series of state transitions of monitored systems [19, 20]. However, the attack mode can only describe the sequence of events and is not suitable for describing complicated events. The keyboard monitoring-based misuse intrusion detection assumes that the intrusion corresponds to a specific keystroke sequence pattern and then monitors the user keystroke pattern and matches this pattern with the intrusion pattern to detect intrusion [21, 22]. But this approach, without operating system support, lacks a reliable way to capture users' keystrokes, and users can easily cheat the technique by using alias commands.

To solve the above problems, we no longer detect the intrusion based on the characteristics of the attacker but consider intrusion detection system's payoff; that is, the intrusion detection system detects the attacker invasion by observing its payoff decrease.

## 3. Intrusion Detection Scheme Based on Repeated Game

This section describes how the intrusion detection system detects the attacker's malicious action and how to educate the malicious attackers to take good strategy. The notations definitions are shown in Table 1.

*3.1. One-Order Game.* In Smart Home, due to a large number of edge devices and limited service capacity [23, 24], it is impossible to run the intrusion detection system on each edge device, so we need to design a strategy to allocate the intrusion detection system on the edge device. We first use the clustering algorithm to divide edge devices into multiple clusters and then configure intrusion detection system for each cluster-head node in Smart Home [25, 26]. Each cluster has a cluster-head node and several member nodes. The former is mainly responsible for information forwarding and executing the intrusion detection program within the cluster, and the latter is responsible for collecting information and passing the information to the cluster-head node [27, 28]. Suppose that there are $N$ edge devices, which are divided into $k$ clusters by KNN algorithm, $C_1, C_2, \ldots, C_k$. We assume that an attacker can attack one cluster head at a time and model interactions between the intrusion detection systems and attackers as a one-order game model. That is,

Table 1: Notations definitions.

| Notations | Definition |
|---|---|
| $C_i$ | The $i$th cluster head |
| $S$ | Attackers and intrusion detection systems' strategy space |
| $c_i$ | The cost of attacking cluster heads $C_i$ |
| $c_i'$ | The cost of attacking cluster heads $C_i$ after $T$ times |
| $r_i$ | The cost of persistently protecting cluster heads $C_i$ |
| $r_i'$ | The cost of protecting cluster heads $C_i$ after $T$ times |
| $p_a^i$ | The payoff of attacking cluster heads $C_i$ |
| $p^{di}$ | The payoff of intrusion detection systems against attacks |
| $M$ | The strategy matrices of attacker and intrusion detection system |
| $X$ | Attackers' payoff matrix |
| $Y$ | Intrusion detection systems' payoff matrix |
| $U_e$ | The cumulative payoff of player $e$ |
| $\delta$ | The discount factor which measures how much players value future payoffs |

$$G_{\text{one-order}} = (P, S, U), \tag{1}$$

where $P$ is the player in one-order game, that is, the intrusion detection system and the attacker, $P = (a, d)$. $S$ is the strategy space, $S = (A_a, D_d)$, and $U$ is the player's payoff. The attacker has four strategies, $A_a = (a_1, a_2, a_3, a_4)$. $a_1$ refers to the fact that attackers do not attack any cluster heads; $a_2$ refers to the fact that attackers attack the cluster-head node $C_i$; $a_3$ refers to the fact that attackers attack cluster heads $C_i$ after $T$ times; $a_4$ refers to the fact that attackers attack the cluster-head node $C_j$. Also, the intrusion detection system has four strategies, $D_d = (d_1, d_2, d_3, d_4)$. $d_1$ refers to the fact that intrusion detection systems do not protect any cluster heads; $d_2$ refers to the fact that intrusion detection systems protect the cluster head $C_i$; $d_3$ refers to the fact that intrusion detection systems protect cluster heads $C_i$ after $T$ times; $d_4$ refers to the fact that intrusion detection systems protect the cluster head $C_j$. Therefore, the strategy profile of attacker and intrusion detection system can be defined as

$$M = \begin{bmatrix} (a_1, d_1) & (a_1, d_2) & (a_1, d_3) & (a_1, d_4) \\ (a_2, d_1) & (a_2, d_2) & (a_2, d_3) & (a_2, d_4) \\ (a_3, d_1) & (a_3, d_2) & (a_3, d_3) & (a_3, d_4) \\ (a_4, d_1) & (a_4, d_2) & (a_4, d_3) & (a_4, d_4) \end{bmatrix}. \tag{2}$$

The row represents the attacker's strategy and the column represents the intrusion detection system's strategy in $M$. Suppose that $U_a$ and $U_d$ are the payoffs of attackers and intrusion detection systems, respectively. Thus,

$$G = (a, d, A_a, D_d, U_a, U_d), \tag{3}$$

where $a$ refers to the attacker and $d$ refers to the intrusion detection system. The strategy profile $M_{22} = (a_2, d_2)$ refers to the fact that the attacker does not attack the cluster head, whereas the intrusion detection system protects the cluster head. At this time, the attacker gains the payoff 0 at the cost of $c_i$, $U_a = -c_i$, and the intrusion detection system at the cost of $r_i$ to gain the payoff $p_i$, $U_d = p_i - r_i$. Similarly, we can get the payoff matrix of attackers and intrusion detection systems, as shown in $X$ and $Y$:

$$\begin{aligned} X &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ p_a^i - c_i & -c_i & p_a^{i'} - c_i & p_a^i - c_i \\ p_a^{i'} - c_i' & -c_i' & p_a^{i'} - c_i' & p_a^{i'} - c_i' \\ p_a^j - c_j & p_a^j - c_j & p_a^j - c_j & -c_j \end{bmatrix}, \\ Y &= \begin{bmatrix} 0 & -r_i & -r_i' & -r_j \\ -p_a^i & p_a^i - r_i & p_d^{i'} - r_i' & -r_j \\ -p_a^{i'} & p_d^{i'} - r_i & p_d^{i'} - r_i' & -r_j \\ -p_a^j & -r_i & -r_i' & p_d^j - r_j \end{bmatrix}, \end{aligned} \tag{4}$$

where $c_i$ is the cost of attacking cluster heads $C_i$, $c_i'$ is the cost of attacking cluster heads $C_i$ after $T$ times, $r_i$ is the cost of persistently protecting cluster heads $C_i$, $r_i'$ is the cost of protecting cluster heads $C_i$ after $T$ times, $p_a^i$ is the payoff of attacking cluster heads $C_i$, and $p_d^i$ is the payoff of intrusion detection systems against attacks. It can be seen from the payoff matrix that there is no pure strategy Nash equilibrium in this game, and the intrusion detection system can observe malicious attackers according to its payoff decrease. Besides, the intrusion detection system always tries to determine the cluster head attacked by the attacker and then protect it to maximize its payoff. Therefore, we use the regret minimization algorithm that determines the selection method of that future action according to the degree of regret to determine the players' mixed strategy Nash equilibrium. Thus, the probability of playing strategy $d_1$ in round $T$ is defined as follows:

$$p(a) = \frac{\text{Regret}_d^T(d_1)}{\sum_{i \in D_d} \text{Regret}_d^T(d_i)}, \tag{5}$$

where $D_d$ is the intrusion detection system's strategy set, $\text{Regret}_d^T(d_1)$ is the regret value of playing strategy $d_1$, and

$\sum_{i\in D_d}\text{Regret}_d^T(d_i)$ is the cumulative regret value for all strategies.

*3.2. Repeated Game.* During the process of interaction between the attacker and intrusion detection system, the intrusion detection system can detect attackers' invasion by observing the changes of their payoff. However, the attacker does not have the effect of his current strategy on the future payoff, that is, he only considers the payoff of one interaction; therefore, it is difficult to prevent the attacker in the one-order game. But if the intrusion detection system punishes the attacker, the attacker will have to consider the cost of the penalty brought by the intrusion detection system in the repeated game; and if the punishment cost of attacking exceeds the payoff of attacking, the attacker will be forced to take a nonattack strategy. Thus, the intrusion detection system does not need to implement supervision and then achieve the purpose of maintaining the normal order of the entire network.

In the repeated game, assuming that $a_{et}$ is the strategy adopted by player $e$ in the $t$th round, the strategy set of player $e$ in the previous $T$ round is $a_{e1}, a_{e2}, \ldots, a_{eT}$. The total payoff of player $e$ can be expressed as

$$U_e = \sum_{t=1}^{T} \delta^{t-1} u\left(a_{et}, a_{-et}\right), \tag{6}$$

where $\delta$ is the discount factor, $\delta \in (0,1)$. The bigger $\delta$ is, the more $e$ pays attention to long-term payoff; and the smaller $\delta$ is, the more player $e$ pays attention to current payoff. Since the intrusion detection system cannot detect the attacker for the first time, we assume that the detection rate of the intrusion detection system to the attacker is less than 1, $q \in (0,1)$. The probability of an attacker being discovered by an intrusion detection system after $k$ times of attack is $(1-q)^{k-1}q$. The total payoff of the attacker is

$$U_a = \sum_{t=0}^{k} (1-q)^t \delta^t \left(p_a^i - c_i\right). \tag{7}$$

In previous researches on network security protection, once an attacker is captured by the intrusion detection system, the network will delete this node. However, it will affect the whole network and will have no containment effect on the attacker's action. Therefore, this paper designs a severe punishment mechanism to educate captured attackers into regular players. When the attacker is found to be uncooperative at the time slot $k$, within $T$ penalty cycles, that is, $k+1, k+2, \ldots, k+T$, the attacker's payoff can be defined as

$$U_a^T = \sum_{i=1}^{T}\sum_{t=0}^{k} \frac{1}{k+i}(1-q)^t \delta^t \left(p_a^i - c_i\right). \tag{8}$$

If the node is detected during the second attack, the node will be punished with a period of $2T$, and the total payoff of the attacker in the penalty cycle is

$$U_a^{2T} = \sum_{i=1}^{2T}\sum_{t=0}^{k} \frac{1}{2(k+i)}(1-q)^t \delta^t \left(p_a^i - c_i\right). \tag{9}$$

The loss of attacker in penalty cycle is

$$\begin{aligned}\Delta U_a^T &= \sum_{t=0}^{T} (1-q)^t \delta^t \left(p_a^i - c_i\right) - \sum_{i=1}^{T}\sum_{t=0}^{k} \frac{1}{k+i}(1-q)^t \delta^t \left(p_a^i - c_i\right),\\ \Delta U_a^{2T} &= \sum_{t=0}^{2T} (1-q)^t \delta^t \left(p_a^i - c_i\right) - \sum_{i=1}^{2T}\sum_{t=0}^{k} \frac{1}{k+i}(1-q)^t \delta^t \left(p_a^i - c_i\right).\end{aligned} \tag{10}$$

We regard the loss of the attacker in the penalty cycle as an additional reward to the intrusion detection system. Therefore, the intrusion detection system's payoff can be defined as

$$U_d = \sum_{t=1}^{T} \delta^{t-1} u\left(a_{et}, a_{-et}\right) + \Delta U_a^T, \tag{11}$$

where $\Delta U_a^T$ is the loss of attackers in the penalty cycle.

By comparing the attacker's payoffs over the two penalty cycles, it can be seen that the attacker's payoffs decrease with increasing the number of betrayals. Besides, if the number of defections by an attacker exceeds the threshold of the intrusion detection system, the attacker will be eliminated; and the cluster-head node will no longer interact with the attacker.

## 4. Simulation Experiment

This paper uses Anaconda integrated development tool to verify the intrusion detection scheme based on repeated game. Firstly, we simulate the classification process of KNN algorithm and set four newly added nodes to prove its effectiveness. Secondly, we compare the payoffs of attackers and the intrusion detection systems in penalty cycles and regular interaction cycles to verify the effectiveness of the penalty mechanism. Thirdly, we determine the optimal strategy for each round of interaction between the attacker and intrusion detection system by using the regret minimization algorithm. Finally, we compare the proposed scheme with four interaction strategies, Winner (take the strategy of the winner), ALL-S (remain strategy Scissor), ALL-P (remain strategy Paper), and ALL-R (remain strategy Rock), to prove that the proposed scheme can improve the player's payoff. The experimental parameters are shown in Table 2.

*4.1. The Classification Results of KNN.* Figure 1 depicts the classification results of the KNN algorithm. Figure 1(a) shows the original distribution of edge device nodes. Figure 1(b) shows the classification results of the KNN algorithm, with each symbol representing a class of edge devices.

Figure 2 analyzes the results of the classification of the newly added nodes, with the newly added nodes marked in blue. For example, in Figure 2(a), the blue node (the newly added node) is classified as a first class.

Table 2: Parameter setting.

| Parameters | $p_a^i$ | $p_e$ | $c_i$ | $\delta$ | $\eta$ | $T$ | $q$ |
|---|---|---|---|---|---|---|---|
| Value | 5 | 3 | 3 | 0.7 | 1 | 5 | 0.6 |

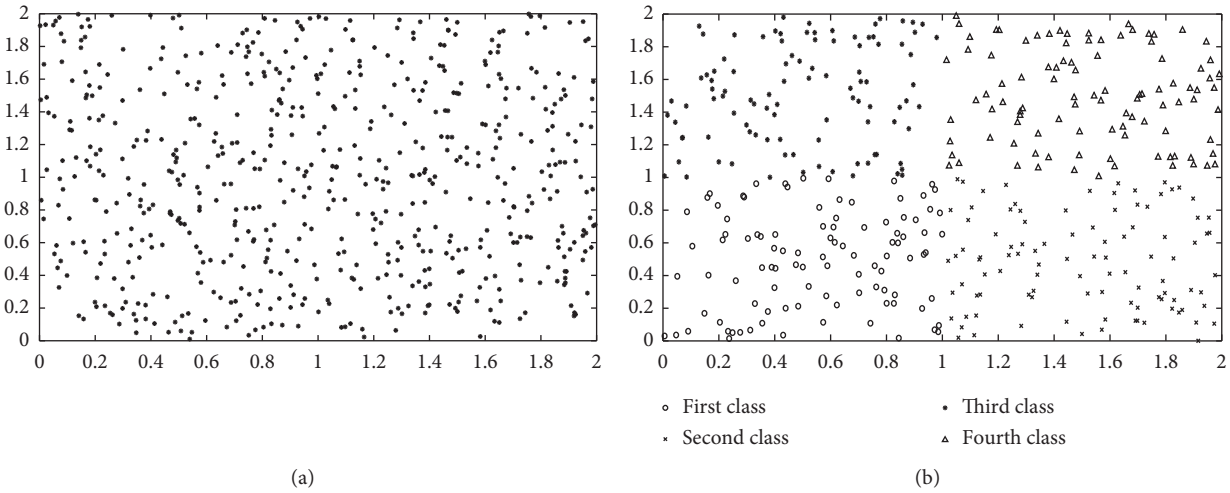



Figure 1: Comparison of classified data. (a) Raw data. (b) Classification results.

