

Security and Privacy Challenges in Internet of Things and Mobile Edge Computing

Lead Guest Editor: Jinbo Xiong

Guest Editors: Qing Yang, Narasimha Shashidhar, and Zuobin Ying





Security and Privacy Challenges in Internet of Things and Mobile Edge Computing

Security and Communication Networks

Security and Privacy Challenges in Internet of Things and Mobile Edge Computing

Lead Guest Editor: Jinbo Xiong

Guest Editors: Qing Yang, Narasimha Shashidhar,
and Zuobin Ying







Copyright © 2021 Hindawi Limited. All rights reserved.

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

Chief Editor

Roberto Di Pietro, Saudi Arabia

Associate Editors

Jiankun Hu , Australia
Emanuele Maiorana , Italy
David Megias , Spain
Zheng Yan , China

Academic Editors



Saed Saleh Al Rabae , United Arab Emirates
Shadab Alam, Saudi Arabia
Goutham Reddy Alavalapati , USA
Jehad Ali , Republic of Korea
Jehad Ali, Saint Vincent and the Grenadines
Benjamin Aziz , United Kingdom
Taimur Bakhshi , United Kingdom
Spiridon Bakiras , Qatar
Musa Balta, Turkey
Jin Wook Byun , Republic of Korea
Bruno Carpentieri , Italy
Luigi Catuogno , Italy
Ricardo Chaves , Portugal
Chien-Ming Chen , China
Tom Chen , United Kingdom
Stelvio Cimato , Italy
Vincenzo Conti , Italy
Luigi Coppolino , Italy
Salvatore D'Antonio , Italy
Juhriyansyah Dalle, Indonesia
Alfredo De Santis, Italy
Angel M. Del Rey , Spain
Roberto Di Pietro , France
Wenxiu Ding , China
Nicola Dragoni , Denmark
Wei Feng , China
Carmen Fernandez-Gago, Spain
AnMin Fu , China
Clemente Galdi , Italy
Dimitrios Geneiatakis , Italy
Muhammad A. Gondal , Oman
Francesco Gringoli , Italy
Biao Han , China
Jinguang Han , China
Khizar Hayat, Oman
Azeem Irshad, Pakistan

M.A. Jabbar , India
Minho Jo , Republic of Korea
Arijit Karati , Taiwan
ASM Kayes , Australia
Farrukh Aslam Khan , Saudi Arabia
Fazlullah Khan , Pakistan
Kiseon Kim , Republic of Korea
Mehmet Zeki Konyar, Turkey
Sanjeev Kumar, USA
Hyun Kwon, Republic of Korea
Maryline Laurent , France
Jegatha Deborah Lazarus , India
Huaizhi Li , USA
Jiguo Li , China
Xueqin Liang, Finland
Zhe Liu, Canada
Guangchi Liu , USA
Flavio Lombardi , Italy
Yang Lu, China
Vincente Martin, Spain
Weizhi Meng , Denmark
Andrea Michienzi , Italy
Laura Mongioi , Italy
Raul Monroy , Mexico
Naghme Moradpoor , United Kingdom
Leonardo Mostarda , Italy
Mohamed Nassar , Lebanon
Qiang Ni, United Kingdom
Mahmood Niazi , Saudi Arabia
Vincent O. Nyangaresi, Kenya
Lu Ou , China
Hyun-A Park, Republic of Korea
A. Peinado , Spain
Gerardo Pelosi , Italy
Gregorio Martinez Perez , Spain
Pedro Peris-Lopez , Spain
Carla Ràfols, Germany
Francesco Regazzoni, Switzerland
Abdalhossein Rezai , Iran
Helena Rifà-Pous , Spain
Arun Kumar Sangaiah, India
Nadeem Sarwar, Pakistan
Neetesh Saxena, United Kingdom
Savio Sciancalepore , The Netherlands


De Rosal Ignatius Moses Setiadi ,
Indonesia
Wenbo Shi, China
Ghanshyam Singh , South Africa
Vasco Soares, Portugal
Salvatore Sorce , Italy
Abdulhamit Subasi, Saudi Arabia
Zhiyuan Tan , United Kingdom
Keke Tang , China
Je Sen Teh , Australia
Bohui Wang, China
Guojun Wang, China
Jinwei Wang , China
Qichun Wang , China
Hu Xiong , China
Chang Xu , China
Xuehu Yan , China
Anjia Yang , China
Jiachen Yang , China
Yu Yao , China
Yinghui Ye, China
Kuo-Hui Yeh , Taiwan
Yong Yu , China
Xiaohui Yuan , USA
Sherali Zeadally, USA
Leo Y. Zhang, Australia
Tao Zhang, China
Youwen Zhu , China
Zhengyu Zhu , China

Contents


GNS: Forge High Anonymity Graph by Nonlinear Scaling Spectrum

Yong Zeng , Yixin Li , Zhongyuan Jiang, and Jianfeng Ma
Research Article (11 pages), Article ID 8609278, Volume 2021 (2021)

Edge Computing Assisted an Efficient Privacy Protection Layered Data Aggregation Scheme for IIoT

Rong Ma, Tao Feng , and Junli Fang
Research Article (10 pages), Article ID 7776193, Volume 2021 (2021)




Private Data Aggregation Based on Fog-Assisted Authentication for Mobile Crowd Sensing

Ruyan Wang, Shiqi Zhang , Zhigang Yang, Puning Zhang, Dapeng Wu, Yongling Lu, and Alexander Fedotov
Research Article (12 pages), Article ID 7354316, Volume 2021 (2021)