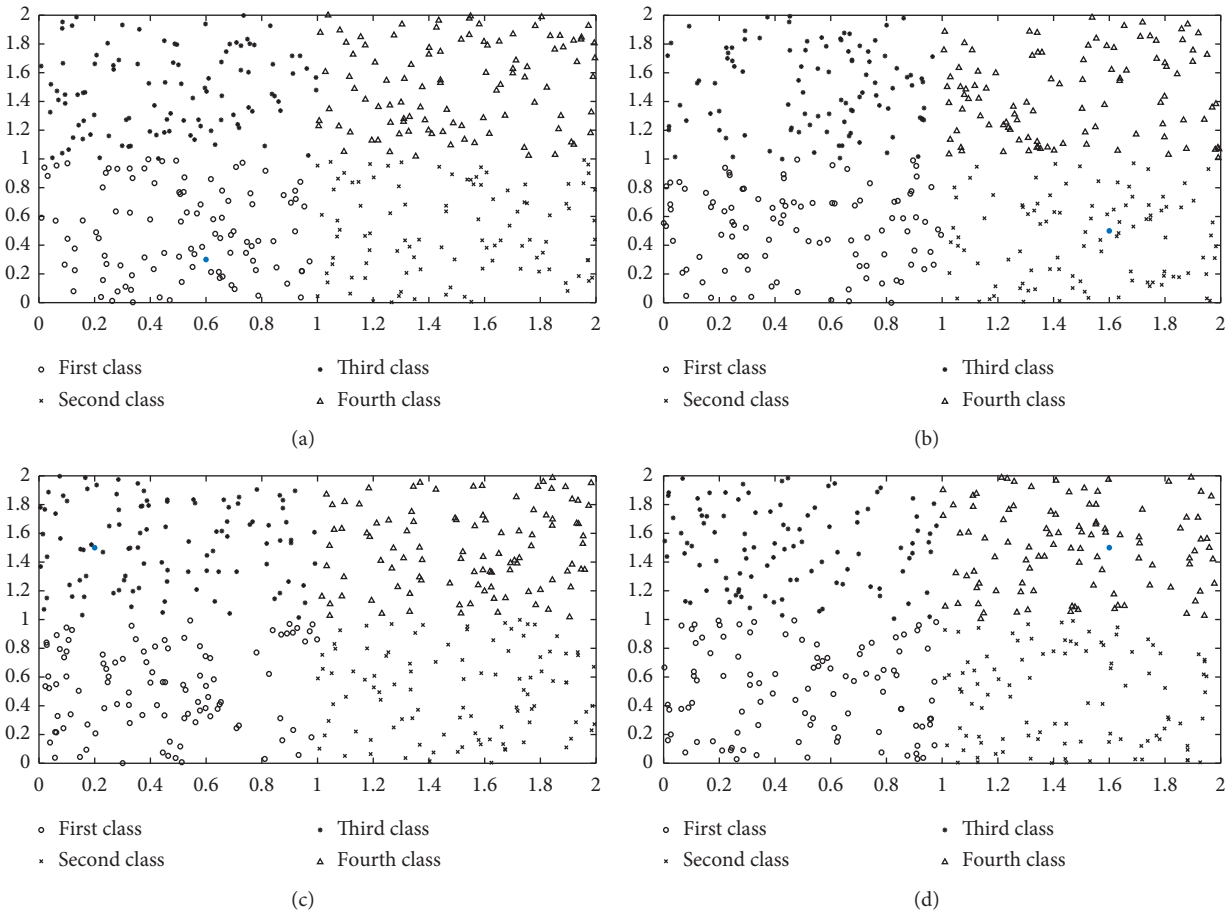



Figure 2: Classification of newly added data. (a) First class. (b) Second class. (c) Third class. (d) Fourth class.

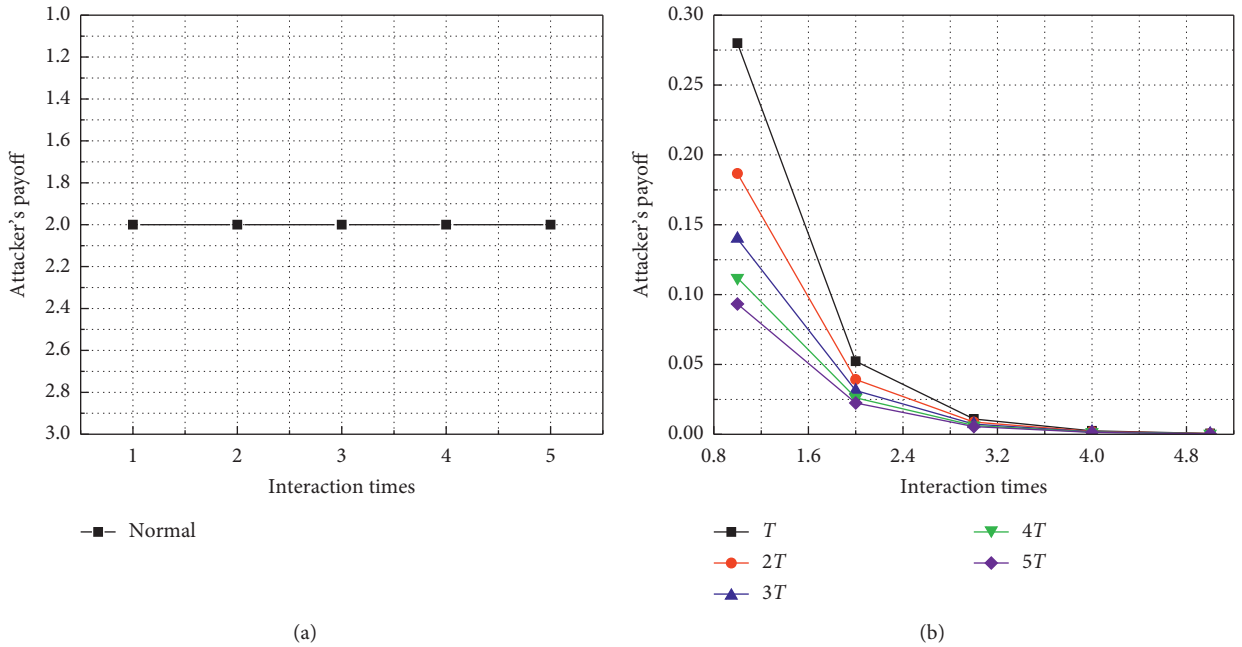



Figure 3: The attacker's payoff comparison. (a) The payoffs of regular interactions. (b) Payoff during the penalty period.

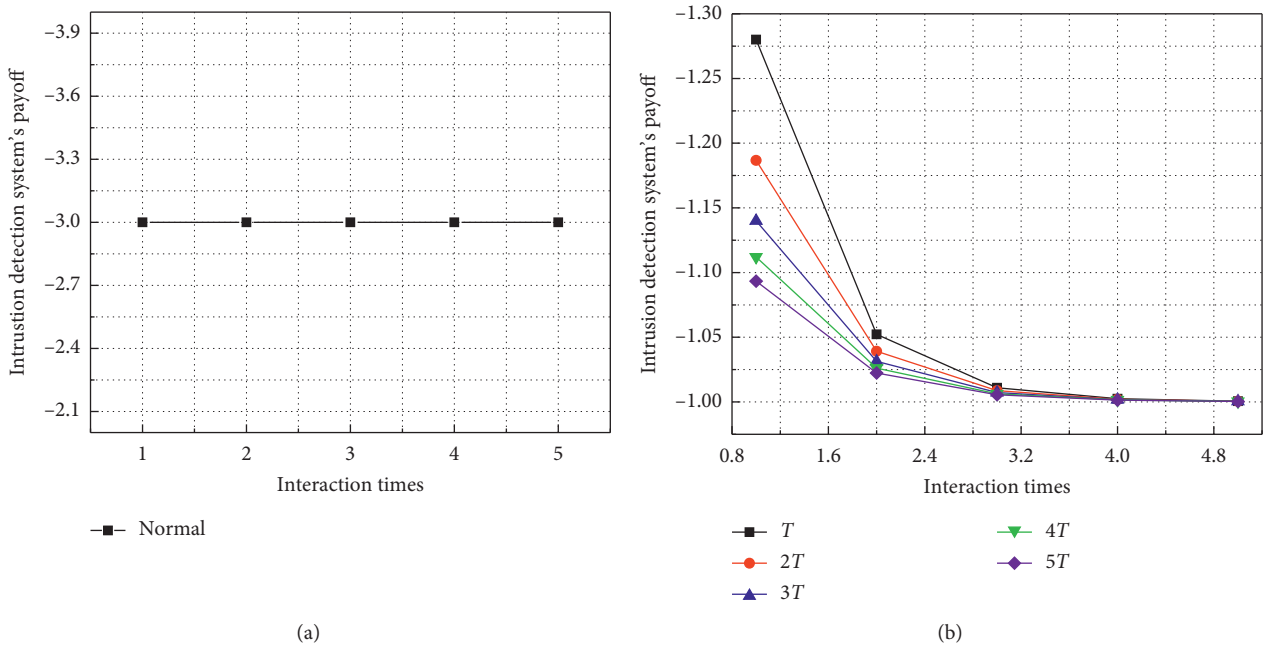



Figure 4: The intrusion detection system's payoff comparison. (a) The payoffs of regular interactions. (b) Payoff during the penalty period.

*4.2. The Comparison of the Attacker's Payoff and Intrusion Detection System's Payoff.* Figure 3 compares the attackers' payoffs in regular interaction cycles and penalty cycles. As you can see in Figure 3(a), the attacker's payoff does not change during regular interaction cycles, because the intrusion detection system does not play the defensive strategy. Figure 3(b) shows that the attacker's payoff gradually decreased with increasing the number of interactions. In the 4th interaction, the attacker's payoff tends to zero. Besides, the longer the penalty cycle is, the faster the attacker's payoffs will go to zero, and the larger the losses will be. This happened due to the punishment mechanism in this paper. Therefore, for a rational attacker, it must normally interact with the intrusion detection system to maximize its payoff.

Figure 4 compares the intrusion detection system's payoffs in the regular interaction cycle and the penalty cycle. It can be seen from Figure 4(a) that the intrusion detection system's payoff is −3 during the regular interaction cycle.

Table 3: Payoff matrix.

| Player A\B | Scissor | Rock | Paper |
|---|---|---|---|
| Scissor | 0, 0 | −1, 1 | 1, −1 |
| Rock | 1, −1 | 0, 0 | −1, 1 |
| Paper | −1, 1 | 1, −1 | 0, 0 |

Table 4: Regret value of player A.

| Iteration number | Player A | | | Optimal strategy |
|---|---|---|---|---|
| | Rock | Scissor | Paper | |
| 1 | 0 | 2 | 1 | (0, 2/3, 1/3) |
| 2 | 1 | 0 | 2 | (1/6, 2/6, 3/6) |
| 3 | 2 | 1 | 0 | (1/3, 1/3, 1/3) |
| 4 | 0 | 2 | 1 | (3/12, 5/12, 4/12) |
| 5 | 1 | 0 | 2 | (4/15, 5/15, 6/15) |
| 6 | 2 | 1 | 0 | (1/3, 1/3, 1/3) |
| 7 | 0 | 2 | 1 | (6/21, 8/21, 7/21) |
| 8 | 1 | 0 | 2 | (7/24, 8/24, 9/24) |
| 9 | 2 | 1 | 0 | (1/3, 1/3, 1/3) |
| 10 | 0 | 2 | 1 | (9/30, 11/30, 10/30) |
| Cumulative regret | 9 | 11 | 10 | — |

Table 5: Players' payoff comparison.

| Number | Payoff | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | A | B | A | B | A | B | A | B |
| 1 | −1 | 1 | −1 | 1 | −1 | 1 | 0 | 0 | 1 | −1 |
| 2 | 1 | −1 | −1 | 1 | 0 | 0 | 0 | 0 | 1 | −1 |
| 3 | −1 | 1 | 0 | 0 | −1 | 1 | 1 | −1 | −1 | 1 |
| 4 | 1 | −1 | 0 | 0 | 0 | 0 | 1 | −1 | −1 | 1 |
| 5 | −1 | 1 | −1 | 1 | −1 | 1 | −1 | 1 | 0 | 0 |
| 6 | 1 | −1 | −1 | 1 | 0 | 0 | −1 | 1 | 0 | 0 |
| 7 | −1 | 1 | 0 | 0 | −1 | 1 | 0 | 0 | 1 | −1 |
| 8 | 1 | −1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | −1 |
| 9 | −1 | 1 | −1 | 1 | −1 | 1 | 1 | −1 | −1 | 1 |
| 10 | 1 | −1 | −1 | 1 | 0 | 0 | 1 | −1 | −1 | 1 |
| Total | 0 | 0 | −6 | 6 | −5 | 5 | 2 | −2 | 0 | 0 |

This is because the attacked intrusion detection system does not play any defective strategy. Figure 4(b) shows that the loss of the intrusion detection system decreases with increasing the number of penalty cycles; and the payoff of the intrusion detection system is the lowest when the penalty period is 5. To sum up, the proposed scheme can reduce the loss of intrusion detection systems when attackers launch attacks.

*4.3. Application of Regret Minimization Algorithm in Rock-Paper-Scissors Game.* Table 3 defines the payoff matrix of two players in the rock-paper-scissors game. In this table, the rows represent the strategy of player A, the columns represent the strategy of player B, the first element in the tuple (0, 0) represents the payoff of player A, and the second element represents the payoff of player B.

Table 4 analyzes how player *A* determines its optimal strategy based on the regret minimization algorithm. For example, in the first round, player A and player B choose Rock and Paper, respectively, and then player A's regret values when playing Scissor, Rock, and Paper are 0, 2, and 1, respectively; thus the probabilities of player playing Rock, Scissor, and Paper are 0, 2/3, and 1/3, respectively. Similarly, we can obtain the optimal strategy of player A in each round.

*4.4. The Payoff Comparison between Player A and Player B.* Table 5 compares the payoffs of player A and player B when player A adopts five strategies: regret minimization strategy (Regret), ALL-R, ALL-P, ALL-S, and Winner, while player *B* adopts a regret minimization strategy. As can be seen from Table 5, when and only if player A adopts ALL-P, player B adopts Regret to obtain a lower payoff than player A, but the difference in payoff between player A and player B is small. However, under several other strategies, player B obtains the highest payoff by taking Regret. This is because player B maximizes the probability of the strategy with the maximum regret value. The payoff change curves of players A and B are shown in Figure 5. In this figure, the sharp increase and

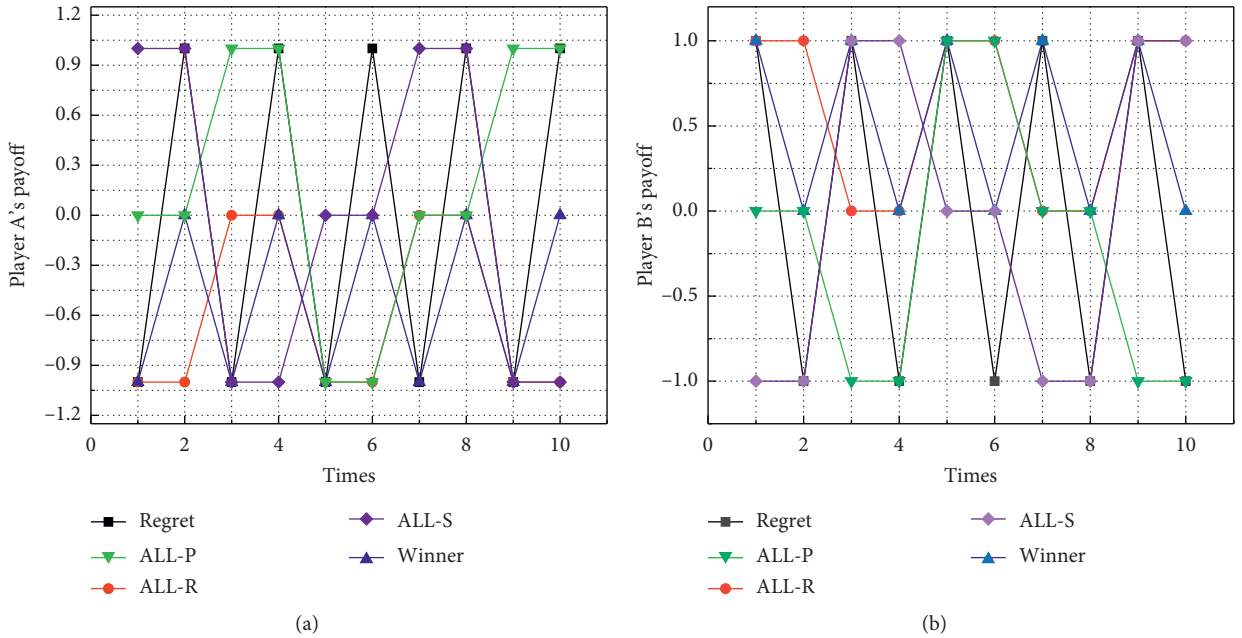



Figure 5: Players' payoff comparison. (a) The payoff of player A. (b) The payoff of player B.

decrease in the payoffs of player A and player B are due to the adjustment of both players' strategies.

## 5. Conclusion

Designing an efficient and safe protection scheme is the key to promoting the application of the system. This paper proposes a security protection scheme based on repeated game. In this scheme, the intrusion detection system detects the malicious attackers by observing its payoff change and punishes the attackers who adopt malicious strategy severely to educate the attackers to take good action. The experimental results show that the proposed scheme can effectively defend against the attackers.

In future research studies, we will continue to explore new methods to determine the player's optimal strategy in the finite model.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (NSFC) under Grant no. 61872205, the Shandong Provincial Natural Science Foundation under Grant no. ZR2019MF018, and the Source Innovation Program of Qingdao under Grant no. 18-2-2-56-jch.

## References

[1] N. Chen, T. Qiu, X. Zhou, K. Li, and M. Atiquzzaman, "An intelligent robust networking mechanism for the internet of things," *IEEE Communications Magazine*, vol. 57, no. 11, pp. 91–95, 2019.

[2] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart internet of things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[3] H. Xia, L. Li, X. Cheng, C. Liu, and T. Qiu, "A dynamic virus propagation model based on social attributes in city IoTs," *IEEE Internet of Things Journal*, 2020.

[4] M. A. Hatef, V. Shaker, M. Reza Jabbarpour, J. Jung, and H. Zarrabi, "HIDCC: a hybrid intrusion detection approach in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 3, p. e4171, 2018.

[5] S. Venkatraman and B. Surendiran, "Adaptive hybrid intrusion detection system for crowd sourced multimedia internet of things systems," *Multimedia Tools and Applications*, vol. 79, no. 5-6, pp. 3993–4010, 2020.

[6] K. K. R. Amrita, "A hybrid intrusion detection system: integrating hybrid feature selection approach with heterogeneous ensemble of intelligent classifiers," *International Journal of Network Security*, vol. 20, no. 1, pp. 41–55, 2018.

[7] T. Qiu, J. Liu, W. Si, and D. O. Wu, "Robustness optimization scheme with multi-population Co-evolution for scale-free wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1028–1042, 2019.

[8] H. Xia, L. Li, X. Cheng, X. Cheng, and T. Qiu, "Modeling and analysis botnet propagation in social internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 8, 2020.

[9] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[10] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: a survey," 2019, https://arxiv.org/abs/1901.03407.

[11] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the 31st*

*IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6479–6488, Salt Lake City, UT, USA, June 2018.

[12] D. Kwon, H. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, vol. 22, pp. 1–13, 2017.

[13] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença, "Network anomaly detection system using genetic algorithm and fuzzy logic," *Expert Systems with Applications*, vol. 92, pp. 390–402, 2018.

[14] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "LoTAD: long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web*, vol. 21, no. 3, pp. 825–847, 2018.

[15] H. Xia, S.-S. Zhang, Y. Li, Z.-K. Pan, X. Peng, and X.-Z. Cheng, "An attack-resistant trust inference model for securing routing in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 7108–7120, 2019.

[16] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but no track: privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1868–1878, 2017.

[17] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "Specguard: spectrum misuse detection in dynamic spectrum access systems," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2925–2938, 2018.

[18] H. A. Seven, H. A. Nguyen, S. Nadi, T. N. Nguyen, and M. Mezini, "Investigating next steps in static API-misuse detection," in *Proceedings of the 16th International Conference on Mining Software Repositories*, pp. 265–275, Montreal, Canada, May 2019.

[19] S. Amann, H. A. Nguyen, S. Nadi, T. N. Nguyen, and M. Mezini, "A systematic evaluation of static API-misuse detectors," *IEEE Transactions on Software Engineering*, vol. 45, no. 12, pp. 1170–1188, 2018.

[20] T. Qiu, B. Li, X. Zhou, H. Song, I. Lee, and J. Lloret, "A novel shortcut addition algorithm with particle swarm for multisink internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3566–3577, 2020.

[21] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, 2018.

[22] H. Li, K. Ota, and M. Dong, "Deep reinforcement scheduling for mobile crowdsensing in fog computing," *ACM Transactions on Internet Technology*, vol. 19, no. 2, pp. 1–18, 2019.

[23] H. Zhang, J. Yu, C. Tian et al., "Efficient and secure outsourcing scheme for RSA decryption in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6868–6881, 2020.

[24] H. Zhang, J. Yu, C. Tian, G. Xu, P. Gao, and J. Lin, "Practical and secure outsourcing algorithms for solving quadratic congruences in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2968–2981, 2020.

[25] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: state of the art and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1608–1631, 2019.

[26] D. Yu, Y. Zou, J. Yu et al., "Implementing abstract MAC layer in dynamic networks," *IEEE Transactions on Mobile Computing*, 2020.

[27] D. Yu, Y. Zou, J. Yu et al., "Stable local broadcast in multihop wireless networks under SINR," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1278–1291, 2018.

[28] F. Li, D. Yu, H. Yang, J. Yu, H. Karl, and X. Cheng, "Multi-armed-bandit-based spectrum scheduling algorithms in wireless networks: a survey," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 24–30, 2020.

Hindawi
Mobile Information Systems
Volume 2020, Article ID 8861947, 15 pages
https://doi.org/10.1155/2020/8861947

*Research Article*

# Multiaccess Edge Computing Empowered Flying Ad Hoc Networks with Secure Deployment Using Identity-Based Generalized Signcryption

**Muhammad Asghar Khan**,[1] **Insaf Ullah**,[2] **Shibli Nisar**,[3] **Fazal Noor**,[4] **Ijaz Mansoor Qureshi**,[5] **Fahimullah Khanzada**,[6] **Hizbullah Khattak**,[2] **and Muhammad Adnan Aziz**[7]

[1]*Hamdard Institute of Engineering & Technology, Islamabad 44000, Pakistan*
[2]*Department of Information Technology, Hazara University, Mansehra, Pakistan*
[3]*Department of Electrical Engineering, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan*
[4]*Department of Computer Science and Information Systems, Islamic University of Madinah, Madinah 400411, Saudi Arabia*
[5]*Department of Electrical Engineering, Air University, Islamabad 44000, Pakistan*
[6]*Descon Engineering Limited, Lahore, Pakistan*
[7]*Department of Electronic Engineering, ISRA University, Islamabad 44000, Pakistan*

Correspondence should be addressed to Muhammad Asghar Khan; khayyam2302@gmail.com

Received 20 April 2020; Revised 19 May 2020; Accepted 2 June 2020; Published 1 July 2020

Academic Editor: Vishal Sharma



A group of small UAVs can synergize to form a flying ad hoc network (FANET). The small UAVs are, typically, prone to security lapses because of limited onboard power, restricted computing ability, insufficient bandwidth, etc. Such limitations hinder the applicability of standard cryptographic techniques. Thus, assuring confidentiality and authentication on part of small UAV remains a far-fetched goal. We aim to address such an issue by proposing an identity-based generalized signcryption scheme. The lightweight security scheme employs multiaccess edge computing (MEC) whereby the primary UAV, as a MEC node, provides offloading to the computationally fragile member UAVs. The scheme is based on the concept of the hyperelliptic curve (HEC), which is characterized by a smaller key size and is, therefore, suitable for small UAVs. The scheme is robust since it offers confidentiality and authentication simultaneously as well as singly. Formal as well as informal security analyses and the validation results, using the Automated Validation for Internet Security Validation and Application (AVISPA) tool, second such notion. Comparative analysis with the existing schemes further authenticates the sturdiness of the proposed scheme. As a case study, the scheme is applied for monitoring crops in an agricultural field. It has been found out that the scheme promises higher security and incurs lower computational and communication costs.

## 1. Introduction

Unmanned Aerial Vehicles (UAVs) have earned recognition in multiple domains owing to their versatile applications for surveillance, agriculture, health services, traffic monitoring, inspection, public safety, etc. [1]. Multiple small UAVs, as a flying ad hoc network (FANET), can combine and accomplish the assigned tasks efficiently in an autonomous manner [2, 3]. In FANETs, small interconnected UAVs synergize and exchange data with one another and with the ground stations [4]. They are characterized by high mobility, easy deployment, and self-organizing behavior [5]. However, such distinctive features, for efficient and effective deployment, demand the compliance of stringent guidelines [6]. For instance, it is mandatory to assure security and Quality of Service (QoS) when choosing a FANET system for on-time data communication services. Moreover, the networks must deploy an efficient networking architecture complemented by an efficient security scheme in order to allow a reliable exchange of information between UAVs and the ground stations.

FANETs can either be deployed independently or they can be integrated with the traditional networks via satellite or cellular communication links. The topic allures experts from the industry as well as academia. Most of the relevant research studies propose to integrate multiple-UAV systems with the traditional networks to assure Quality of Service (QoS), unhampered security, and sustained reliability. Therefore, it is imperative to identify loopholes in existing solutions. This can pave the way for solutions that support high throughput and a secure data communication regime. The envisioned Fifth Generation (5G) of wireless cellular communication systems is expected to offer higher capacity, enhanced data rate, and lower latency [7]. Besides, 5G offers multiaccess edge computing (MEC) architecture, which is characterized by cloud computing functionalities. Thus, 5G, when integrated into a UAV environment, by leveraging MEC, can relieve the resource-constrained UAVs from processing the computational tasks. Instead, the computationally intensive tasks will be offloaded to the edge of the network.

Generally, the small UAVs are not designed with security considerations and are, therefore, prone to security and privacy pitfalls [8]. UAV's sensing portion is also worth consideration. For instance, in the worst case, a sensor might transmit wrong information and that can result in UAVs making erroneous decisions. Similarly, the case of the faulty sensor is far more sinister. A damaged sensor can severely hamper the UAV's attempt to obtain information and might result in an event of a crash. Furthermore, a strong communication link is essential to allow the exchange of information between a UAV and a Base Station. An insecure and vulnerable link, on the other hand, is susceptible to attacks [9]. The concerns of confidentiality and authentication can be addressed by employing encryption and digital signature, respectively. And, in case both the attributes are desired, a hybrid version, the sign-then-encrypt approach, is utilized mostly.

However, the stringent constraints associated with a flying ad hoc network (FANET), such as limited onboard energy and limited computing capability, do not permit complex cryptographic operations. Moreover, undertaking computationally intensive tasks may result in slow response time which can, in turn, deteriorate the performance of FANETs. Fortunately, such deficiencies can be resolved by employing an amalgamated scheme, named "signcryption" [10]. It is a public key cryptosystem that performs the function of encryption and digital signature simultaneously. It is far more efficient and cost-effective than each of the alternates, i.e., encryption and digital signature. To simplify the key management process and to allow flexibility, Han et al. [11] presented an extension of the signcryption scheme, i.e., generalized signcryption (GSC). Not only does GSC offer encryption and digital signature in one go, but it also has the option to offer them separately, if demanded. Such feature is helpful in case either of the two key attributes, confidentiality or authenticity, is required.

In the public key cryptosystems, two basic approaches, Public Key Infrastructure (PKI) and Identity-Based Cryptography (IBC), are used to authenticate public keys [12]. In the PKI environment, it is crucial to ensure a trustworthy unforgeable link between the identity of the participant and its public key. This further stipulates the need for a signature Certificate Authority (CA) that assigns the link a unique signature. In the certification stage, the CA bounds the public key as the identity of a participant with certificates. The Public Key Infrastructure (PKI) approach encounters issues with certificate distribution and storage. On the other hand, an identity-based cryptosystem is used to reduce the cost of public key management [13]. In ID-based systems, a trusted third party named private key generator (PKG) computes private keys from a master secret and users' identity information. It then distributes these private keys to the users participating in the scheme. This eradicates the necessity for certificates as used in a conventional PKI.

The security and efficiency of the aforementioned security schemes are based on computationally hard problems. The RSA cryptography [14, 15] is based on a large factorization problem, which utilizes a large key, parameter certificate, and the identity stretches as much as 1024 bits [16]. This is not suitable for resource-constrained networks, or FANETs, because small UAVs lack onboard processing resources. Furthermore, bilinear pairing is 14.31 times worse than RSA [17], due to huge pairing and map-to-point function computation. In order to eliminate the discrepancies accompanying RSA and bilinear pairing, a new type of cryptography called the elliptic curve was introduced [18]. The elliptic curve cryptography is characterized by smaller parameter size, smaller public/private key size, smaller identity, and smaller certificate size. Moreover, unlike bilinear pairing and RSA, the security hardiness and efficiency of the elliptic curve cryptography scheme are based on 160-bit small keys [19]. The 160-bit key is, still, not suitable for and affordable by resource-hungry devices such as small UAVs. Thus, the hyperelliptic curve, a more modern version of the elliptic curve cryptography, was proposed [20]. The hyperelliptic curve uses an 80-bit key, identity, and certificate size and, at the same time, promises the security features assured by the elliptic curve, bilinear pairing, and RSA [21, 22]. Therefore, the hyperelliptic curve is a cogent choice for energy-constrained devices.

*1.1. Authors' Motivation and Contributions.* To reap the extensive benefits of multi-UAV systems, the underlying technical challenges need to be addressed. For instance, the small UAVs have limited onboard energy, which restricts the flying time to a specified period and the UAV's limited computational capability does not permit complex cryptographic operations. Therefore, there is a need to harness a state-of-the-art communication architecture with a lightweight security mechanism, which can, significantly, stabilize the battery lifetime, offer limited computation cost, and provide better connectivity.

Motivated by such objectives, for FANETs, the authors, here, suggest an identity-based generalized signcryption scheme. The very scheme makes use of multiaccess edge computing (MEC) and is based on a much advanced version of the elliptic curve, i.e., the hyperelliptic curve (HEC). HEC

is characterized by a smaller key size and, at the same time, promises security comparable to that of the counterparts, i.e., elliptic curve, bilinear pairing, and modular exponentiation. Incorporation of HEC reduces power consumption and improves the device's performance, thereby making it suitable for a wide range of devices, ranging from sensors to UAVs.

Some of the salient features signifying the contribution of our research work, in this paper, are as follows:

(i) We introduce a new architecture for flying ad hoc networks (FANETs) leveraging multiaccess edge computing (MEC) facility, where the primary UAV acts as a MEC node in order to provide computational offloading services for the member UAVs having limited local computing capabilities

(ii) We propose an efficient and provably secure identity-based generalized signcryption scheme for the architecture using the concept of a hyperelliptic curve

(iii) The proposed scheme is potent enough to thwart attacks, both known and unknown, and the validation results using the Automated Validation for Internet Security Validation and Application (AVISPA) tool second such notion

(iv) Moreover, upon doing a comparative analysis with the extant schemes, it is revealed that our proposed scheme is superior, particularly, in terms of computational and communication costs

*1.2. Structure of the Paper.* The rest of the paper is organized as follows. In Section 2, we provide a brief about the related work. Foundational concepts of the research work are presented in Section 3. Section 4 is dedicated to present the two system models, i.e., network model and threat model. In Section 5, we explain the salient features of the proposed scheme. Informal security analysis is provided in Section 6. Section 7 presents the practical deployment of the proposed scheme. For performance evaluation, the proposed scheme is compared with the existing schemes in Section 8. Section 9 contains a brief about a case study in which the scheme is applied for precision agriculture. Finally, Section 10 concludes the work.

## 2. Related Work

*2.1. UAV-Enabled Multiaccess Edge Computing.* Owing to the promising features of on-demand communication services and flexible deployment, UAV-enabled multiaccess edge computing capabilities have received much attention in recent years. So far, various studies have been conducted to examine the usability of edge computing for UAVs [23, 24]. However, the studies do not address the topic of security. Garg et al. [25] aimed to answer the surveillance-related concerns by proposing a framework based on probabilistic data structures. The framework treats UAVs as intermediate aerial nodes that offer a cyberthreat detection mechanism complemented with a real-time analysis. Four major elements of the framework are as follows: UAV, dispatcher, aggregator, and edge devices. The UAV is responsible for capturing and validating the data. The processing tasks in the edge computing devices are scheduled by the dispatcher. The aggregator assures the secure transmission of data. And, the edge devices analyze the data.

In [26], the authors extend the concept of network slicing to the case of UAV-based 5G network deployment and investigate the feasibility of a backhaul of an aerial node utilizing a UAV. The LTE signals are monitored to evaluate the suitability of UAVs in two scenarios: network capacity enhancement and increasing network coverage.

The methodology proposed by Christian et al. [27] increases the system reliability and reduces the end-to-end source-actuator latency. Their work intends to broaden the 5G network edge by making the FANET UAVs fly close to the monitoring layer. For enhanced operations, the UAVs follow a policy of mutual help and are accoutered with MEC facilities. However, the work fails to address the issue of the limited battery duration of the MEC-UAVs. In [28], the authors proposed a UAV edge-cloud computing model that utilizes a UAV swarm to provide the users real-time support. The end data are stored in the cloud server. In [29], the authors presented an architectural design of a slice orchestrator that enables new application models where the Internet of Things related functions can be applied on small Unmanned Aerial Vehicles, thus paving the way for implementing these functions on the edge network.

*2.2. Security Mechanisms in Flying Ad Hoc Networks.* The primary security mechanisms for FANETs emphasize authenticity, confidentiality, and integrity of data via cryptography. A well-designed data protection mechanism can significantly reduce the probability of the data get compromised, irrespective of the devilish technique involved. There are a few studies dedicated to investigating the data protection issues for UAV Networks. In a secure communication scheme proposed by He et al. [30], the requirement of an online centralized authority is waived off. The UAVs manage the area themselves and the authorized devices can obtain a broadcast key. The scheme is characterized by employing hierarchical identity-based broadcast encryption and a pseudonym mechanism, whereby the devices can, anonymously, broadcast the encrypted messages and decrypt the legal ciphertext. The work done seconds the notion that the very scheme, satisfactorily, addresses the four important security concerns: confidentiality, authentication, partial privacy preservation, and resistance to Denial of Service (DoS) attacks. However, it inherits a restriction in the registration phase, i.e., the concern of finding a hash value's preimage persists.

Three communication scenarios have been described by Won et al. [31, 32] to propose cryptographic protocols for drones and smart objects. The first scenario, i.e., one-to-one, implies a certificateless signcryption tag key for facilitating an authenticated key agreement and for providing nonrepudiation and user revocation. One-to-many, or the second scenario, enables a UAV to broadcast privacy-

sensitive data to multiple smart objects using a certificateless multirecipient encryption scheme. The third scenario is termed "many-to-one" and is characterized by UAVs capable of collecting data from multiple smart objects. However, for such protocols [31, 32], transmitting encrypted messages and assuring privacy simultaneously are too difficult to undertake. Such novel cryptographic mechanisms are efficient and secure. However, they are supposed to be used in group communication where nodes are of equal computational capability. In 2019, Asghar et al. [33] proposed a blind signature scheme for flying ad hoc networks in a certificateless setting. The scheme is suitable for authentication; however, it does not offer confidentiality and authentication simultaneously.

*2.3. Identity-Based Generalized Signcryption Schemes.* Lal et al. [34], in 2008, introduced the first identity-based generalized signcryption scheme and proposed a security model for it. However, Yu et al. [13] pointed out that the security model presented by Lal et al. [34] scheme is incomplete and proposed a new scheme, which is efficient in terms of computation and is secure. Later, in 2011, Kushwah et al. [35] simplified the security model introduced by Yu et al. [13] and proposed a more efficient identity-based generalized signcryption scheme. Wei et al. [36], in 2015, presented an identity-based generalized signcryption scheme, which demonstrated to be secure enough in the random oracle model. Shen et al. [37], in 2017, proposed an identity-based generalized signcryption scheme in the standard model. Nevertheless, the proposed scheme is based on bilinear pairing that is computationally expensive. In 2019, Waheed et al. [38] analyzed the work done by Wei et al. [36] and suggested an improved scheme that is far more secure and cost-effective. Lastly, in 2019, Zhou et al. [39] proposed an identity-based combined public key scheme for signature, encryption, and signature (IBCSESC). Under the premise of ensuring the confidentiality, integrity, authentication, and nonrepudiation of data, the combined cryptosystem reduces the key management work, saves storage space, and offers decreased computational consumption.

## 3. Preliminaries

*3.1. Hyperelliptic Curve Cryptography (HECC).* HECC is the advanced form of elliptic curve cryptography (ECC), and it is used to exchange keys and facilitate secure communications between two parties with very small size keys and incur lower computational and communication costs. For instance, an encryption activity done using RSA with a 1024-bit key and ECC with a 160-bit key is equivalent in performance to HECC encryption with an 80-bit key [40].

Suppose that $\mathfrak{F}q$ is a predetermined set and presume $\partial$ as the genus of $h\varepsilon c$ having order as $\partial \geq 2$. Let $(v), f(v) \in \mathfrak{F}q\,[v]$, deg $(h(v)) \leq \partial$, and $f(v)$ is a monic-polynomial having deg $(f(v)) = 2\partial + 1$. Thus, $h\varepsilon c$ of genus $\partial \geq 2$ over $\mathfrak{F}q$ is set of points $(v,)$ $\mathfrak{F}q * \mathfrak{F}q$ as shown in

$$h\varepsilon c\colon w^2 + (v)w = f(v). \tag{1}$$

It forms the divisors which are the formal sum of finite integers like $d = \sum x_i z_i$ where $x_i \in \mathfrak{F}q$ and $z_i \in h\varepsilon c$. Further, it forms a Jacobian group $\mathfrak{F}_{h\varepsilon c}(\mathfrak{F}q)$ having the following order:

$$(\sqrt{t} - 1)^{2\partial} \leq \mathcal{J}_{h\varepsilon c}\mathfrak{F}_q \leq (\sqrt{t} + 1)^{2\partial}. \tag{2}$$

*3.2. Hyperelliptic Curve Discrete Logarithm Problem* ($h\varepsilon c - dl$P). Assume that $d$ is the divisor that is publicly available in the network and $\mathscr{L}$ is a randomly picked private number from $\mathfrak{F}_t$. Upon recovering $\mathscr{L}$ from $d_1 = d$, $\mathscr{L}$ is said to be ($h\varepsilon c - dl$P).

## 4. System Models

To elaborate on the operation and applicability of the proposed scheme, two models are used.

*4.1. Network Model.* We devise a novel architecture for a flying ad hoc network (FANET), constituted by UAVs, with a multiaccess edge computing (MEC) facility that makes use of the Fifth Generation (5G) wireless communication technology on backhaul and the Wi-Fi technology on fronthaul, as shown in Figure 1. The 5G and Wi-Fi wireless technologies are enabled on MEC-UAV in order to link it with the Macro Base Station (MBS) and to provide a hotspot service over the M-UAVs. The M-UAVs are connected with each other via a Wi-Fi link. The primary reason behind opting for such a hybridized approach is to utilize the prominent features of both technologies. This ends up in the resulting solution being of low cost, low power, high range, and high speed. A huge bandwidth is required when linking the Macro Base Stations with the core network. The proposed architecture involves the UAVs connected together via either of the two classes: monitoring UAV (M-UAV), responsible for performing the monitoring function from an assigned zone; and multiaccess edge computing UAV (MEC-UAV), utilizing MEC to handle a set of M-UAVs connected to it. It is the load generated by an M-UAV that acts as a decisive factor when assigning M-UAV(s) to a MEC-UAV, or the primary UAV. In the maneuver, each of the MEC-UAVs is equipped with Raspberry PI (RPI) powered with a 1.5 GHz 64-bit quad-core ARM Cortex-A72 processor [41].

*4.2. Threat Model.* The proposed scheme employs the Dolev-Yao (DY) threat model [42]. The model indicates that an untrustworthy nature prevails between the end-point entities and that there is an insecure open channel between the parties. Thus, for an attacker, it eases the task to eavesdrop and delete/modify the exchanged messages. Far worse is the scenario when a drone, while hovering over a hostile area, is physically captured and the data is compromised. Recently, the widely accepted "Canetti and Krawczyk's adversary model (CK-adversary model)" [43] becomes the "current de facto standard model in modeling authenticated key exchange protocols." According to the CK-adversary model,

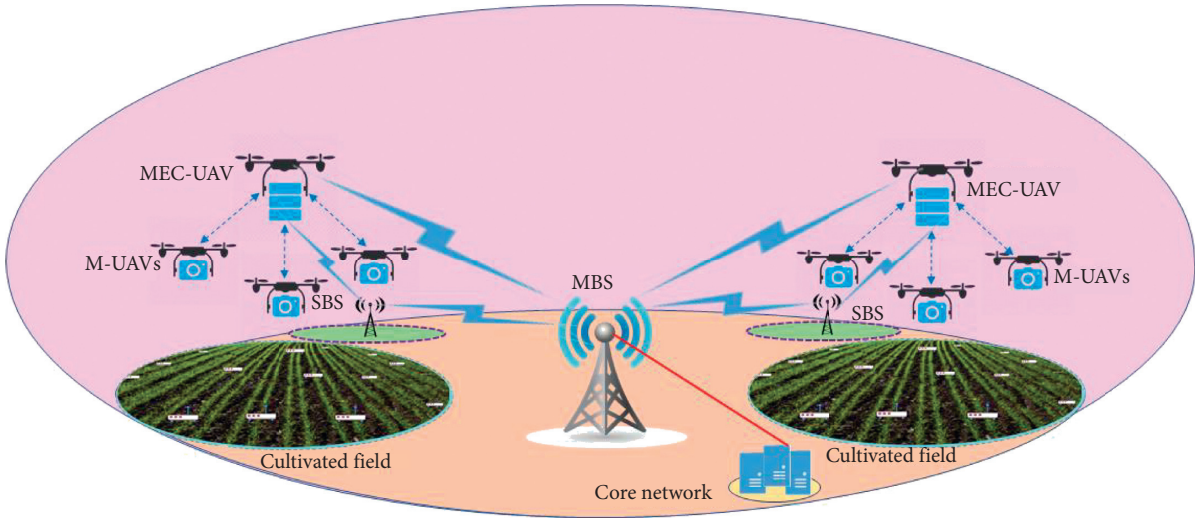



Figure 1: Multiaccess edge computing empowered FANET architecture of the proposed scheme when applied for monitoring.

"the adversary can not only deliver the messages (as in the DY model), but can compromise the secret credentials, secret keys and session states a well, particularly, when stored in the insecure memory." Therefore, it becomes an essential requirement that "the leakage of some forms of secret credentials, such as session ephemeral secrets or secret key, should minimally effect the secrecy of the communicating participants" [33].

## 5. Proposed Identity-Based Generalized Signcryption Scheme

*5.1. Syntax of Identity-Based Generalized Signcryption Scheme.* A formal model of identity-based generalized signcryption scheme consists of the following four algorithms [13, 37]: setup, key extraction, generalized signcryption, and generalized unsigncryption. The notations used in the proposed scheme are illustrated in Table 1.

(i) *Setup.* In the setup phase, the private key generation (PKG) generates the public parameters, randomly selects their master private key, and computes the master public key with the input of security parameter.

(ii) *Key Extraction.* When each of the participated contestants transmits their respective identities ($ID_{ps}$) to the PKG, PKG generates the private ($A_{pc}$) and public ($B_{pc}$) keys for each of them and delivers them using the private network.

(iii) *Generalized Signcryption.* The sender performs this process for producing generalized signcryption of a message ($m$). It initially takes the input parameter such as the identity of the sender and receiver ($ID_{cs}, ID_{cr}$), message ($m$), the private key of the sender ($A_{cs}$), the public key of the receiver ($B_{cr}$), and a fresh nonce ($n_{cs}$).

(iv) *Generalized Unsigncryption.* The receiver performs this process for recovering a message ($m$) and verifying generalized signcryption text $\psi$. It takes the input parameter like generalized signcryption text $\psi$, the identity of the sender and receiver ($ID_{cs}, ID_{cr}$), the private key of the receiver ($A_{cr}$), the public key of the receiver($B_{cr}$), and the public key of the sender ($B_{cs}$).

*5.2. Construction of the Proposed Identity-Based Generalized Signcryption Scheme.* It includes the following four subphases [13, 37]:

Setup: in this phase, the private key generation (PKG) center performs essential steps. It

(a) Selects a security parameter $\kappa$
(b) Selects a hyperelliptic curve (HEC) of genus 2
(c) Selects a parameter $q$ where the length is equivalents to 80 bits
(d) Selects a finite field $f_q$, where its order is $q$
(e) Selects a divisor $D$ of the order $q$
(f) Selects two one-way hash function, i.e., $h_a$ and $h_b$
(g) Selects a number uniformly for its private key as $\delta \in [1, 2, \ldots, (q-1)]$
(h) Computes its public key as $\Lambda = \delta.D$
(i) Produces all the public parameter param $E = [q, h_a, h_{b,} f_q, \kappa, \Lambda, HEC, D]$ and publish them to the network

Key extraction: when each of the participating contestants transmits their identity ($ID_{pc}$) to the PKG, the PKG generates the private and public keys by utilizing the performing the following computations:

(a) It computes private key for identity ($ID_{pc}$) as $A_{pc} = \delta \cdot h_a(ID_{pc}) \bmod q$
(b) It computes public key for identity ($ID_{pc}$) as $B_{pc} = A_{pc}.D$
(c) It delivers the pair of the public and private keys ($B_{pc}, A_{pc}$) to the participating contestants with its identity ($ID_{pc}$) by using the private network

Table 1: Notations used in the proposed algorithm.

| S.NO | Symbol | Definition |
|---|---|---|
| 1 | $h\varepsilon c$ | Hyperelliptic curve |
| 2 | $\kappa$ | Security parameter |
| 3 | PKG | Private key generation center |
| 4 | $q$ | A large prime number with length equivalents to 80 bits |
| 5 | $\mathfrak{F}q$ | A finite field of the order $q$ |
| 6 | $h_a$, $h_b$. | Hash functions |
| 7 | $\delta$ | Master private key of PKG |
| 8 | $\Lambda$ | Master public key of PKG |
| 9 | E | Public parameter param |
| 10 | $ID_{cs}$ | Identity sender |
| 11 | $ID_{cr}$ | Identity receiver |
| 12 | $A_{cs}$ | Private key of the sender |
| 13 | $A_{cr}$ | Private key of the receiver |
| 14 | $B_{cs}$ | Public key of the sender |
| 15 | $B_{cr}$ | Public key of receiver |
| 16 | $\eta$, $m$ | Ciphertext and plain text |
| 17 | $n_{cs}$ | A fresh nonce |
| 18 | $\beta$ | Encryption and decryption key |
| 19 | $e_\beta, d_\beta$ | Encryption and decryption through $\beta$ |
| 20 | $\psi = (\partial, \sigma, \eta, \Delta)$ | Generalized signcryption text for the receiver |
| 21 | // | Used for concatenation |
| 22 | $\perp\!\!\!\perp$ | Used for error |

Generalized signcryption: given a message (m), the private key of the sender ($A_{cs}$), the public key of the receiver ($B_{cr}$), the identity of the sender and receiver ($ID_{cs}, ID_{cr}$), and a fresh nonce ($n_{cs}$), the sender performs this process for producing generalized signcryption by undertaking the following steps

(a) It selects a number in an irregular manner as $\varphi \in [1, 2, \ldots, (q-1)]$ and calculates $\Delta = \varphi \cdot D$
(b) It calculates $\beta = \varphi \cdot B_{cr} \cdot ID_{cr}$
(c) It computes $\eta = e_\beta(m//ID_{cs}//ID_{cr}//n_{cs})$
(d) It calculates $\sigma = h_b(m//ID_{cs}//ID_{cr}//n_{cs})$
(e) It computes $\partial = (ID_{cr} \cdot \varphi - \sigma \cdot \Delta \cdot A_{cs} \cdot ID_{cs}) \bmod q$
(f) It produces the final generalized signcryption text for the receiver as $\psi = (\partial, \sigma, \eta, \Delta)$

Generalized unsigncryption: given a generalized signcryption text $\psi = (\partial, \sigma, \eta, \Delta)$, the private key of the receiver ($A_{cr}$), the public key of sender and receiver ($B_{cs}, B_{cr}$), and the identity of the receiver ($ID_{cr}$), the sender performs this process for verifying the signature, and recovering a plain text ($m$) by undertaking the following steps:

(a) It computes $\beta = \partial \cdot B_{cr} + ID_{cs} \cdot \Delta \cdot \sigma \cdot B_{cs} \cdot A_{cr}$
(b) It decrypts $(m//ID_{cs}//ID_{cr}//n_{cs}) = d_\beta(\eta)$
(c) It computes $\sigma^\wedge = h_b(m//ID_{cs}//ID_{cr}//n_{cs})$
(d) It compares $\sigma^\wedge = \sigma$, if holds, then accept $\psi$ otherwise generate the error symbol $\perp\!\!\!\perp$

Note that, in the above algorithm, if $ID_{cs} = \text{null}$ and $ID_{cr} \neq \text{null}$, then generalized signcryption proceeds in an encryption process. If $ID_{cr} = \text{null}$ and $ID_{cs} \neq \text{null}$, then generalized signcryption will run in the signature mode. And, if $ID_{cs} \neq \text{null}$ and $ID_{cr} \neq null$, then generalized signcryption will run in signcryption mode.

*5.3. Correctness.* The receiver can compute the decryption key as

$$\begin{aligned}
\beta = {} & \partial.B_{cr} + ID_{cs} \cdot \Delta \cdot \sigma \cdot B_{cs}.A_{cr}, \\
& (ID_{cr} \cdot \varphi - \sigma \cdot \Delta \cdot A_{cs} \cdot ID_{cs}) \cdot B_{cr} + ID_{cs} \cdot \Delta \cdot \sigma \cdot B_{cs} \cdot A_{cr}, \\
& (ID_{cr} \cdot \varphi - \sigma \cdot \Delta \cdot A_{cs} \cdot ID_{cs}) \cdot B_{cr} + ID_{cs} \cdot \Delta \cdot \sigma \cdot B_{cs} \cdot A_{cr}, \\
& (ID_{cr} \cdot \varphi \cdot B_{cr} - \sigma \cdot \Delta \cdot A_{cs} \cdot ID_{cs} \cdot B_{cr}) + ID_{cs} \cdot \Delta \cdot \sigma \cdot B_{cs} \cdot A_{cr}, \\
& (ID_{cr} \cdot \varphi \cdot B_{cr} - \sigma.\Delta \cdot A_{cs} \cdot ID_{cs} \cdot B_{cr}) + ID_{cs} \cdot \Delta \cdot \sigma \cdot A_{cs} \cdot D \cdot A_{cr}, \\
& (ID_{cr} \cdot \varphi \cdot B_{cr} - \sigma \cdot \Delta \cdot A_{cs} \cdot ID_{cs} \cdot B_{cr}) + ID_{cs} \cdot \Delta \cdot \sigma \cdot A_{cs} \cdot B_{cr}, \\
& (ID_{cr} \cdot \varphi \cdot B_{cr} - \sigma \cdot \Delta \cdot A_{cs} \cdot ID_{cs} \cdot B_{cr}) + ID_{cs} \cdot \sigma \cdot \Delta \cdot A_{cs} \cdot B_{cr}, \\
& ID_{cr} \cdot \varphi \cdot B_{cr} = \beta,
\end{aligned} \tag{3}$$

and it verifies $\psi$ as it computes $\sigma\wedge = h_b(m//ID_{cs}//ID_{cr}//n_{cs})$ and compares $\sigma^\wedge = \sigma$. In case of equality, it accepts $\psi$ and else generates the error symbol $\perp\!\!\!\perp$.

## 6. Informal Security Analysis

This section is dedicated to spotlight the proposed scheme's contribution in upholding basic security including resistance to replay attack, confidentiality, integrity, and unforgeability. Each of the characteristics is briefly analyzed in the following sections.

*6.1. Confidentiality.* The proposed scheme ensures confidentiality. In case an intruder wants to steal the original contents of a message or the secret key, he/she must have beforehand information about the key as $\beta = \varphi \cdot B_{cr} \cdot B\,ID_{cr}$. In order to determine $\beta$, it is required to compute $\varphi$ from $\Delta = \varphi.D$, which is the discrete log problem in the hyperelliptic curve.

*6.2. Replay Attack.* The scheme offers replay attack resistance. Each session implies a fresh key ($\beta$) and a nonce ($n_{cs}$) i.e., $\eta = e_{\beta}(m//ID_{cs}//ID_{cr}//n_{cs})$. Therefore, it is, literally, not possible for an intruder of a session to penetrate another session with the same session key. Besides, the receiver is required to run a check for ascertaining the freshness of a message at every instance of reception. An obsoleteness, if spotted, renders the message useless.

*6.3. Integrity.* The sender takes the "hash value" of the message before sending the message, i.e.,: $\sigma = h_b(m//ID_{cs}//ID_{cr}//n_{cs})$. The "hash" exhibits a property of being an irreversible function. For the confirmation if either of the ciphertexts is altered or not, the receiver performs the following steps: it first decrypts $(m//ID_{cs}//ID_{cr}//n_{cs}) = d_{\beta}(\eta)$ and computes $\sigma^{\wedge} = h_b(m//ID_{cs}//ID_{cr}//n_{cs})$. After it compares $\sigma^{\wedge} = \sigma$, if it holds, then it accepts $\psi$; otherwise, it generates the error symbol $\perp\!\!\!\perp$.

*6.4. Unforgeability.* In our proposed scheme, if the intruder tries to generate a valid signature, then he/she is, first of all, required to compute $\partial = (ID_{cr} \cdot \varphi - \sigma \cdot \Delta \cdot A_{cs} \cdot ID_{cs})$, and to do so, the intruder needs to find $\varphi$ from $\Delta = \varphi \cdot D$ and $A_{cs}$ from $B_{cs} = A_{cs} \cdot D$. This equates to solving two hard problems with commensurate efforts. Thus, it is ensured that our designed approach offers resistance against the signature forging attack.

## 7. Deployment of the Proposed Scheme

In this phase, we provide the practical deployment of our proposed technique in the UAVs network for precision agriculture that involves monitoring of crop health in a cultivated field. The proposed scheme includes three subphases that are initializations, registration, and data transmission and verification, respectively.

*7.1. Initialization.* Figure 2 illustrates the initialization process, in which the PKG first calls the setup algorithm; i.e., it first selects a security parameter $\kappa$, picks a hyperelliptic curve (HEC) of the genus, chooses a parameter $q$ where the length is equivalent to 80 bits, selects a finite field $f_q$, where its order is $q$, picks a divisor $D$ of order $q$, select two one-way hash functions, i.e., $h_a$ and $h_b$, chooses a number uniformly for its private key as $\delta \in [1, 2, \ldots, (q-1)]$, computes its public as $\Lambda = \delta \cdot D$, produces all the public parameter $E = [q, h_a, h_b, f_q, \kappa, \Lambda, HEC, D]$, and published it to the network. Note that, in this subphase, we used $ID_{mec}$, $ID_{mbs}$, and $ID_{m-uav}$ for the identity of MEC-UAV, MBS/SBS, and M-UAV.

*7.2. Registration.* Figure 3 illustrates the registration process in which the PKG first calls the key extraction algorithm; i.e., when each of the participated contestants transmits its identity ($ID_{pc}$) to the PKG, then PKG generates the private and public keys as follows: it computes the private key for identity ($ID_{pc}$) as $A_{pc} = \delta \cdot h_a(ID_{pc}) \bmod q$, and then it computes public key for identity ($ID_{pc}$) as $B_{pc} = A_{pc} \cdot D$ Finally, PKG delivers the pair of public and private keys ($B_{pc}, A_{pc}$) to the participated contestants with its identity ($ID_{pc}$) by using the private network.note; in this subphase, we used ($A_{mec}$, $B_{mec}$), ($A_{mbs}$, $B_{mbs}$), and ($A_{m-uav}$, $B_{m-uav}$) for the private and public keys of MEC-UAV, MBS/SBS, and M-UAV.

*7.3. Data Transmission and Verification.* Figure 4 illustrates the data transmission and verification of the proposed scheme. In this phase, MEC-UAV performs the following process for generating a signcrypted ciphertext: it first selects a number in an irregular manner as $\varphi \in [1, 2, \ldots, (q-1)]$ and calculates $\Delta = \varphi \cdot D$. It also calculates $\beta = \varphi \cdot B_{mbs} \cdot D_{mbs}$ and computes $\eta = e_{\beta}(m//ID_{mec}//ID_{mbs}//n_{mec})$. Then, it computes $\sigma = h_b(m//ID_{mec}//ID_{mbs}//n_{mec})$ and $\partial = (ID_{mbs} \cdot \varphi - \sigma \cdot \Delta \cdot A_{mec} \cdot ID_{mec}) \bmod q$. Finally, it sends $\psi$ to MBS/SBS using an open network. Upon reception of $\psi$ MBS/SBS, it performs the verification and decryption process as follows: it computes $\beta = \partial \cdot B_{mbs} + ID_{mec} \cdot \Delta \cdot \sigma \cdot B_{mec} \cdot A_{mbs}$ and decrypts $(m//ID_{mec}//ID_{mbs}//n_{mec}) = d_{\beta}(\eta)$. It also computes $\sigma^{\wedge} = h_b(m//ID_{mec}//ID_{mbs}//n_{mec})$ and compares $\sigma^{\wedge} = \sigma$; if it holds, then, it accepts $\psi$; otherwise, it generates the error symbol $\perp\!\!\!\perp$.

In the above process, if $ID_{mec}$ = null and $ID_{mbs} \neq$ null, then MEC-UAV performs the encryption process. If $ID_{mbs}$ = null and $ID_{mec} \neq$ null, then MEC-UAV performs the signature method. If $ID_{mbs} \neq$ null and $ID_{mec} \neq$ null, then MEC-UAV performs the signcryption mode.

## 8. Performance Comparison

This section equates the performance of the proposed scheme with the existing counterparts suggested by Yu et al.'s scheme [13], Kushwah et al.'s scheme [35], Wei et al.'s scheme [36], Shen et al.'s scheme [37], and Zhou et al.'s scheme [39].

*8.1. Computational Cost.* For evaluating the effectiveness, the proposed scheme is compared with five existing schemes proposed by Yu et al. [13], Kushwah et al. [35], Wei et al. [36], Shen et al. [37], and Zhou et al. [39]. The major findings obtained from the comparison are depicted in Table 2. The five existing schemes utilize elliptic curve scalar multiplication and bilinear pairings, both of which are costlier options. Therefore, we apply the hyperelliptic divisor multiplication. From the observations, it has been revealed that the time taken for processing a single scalar multiplication varies considerably: Elliptic Curve Point Multiplication (ECPM), 0.97 ms; bilinear pairing, 14.90 ms; pairing-based point multiplications, 4.31 ms; and modular exponentiation, 1.25 ms [44]. In order to measure the performance of the proposed scheme, the Multiprecision Integer and Rational Arithmetic C Library (MIRACL) [12] is used. It tests the runtime of the basic cryptographic operations for about 1000 times. For testing the simulation results, a workstation

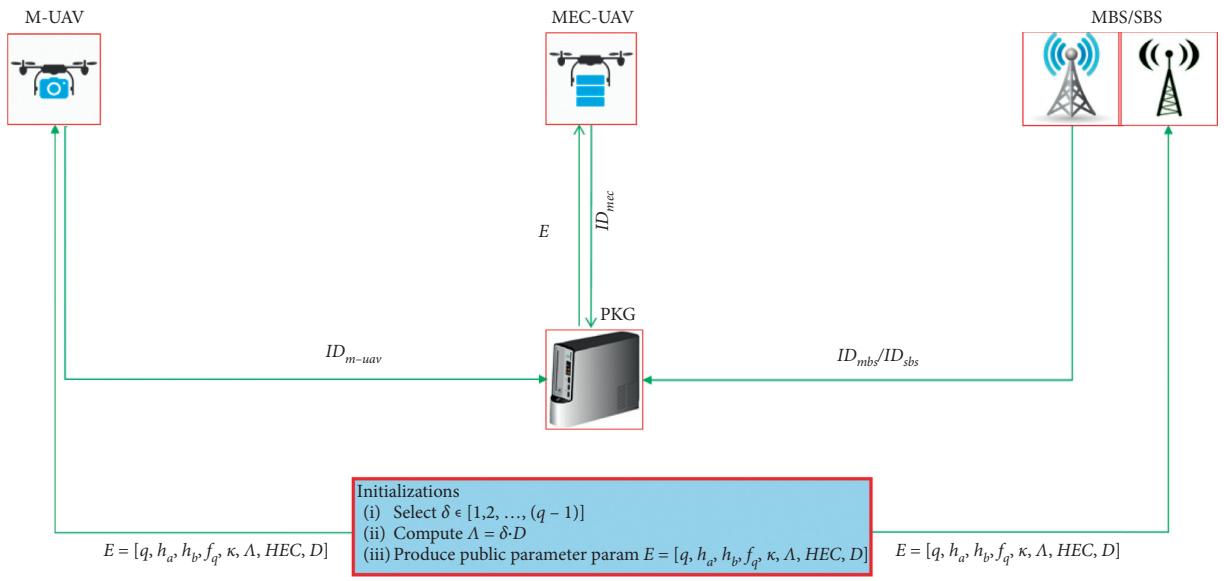



Figure 2: Initialization phase.

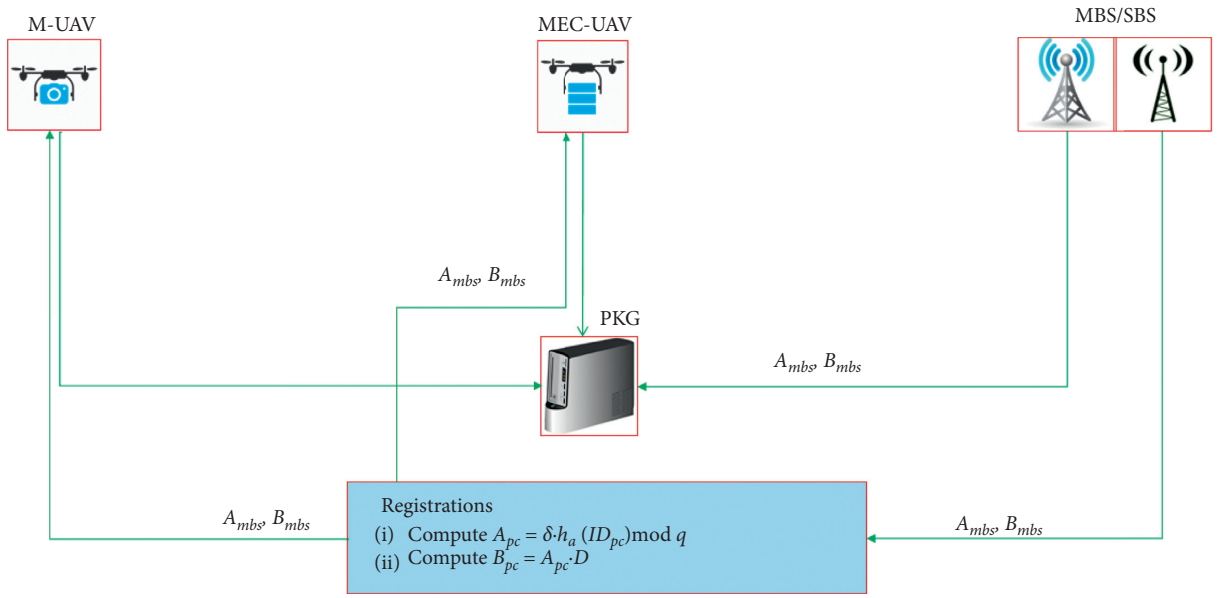



Figure 3: Registration phase.

having the following specifications is used: Intel Core i7-4510U CPU @ 2.0 GHz, 8 GB RAM, and Windows 7 Home Basic 64-bit Operating System [42]. Owing to a smaller key size of 80 bits, the Hyperelliptic Curve Divisor Multiplication (HCDM) is assumed to be of 0.48-millisecond duration [45, 46].

From the findings in Tables 2–4 and Figure 5, it is evident that our approach is far more efficient in terms of computational costs.

*8.2. Communication Cost.* This section is dedicated to discuss the comparison results in the perspective of communication costs. The proposed approach is compared with the existing five schemes presented by Yu et al. [13], Kushwah et al. [35], Wei et al. [36], Shen et al. [37], and Zhou et al. [39]. In the comparative analysis, the variables used along with the respective values are shown in Table 5 [40].

It is assumed that each of the schemes has associated communication costs as shown in Table 6.

From Figure 6, it is evident that a decision to opt for our proposed scheme results in a significant reduction in the associated communication costs. Table 7 depicts the percentage reduction in communication costs.

*8.3. Security Functionalities.* Here, the proposed scheme is compared with the existing schemes in terms of security functionalities. Table 8 lists the comparison outcomes based

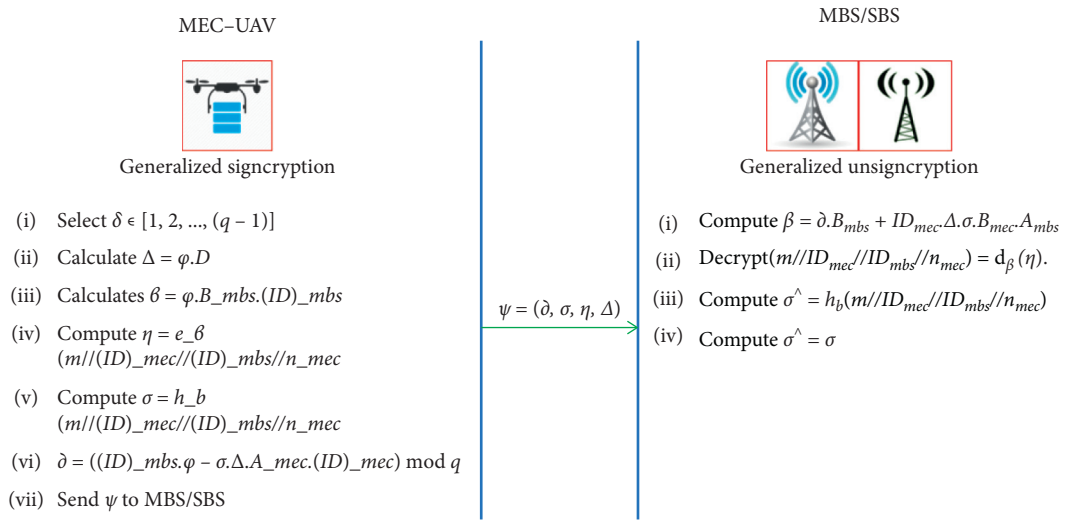



Figure 4: Data transmission and verification phase.

Table 2: Computational cost.

| Schemes | Generalized signcrypt | Generalized unsigncrypt | Total |
|---|---|---|---|
| Yu et al.'s scheme [13] | 4bpm + 1bp + 1mexp | 1bpm + 3bp + 3mexp | 5bpm + 4bp + 4mexp |
| Kushwah et al.'s scheme [35] | 5bpm + 2mexp | 4bpm + 2bp + 3mexp | 9bpm + 2bp + 5mexp |
| Wei et al.'s scheme [36] | 9bpm + 1bp + 7mexp | 2bpm + 4bp | 11bpm + 5bp + 7mexp |
| Shen et al.'s scheme [37] | 2bpm + 6mxp | 5bpm + 2mexp | 7bpm + 8mexp |
| Zhou et al.'s scheme [39] | 3bpm + 1bp | 1bpm + 2bp | 4bpm + 3bp |
| Proposed | 6 hm | 5 hm | 11 hm |

hm = hyperelliptic curve divisor multiplication, em = elliptic curve scalar multiplication, bp = bilinear pairing, bpm = pairing-based point multiplications, mexp = modular exponentiation.

Table 3: Computational cost in milliseconds.

| Schemes | Generalized signcrypt (ms) | Generalized unsigncrypt (ms) | Total (ms) |
|---|---|---|---|
| Yu et al.'s scheme [13] | 33.39 | 58.38 | 86.23 |
| Kushwah et al.'s scheme [35] | 24.05 | 50.79 | 74.84 |
| Wei et al.'s scheme [36] | 62.44 | 68.22 | 130.66 |
| Shen et al.'s scheme [37] | 16.12 | 24.05 | 40.17 |
| Zhou et al.'s scheme [39] | 27.83 | 34.11 | 61.94 |
| Proposed | 2.88 | 2.40 | 5.28 |

Table 4: Percentage improvement in computational cost.

| Schemes | Total computational cost of extant scheme ($x$) (%) | Total computational cost of proposed scheme ($y$) (%) | $z$ (using the formula**) (%) |
|---|---|---|---|
| Yu et al.'s scheme [13] | 86.23 | 5.28 | 93.87 |
| Kushwah et al.'s scheme [35] | 74.84 | 5.28 | 92.94 |
| Wei et al.'s scheme [36] | 130.66 | 5.28 | 95.95 |
| Shen et al.'s scheme [37] | 40.17 | 5.28 | 86.85 |
| Zhou et al.'s scheme [39] | 61.94 | 5.28 | 91.47 |

** Percentage change, $z = x - y/x * 100$.

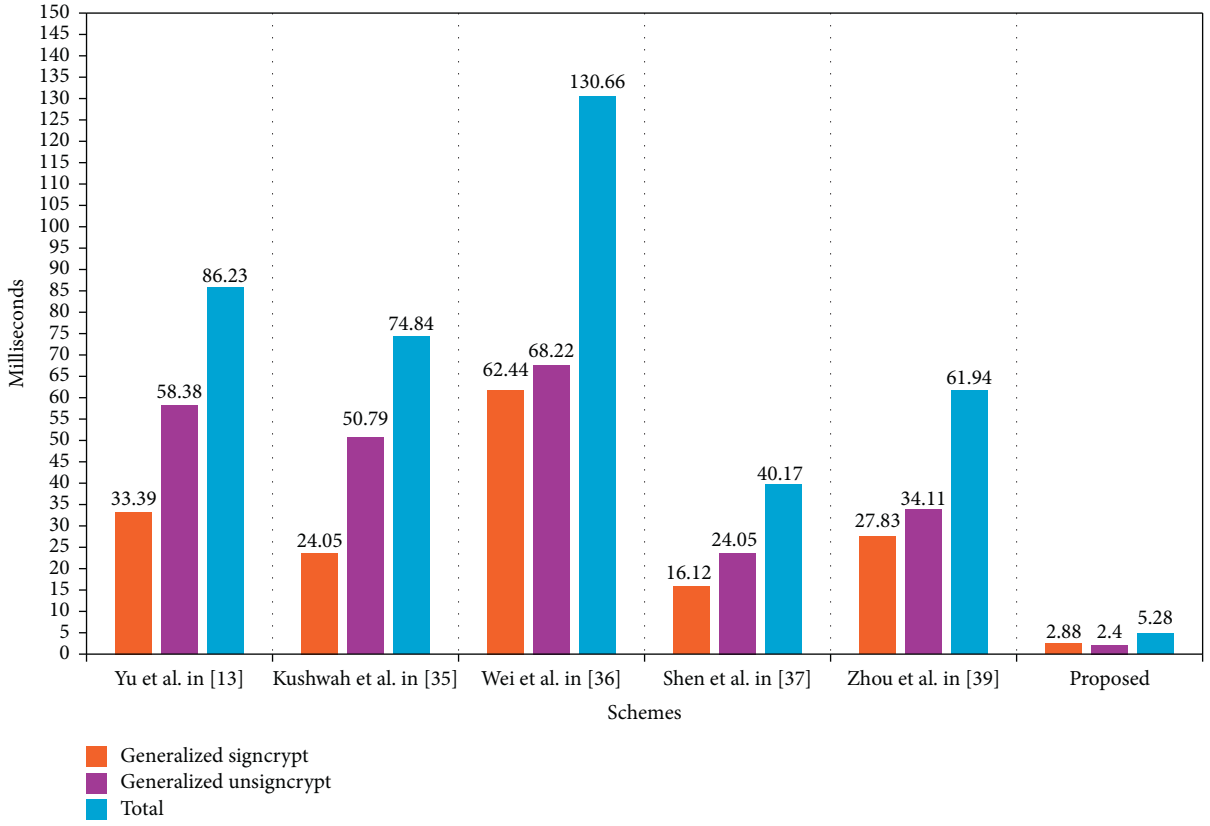



Figure 5: Computational cost (in ms).

Table 5: Variables used for a communication cost comparison.

| Variable | Value (bits) |
|---|---|
| $\|S\|$ | 1024 |
| $\|Z_q\|$ | 160 |
| $\|Z_n\|$ | 80 |
| $\|H\|$ | 512 |
| $\|m\|$ | 1024 |
| $\|W\|$ | 1024 |

Table 6: Communication cost.

| Schemes | Communication cost |
|---|---|
| Yu et al.'s scheme [13] | $\|S\| + \|m\|$ |
| Kushwah et al.'s scheme [35] | $\|S\| + \|m\|$ |
| Wei et al.'s scheme [36] | $7\|S\| + \|m\|$ |
| Shen et al.'s scheme [37] | $4\|S\| + \|m\|$ |
| Zhou et al.'s scheme [39] | $\|S\| + \|m\|$ |
| Proposed scheme | $3\|Z_n\| + \|m\|$ |

on the following security parameters: unforgeability, integrity, replay attack, and formal analysis. From the table, it can be seen that none of the existing schemes offer a replay attack.

## 9. Flying Ad Hoc Network-Based Precision Agriculture: A Case Study

To further assess the practicability, the proposed scheme is applied to a precision agriculture case that involves FANETs for monitoring the health of the crops. Small UAVs are used to capture the images, which are, in the next step, processed to extract useful information. Values from the Normalized Difference Vegetation Index (NDVI) are computed to differentiate healthy plants from the nonhealthy ones. This is done by measuring the chlorophyll content. It further helps in the localization of the area under stress. The images captured by the M-UAVs are transferred to the MEC-UAV, which, utilizing the onboard microcontroller, generates the respective tasks to be carried on by the Decision Support Engine (DSE). For value addition and versatility, the M-UAVs can have additional gadgets, such as cameras, IMU, sensors, and GPS units. The web portal contains a variety of services such as visualization of historical/real data, NDVI mapping, and the correlation functionality.

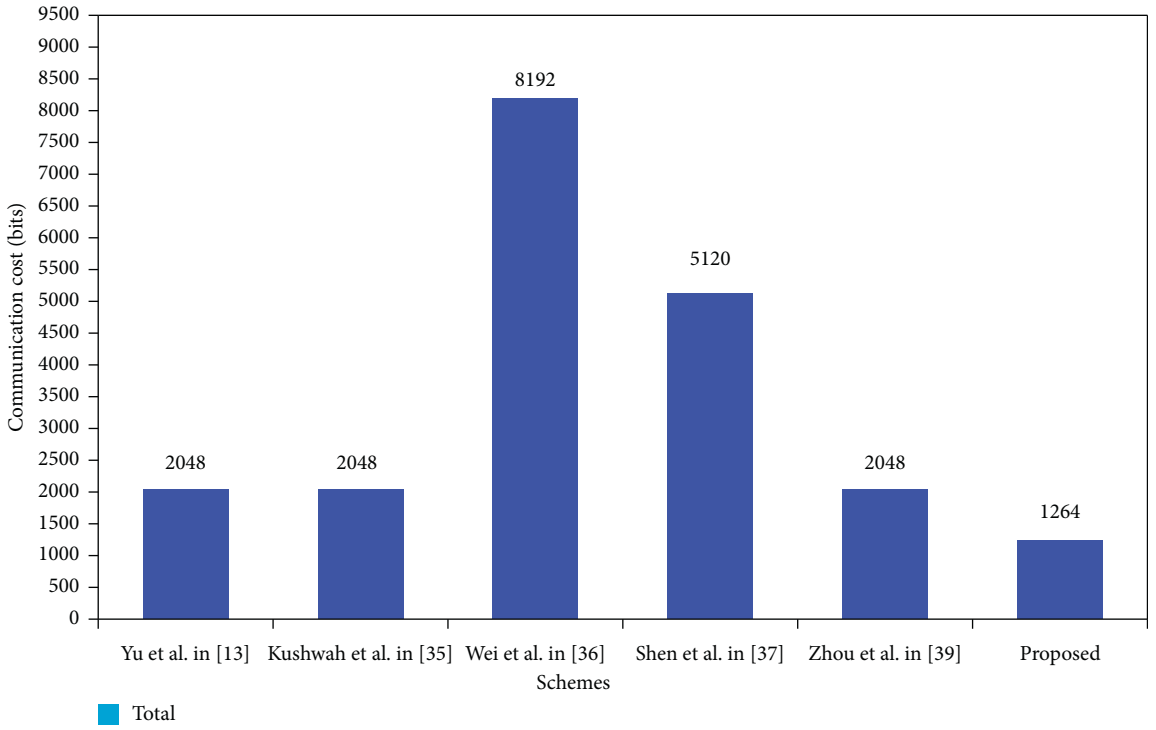



Figure 6: Total communication cost (in bits).

Table 7: Percentage reduction in communication cost.

| Scheme | Equation for evaluating reduction | Resulting reduction in communication cost (%) |
|---|---|---|
| Yu et al.'s scheme [13] | $(\|S\| + \|m\|) - (3\|Z_n\| + \|m\|)/(\|S\| + \|m\|)$ | 38.28 |
| Kushwah et al.'s scheme [35] | $(\|S\| + \|m\|) - (3\|Z_n\| + \|m\|)/(\|S\| + \|m\|)$ | 38.28 |
| Wei et al.'s scheme [36] | $(7\|S\| + \|m\|) - (3\|Z_n\| + \|m\|)/(\|S\| + \|m\|)$ | 84.57 |
| Shen et al.'s scheme [37] | $(4\|S\| + \|m\|) - (3\|Z_n\| + \|m\|)/(\|S\| + \|m\|)$ | 75.31 |
| Zhou et al.'s scheme [39] | $(\|S\| + \|m\|) - (3\|Z_n\| + \|m\|)/(\|S\| + \|m\|)$ | 38.28 |

Table 8: Comparison with relevant existing schemes.

| Schemes | Security functionalities | | | | |
|---|---|---|---|---|---|
| | Informal | | | | Formal |
| | U | I | C | RA | FA |
| Yu et al.'s scheme [13] | ✓ | ✓ | ✓ | ✗ | ✗ |
| Kushwah et al.'s scheme [35] | ✓ | ✓ | ✓ | ✗ | ✗ |
| Wei et al.'s scheme [36] | ✓ | ✓ | ✓ | ✗ | ✗ |
| Shen et al.'s scheme [37] | ✓ | ✓ | ✓ | ✗ | ✗ |
| Zhou et al.'s scheme [39] | ✓ | ✓ | ✓ | ✗ | ✗ |
| Proposed | ✓ | ✓ | ✓ | ✓ | ✓ |

U: unforgeability, I: integrity, RA: replay attack, FA: formal analysis. The symbol ✓ satisfies the security functionality; ✗ does not satisfy the security functionality.

## 10. Conclusions

There is an evolving trend of combining multiple small UAVs, as a flying ad hoc network (FANET), to cater to the needs of future applications that demand autonomy and pervasiveness. However, the small UAVs inherent limited onboard energy and restricted computational capability. Such limitations hinder their deployment for longer time-

```
role
role_Mecuav(Mecuav:agent, Mbssbs:agent, Bmec:public_key, Bmbs:public_key, SND, RCV:channel(dy))
played_by Mecuav
def=
        local
        State:nat, Add:hash_func, Phii:text, Idmec:text, Delta:text, Idmbs:text, Nmec:text,M:text, Encrypts:hash_func, Beeta:
symmetric_key
        init
           State := 0
        transition
           1. State = 0 /\ RCV(start) = |> State' : = 1 /\ SND(Mecuav.Mbssbs)
           2. State = 1 /\ RCV(Mbssbs.{Nmec'}_Bmbs) = |> State' : = 2 /\ Idmbs' : = new() /\ Phii' : = new() /\ Delta' : = new() /\ Idmec' :
 = new() /\ Beeta' : = new() /\ M' : = new() /\ secret(M',sec_2,{Mecuav}) /\ witness(Mecuav, Mbssbs,auth_1,M') /\
SND(Mecuav.{Encrypts(M'.Nmec'.Idmec'.Idmbs')}_Beeta'.{Add(Idmec'.Phii'.Delta'.Phii'.Idmbs')}_inv(Bmec))
end role
```

Algorithm 1: High-level protocol specification language (HLPSL) code for the MEC-UAV role.

```
role
role_Mbssbs(Mecuav:agent, Mbssbs:agent, Bmec:public_key,Bmbs:public_key,SND,RCV:channel(dy))
played_by Mbssbs
def=
        local
        State:nat,Add:hash_func, Phii:text, Idmec:text, Delta:text, Idmbs:text, Nmec:text,M:text, Encrypts:hash_func, Beeta:
symmetric_key
        init
           State := 0
        transition
           1. State = 0 /\ RCV(Mecuav.Mbssbs) = |> State' : = 1 /\ Nmec' : = new() /\ SND(Mbssbs.{Nmec'}_Bmbs)
           6. State = 1 /\ RCV(Mecuav.{Encrypts(M'.Nmec.Idmec'.Idmbs')}_Beeta'.{Add(Idmec'.Phii'.Delta'.Phii'.Idmbs')}
_inv(Bmec)) = |> State' : = 2 /\ request(Mbssbs, Mecuav, auth_1, M') /\ secret(M',sec_2,{Mecuav})
end role
```

Algorithm 2: High-level protocol specification language (HLPSL) code for MBS role.

```
role session1(Mecuav:agent, Mbssbs:agent, Bmec:public_key, Bmbs:public_key)
def=
        local
           SND2, RCV2, SND1, RCV1: channel(dy)
        composition
           role_Mbssbs(Mecuav, Mbssbs,Bmec, Bmbs,SND2,RCV2) /\ role_Mecuav(Mecuav, Mbssbs, Bmec, Bmbs, SND1, RCV1)
end role
role session2(Mecuav:agent, Mbssbs:agent, Bmec:public_key, Bmbs:public_key)
def=
        local
           SND1, RCV1:channel(dy)
        composition
           role_Mecuav(Mecuav, Mbssbs,Bmec, Bmbs, SND1, RCV1)
end role
```

Algorithm 3: High-level protocol specification language (HLPSL) code for Sessions role.

intervals and complex cryptographic operations. Addressing such deficiency, in this article, utilizing the concept of the hyperelliptic curve (HEC), we propose an efficient lightweight security scheme, called identity-based generalized signcryption. The scheme is based on multiaccess edge computing (MEC). The HEC approach is

```
role environment()
def=
        const
        hash_0:hash_func, bmec:public_key,alice:agent,bob:agent, bmbs:public_key,const_1:agent, const_5:public_key,const_9:
public_key,auth_1:protocol_id,sec_2:protocol_id
        intruder_knowledge = {alice, bob}
        composition
           session2(i, const_1,const_5,const_9) /\ session1(alice, bob, bmec, bmbs)
end role
goal
        authentication_on auth_1
        secrecy_of sec_2
end goal
environment()
```

Algorithm 4: High-level protocol specification language (HLPSL) code for environment role.

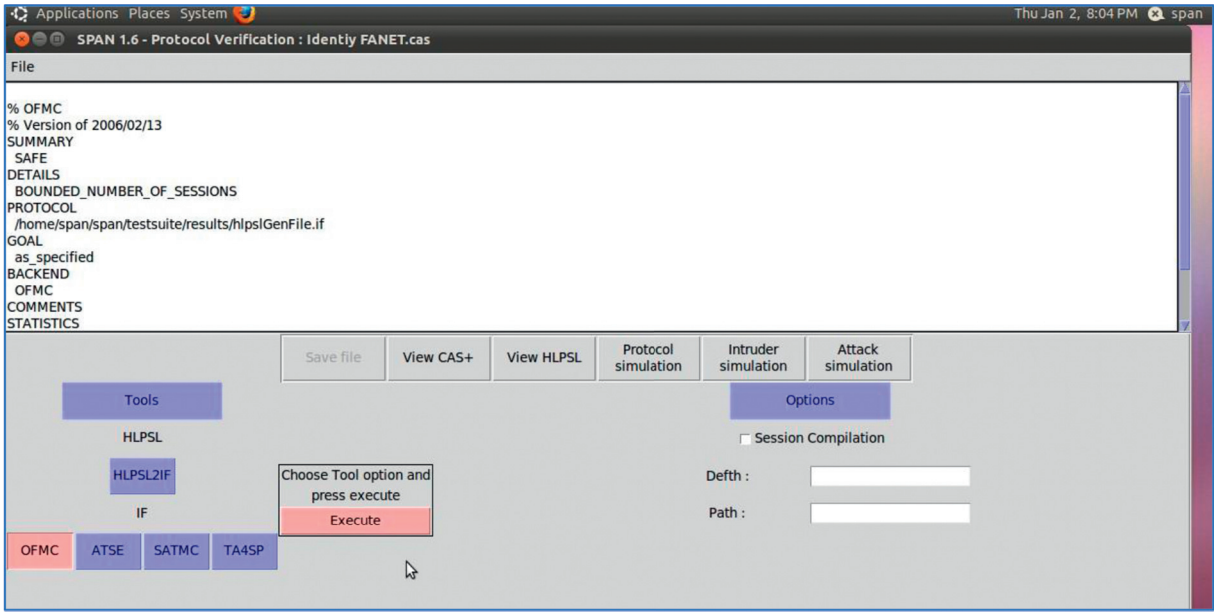



Figure 7: Simulation results for on-the-fly model-checker (OFMC).

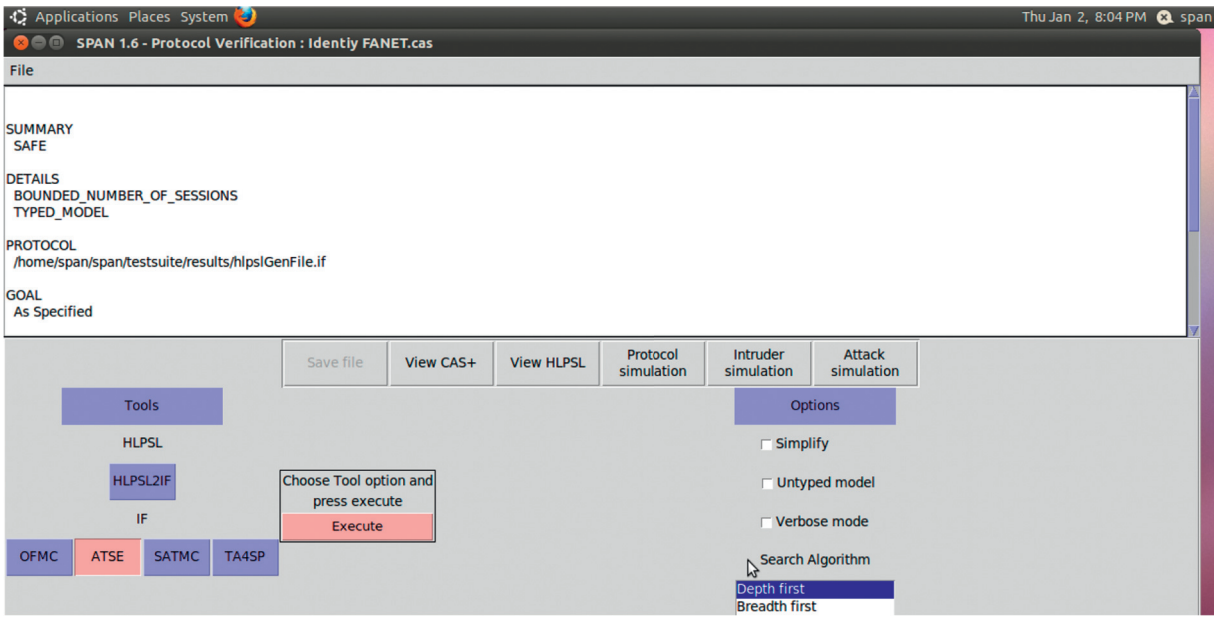



Figure 8: Simulation results for AtSe.

effective in generating small keys and is, therefore, suitable for low-computational devices such as small UAVs. Both formal and informal security analyses, using the AVISPA tool, demonstrate the potency of the proposed scheme in thwarting various known and unknown cyberattacks. Moreover, upon comparative analysis with the major existing counterparts, the scheme has demonstrated to be efficient in terms of computational and communication costs.

For our future work, we aim to complement the research work by including other aspects of formal analysis, such as the Real-Or-Random (ROR) model and Random Oracle Model (ROM). Moreover, we also intend to incorporate a computational offloading and scheduling mechanism, in which the M-UAVs will be able to offload and schedule the computing tasks to the MEC-UAV for improved processing power and faster execution.

## Appendix

## Implementation of Our Proposed Scheme in AVISPA

High-level protocol specification language (HLPSL) has been consulted to implement the proposed scheme for MEC-UAV and MBS. This has been illustrated in Algorithms 1 and 2. To run the simulations, a Haier Win8.1 PC computer workstation powered with an Intel (R) Core (TM) i3-4010U CPU @ 1.70 GHz and 64-bit Operating System was chosen. The software part of the setup is composed of Oracle VM Virtual Box (version: 5.2.0.118431) and SPAN (version: SPAN-Ubuntu-10.10-light_1). From Algorithms 3 and 4, the roles for session, goal, and environment have been executed to comply with the conventions. The execution test considers OFMC and CL-AtSe back ends for evaluating the system's susceptibility to attacks. The simulation results do not include the results of SATMC and TA4SP. It is because SATMC and TA4SP are not compatible with bitwise XOR operations. Another factor worthy of consideration is the requirement to monitor the execution of a specified protocol. Therefore, the back ends delegated the responsibility to check operations. In order to verify the Dolev-Yao (DY) model, the back ends also estimate the vulnerability of the system to man-in-the-middle attack [42]. The widely known web-tool SPAN (Specific Protocol Animator for AVISPA) is also used to simulate the proposed scheme. The results obtained from OFMC (Figure 7) and AtSe (Figure 8) further demonstrate the scheme's potency against replay and man-in-the-middle attacks.

## Data Availability

All data generated or analysed during this study are included in this published article.

## Conflicts of Interest

The authors declare no conflicts of interest with respect to the research, authorship, and/or publication of this article.

## References

[1] M. A. Khan, B. A. Alvi, A. Safi, and I. U. Khan, "Drones for good in smart cities: a review," in *Proceedings of the 2018 International Conference on Electrical, Electronics, Computers, Communication, Mechanical and Computing (EECCMC*, pp. 1–6, Vaniyambadi, India, January 2018.

[2] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, "Drone networks: communications, coordination, and sensing," *Ad Hoc Networks*, vol. 68, pp. 1–15, 2018.

[3] V. Sharma, "Advances in drone communications, state-of-the-art and architectures," *Drones*, vol. 3, no. 1, p. 21, 2019.

[4] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, "Routing in flying ad hoc networks: survey, constraints, and future challenge perspectives," *IEEE Access*, vol. 7, pp. 81057–81105, 2019.

[5] V. Sharma and R. Kumar, "G-FANET: an ambient network formation between ground and flying ad hoc networks," *Telecommunication Systems*, vol. 65, no. 1, pp. 31–54, 2017.

[6] M. A. Khan, I. M. Qureshi, and F. Khanzada, "A hybrid communication scheme for efficient and low-cost deployment of future flying ad-hoc network (FANET)," *Drones*, vol. 3, no. 1, p. 16, 2019.

[7] M. Marchese, A. Moheddine, and F. Patrone, "IoT and UAV integration in 5G hybrid terrestrial-satellite networks," *Sensors*, vol. 19, no. 17, p. 3704, 2019.

[8] C. Lin, D. He, N. Kumar, K.-K. R. Choo, A. Vinel, and X. Huang, "Security and privacy for the Internet of drones: challenges and solutions," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 64–69, 2018.

[9] Y. Zhi, Z. Fu, X. Sun, and J. Yu, "Security and privacy issues of UAV: a survey," *Mobile Networks and Applications*, vol. 25, pp. 95–101, 2019.

[10] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) ≪ cost(signature) + cost(encryption)," in *Proceedings of the Advances in Cryptology - CRYPTO '97*, pp. 165–179, Springer, Santa Barbara, CA, USA, August 1997.

[11] Y. Han, X. Yang, P. Wei, Y. Wang, and Y. Hu, "ECGSC: elliptic curve based generalized signcryption," in *Proceedings of the Third International Conference Ubiquitous Intelligence and Computing, Vol. 4159 of Lecture Notes in Computer Science*, Springer, Wuhan, China, pp. 956–965, September 2006.

[12] Shamus Sofware Ltd, "Miracl Library," GitHub, Inc., San Francisco, CA, USA, http://github.com/miracl/MIRACL.

[13] G. Yu, X. Ma, Y. Shen, and W. Han, "Provable secure identity based generalized signcryption scheme," *Theoretical Computer Science*, vol. 411, no. 40-42, pp. 3614–3624, 2010.

[14] M. Suárez-Albela, P. Fraga-Lamas, and T. Fernández-Caramés, "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, p. 3868, 2018.

[15] M. Yu1, J. Zhang, J. Wang et al., "Internet of Things security and privacy-preserving method through nodes differentiation, concrete cluster centers, multi-signature, and blockchain," *International Journal of Distributed Sensor Network*, vol. 14, p. 12, 2018.

[16] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, p. 8, 2018.

[17] C. Zhou, Z. Zhao, W. Zhou, and Y. Mei, "Certificateless key-insulated generalized signcryption scheme without bilinear pairings," *Security and Communication Network*, vol. 2017, Article ID 8405879, 17 pages, 2017.

[18] 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, p. 12, 2017.

[19] A. Omala, A. Mbandu, K. Mutiria, C. Jin, and F. Li, "Provably secure heterogeneous access control scheme for wireless body area network," *Journal of Medical Systems*, vol. 42, p. 6, 2018.

[20] C. Tamizhselvan and V. Vijayalakshmi, "An energy efficient secure distributed naming service for IoT," *International Journal of Advanced Studies of Scientific Research*, vol. 3, p. 8, 2019.

[21] V. S. Naresh, R. Sivaranjani, and N. V. E. S. Murthy, "Provable secure lightweight hyper elliptic curve-based communication system for wireless sensor Network," *International Journal of Communication Systems*, vol. 31, p. 15, 2018.

[22] A. Rahman, I. Ullah, M. Naeem, R. Anwar, H. Khattak, and S. Ullah, "A lightweight multi-message and multi-receiver heterogeneous hybrid signcryption scheme based on hyper elliptic curve," *International Journal of Advanced Computer Science and Applications*, vol. 9, p. 5, 2018.

[23] S. Ouahouah, T. Taleb, J. Song, and C. Benzaid, "Efficient offloading mechanism for UAVs-based value-added services," in *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, May 2017.

[24] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based iot platform: a crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[25] S. Garg, A. Singh, S. Batra, N. Kumar, and L. T. Yang, "UAV-empowered edge computing environment for cyber-threat detection in smart Vehicles," *IEEE Network*, vol. 32, no. 3, pp. 42–51, 2018.

[26] G. K. Xilouris, M. C. Batistatos, G. E. Athanasiadou, G. Tsoulos, H. B. Pervaiz, and C. C. Zarakovitis, "UAV-assisted 5G network architecture with slicing and virtualization," in *Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–7, Abu Dhabi, UAE, December 2018.

[27] C. Grasso and G. Schembra, "A fleet of MEC UAVs to extend a 5G network slice for video monitoring with low-latency constraints," *Journal of Sensor and Actuator Networks*, vol. 8, no. 1, p. 3, 2019.

[28] W. Chen, B. Liu, H. Huang, S. Guo, and Z. Zheng, "When UAV swarm meets edge-cloud computing: the QoS perspective," *IEEE Network*, vol. 33, no. 2, pp. 36–43, 2019.

[29] J.-M. Fernandez, I. Vidal, and F. Valera, "Enabling the orchestration of IoT slices through edge and cloud microservice platforms," *Sensors*, vol. 19, no. 13, p. 2980, 2019.

[30] S. He, Q. Wu, J. Liu, W. Hu, B. Qin, and Y. N. Li, "Secure communications in unmanned aerial vehicle network," in *Proceedings of the International Conference on Information Security Practice and Experience*, Springer, Melbourne, Australia, pp. 601–620, December 2017.

[31] J. Won, S.-H. Seo, and E. Bertino, "Certificateless cryptographic protocols for efficient drone-based smart city applications," *IEEE Access*, vol. 5, pp. 3721–3749, 2017.

[32] J. Won, S. H. Seo, and E. Bertino, "A secure communication protocol for drones and smart objects," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ser. ASIA CCS'15*, ACM, Singapore, pp. 249–260, April 2015.

[33] J. Srinivas, A. K. Das, N. Kumar, and J. P. C. Rodrigues, "CloudCentric authentication for wearable healthcare monitoring system," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[34] S. Lal and P. Kushwah, "ID based generalized signcryption," Cryptology ePrint Archive, Report 2008/084, October 2019, http://eprint.iacr.org/2008/084.

[35] P. Kushwah and S. Lal, "An efficient identity based generalized signcryption scheme," *Theoretical Computer Science*, vol. 412, no. 45, pp. 6382–6389, 2011.

[36] G. Wei, J. Shao, Y. Xiang, P. Zhu, and R. Lu, "Obtain confidentiality or/and authenticity in Big Data by ID-based generalized signcryption," *Information Sciences*, vol. 318, pp. 111–122, 2015.

[37] X. Shen, Y. Ming, J. Feng, X. Shen, Y. Ming, and J. Feng, "Identity based generalized signcryption scheme in the standard model," *Entropy*, vol. 19, no. 3, p. 121, 2017.

[38] A. Waheed, A. I. Umar, N. Din, N. U. Amin, S. Abdullah, and P. Kumam, "Cryptanalysis of an authentication scheme using an identity based generalized signcryption," *Mathematics*, vol. 7, no. 9, p. 782, 2019.

[39] Y. Zhou, Z. Li, F. Hu, and F. Li, "Identity-based combined public key schemes for signature, encryption, and signcryption," in *Information Technology and Applied Mathematics*, pp. 3–22, Springer, Singapore, 2019.

[40] I. Ullah, A. Alomari, N. Ul Amin, M. A. Khan, and H. Khattak, "An energy efficient and formally secured certificate-based signcryption for wireless body area networks with the Internet of Things," *Electronics*, vol. 8, no. 10, p. 1171, 2019.

[41] Raspberry pi 4," 2019, https://www.raspberrypi.org/.

[42] 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.

[43] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques—Advances in Cryptology (EUROCRYPT'02)*, pp. 337–351, Amsterdam, The Netherlands, May 2002.

[44] C. Zhou, "An improved lightweight certificateless generalized signcryption scheme for mobile-health system," *International Journal of Distributed Sensor Networks*, vol. 15, no. 1, 2019.

[45] M. A. Khan, I. M. Qureshi, I. Ullah, S. Khan, F. Khanzada, and F. Noor, "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, 2019.

[46] M. A. Khan, I. Ullah, S. Nisa et al., "An efficient and provably secure certificateless key-encapsulated signcryption scheme for flying ad-hoc network," *IEEE Access*, vol. 8, pp. 36807–36828, 2020.