TASC-MADM: Task Assignment in Spatial Crowdsourcing Based on Multiattribute Decision-Making

Yunhui Li , Liang Chang , Long Li , Xuguang Bao , and Tianlong Gu 
Research Article (14 pages), Article ID 5448397, Volume 2021 (2021)

Privacy-Preserving Incentive Mechanism for Mobile Crowdsensing

Tao Wan , Shixin Yue , and Weichuan Liao 
Research Article (17 pages), Article ID 4804758, Volume 2021 (2021)



An Efficient and Provable Multifactor Mutual Authentication Protocol for Multigateway Wireless Sensor Networks

Shuailiang Zhang , Xiujuan Du , and Xin Liu
Research Article (17 pages), Article ID 2037188, Volume 2021 (2021)

From Unknown to Similar: Unknown Protocol Syntax Analysis for Network Flows in IoT

Yichuan Wang , Han Yu , Xinhong Hei , Binbin Bai, and Wenjiang Ji
Research Article (13 pages), Article ID 9179286, Volume 2021 (2021)




Improved Outsourced Provable Data Possession for Secure Cloud Storage

Haibin Yang, Zhengge Yi , Ruifeng Li, Zheng Tu, Xu An Wang , Yuanyou Cui, and Xiaoyuan Yang
Research Article (12 pages), Article ID 1805615, Volume 2021 (2021)





Anticollusion Attack Strategy Combining Trust Metrics and Secret Sharing for Friendships Protection

Junfeng Tian  and Yue Li 
Research Article (14 pages), Article ID 9717747, Volume 2021 (2021)




A Secure Truth Discovery for Data Aggregation in Mobile Crowd Sensing

Taochun Wang , Chengmei Lv, Chengtian Wang, Fulong Chen , and Yonglong Luo 
Research Article (15 pages), Article ID 2296386, Volume 2021 (2021)



A Lightweight Three-Factor Authentication and Key Agreement Scheme for Multigateway WSNs in IoT

Lingyan Xue, Qinglong Huang , Shuaiqing Zhang , Haiping Huang , and Wenming Wang 
Research Article (15 pages), Article ID 3300769, Volume 2021 (2021)





A Privacy-Preserving Identity Authentication Scheme Based on the Blockchain

Sheng Gao , Qianqian Su , Rui Zhang , Jianming Zhu, Zhiyuan Sui, and Junsheng Wang
Research Article (10 pages), Article ID 9992353, Volume 2021 (2021)




Privacy-Preserving Attribute-Based Keyword Search with Traceability and Revocation for Cloud-Assisted IoT

Kai Zhang , Yanping Li , and Laifeng Lu 
Research Article (13 pages), Article ID 9929663, Volume 2021 (2021)



Aggregating Heterogeneous Sensor Ontologies with Fuzzy Debate Mechanism

Xingsi Xue , Xiaojing Wu , Jie Zhang, Lingyu Zhang , Hai Zhu , and Guojun Mao
Research Article (12 pages), Article ID 2878684, Volume 2021 (2021)

PUF-Based Mutual-Authenticated Key Distribution for Dynamic Sensor Networks

Yanan Liu , Yijun Cui, Lein Harn, Zheng Zhang , Hao Yan, Yuan Cheng, and Shuo Qiu 
Research Article (13 pages), Article ID 5532683, Volume 2021 (2021)


A Black-Box Attack Method against Machine-Learning-Based Anomaly Network Flow Detection Models

Sensen Guo , Jinxiong Zhao , Xiaoyu Li, Junhong Duan, Dejun Mu, and Xiao Jing
Research Article (13 pages), Article ID 5578335, Volume 2021 (2021)

Security Analysis of a Lightweight Identity-Based Two-Party Authenticated Key Agreement Protocol for IIoT Environments

Yuting Li , Qingfeng Cheng , and Wenbo Shi
Research Article (6 pages), Article ID 5573886, Volume 2021 (2021)

Exploring the Optimum Proactive Defense Strategy for the Power Systems from an Attack Perspective

Jinxiong Zhao , Xun Zhang, Fuqiang Di, Sensen Guo, Xiaoyu Li, Xiao Jing, Panfei Huang, and Dejun Mu
Research Article (14 pages), Article ID 6699108, Volume 2021 (2021)

Research Article

GNS: Forge High Anonymity Graph by Nonlinear Scaling Spectrum

Yong Zeng , Yixin Li , Zhongyuan Jiang, and Jianfeng Ma

School of Cyber Engineering, Xidian University, Xian 710126, China

Correspondence should be addressed to Yong Zeng; yzeng@mail.xidian.edu.cn

Received 7 May 2021; Accepted 15 September 2021; Published 30 September 2021

Academic Editor: Qing Yang

Copyright © 2021 Yong Zeng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is crucial to generate random graphs with specific structural properties from real graphs, which could anonymize graphs or generate targeted graph data sets. The state-of-the-art method called spectral graph forge (SGF) was proposed at INFOCOM 2018. This method uses a low-rank approximation of the matrix by throwing away some spectrums, which provides privacy protection after distributing graphs while ensuring data availability to a certain extent. As shown in SGF, it needs to discard at least 20% spectrum to defend against deanonymous attacks. However, the data availability will be significantly decreased after more spectrum discarding. Thus, is there a way to generate a graph that guarantees maximum spectrum and anonymity at the same time? To solve this problem, this paper proposes graph nonlinear scaling (GNS). We firmly prove that GNS can preserve all eigenvectors meanwhile providing high anonymity for the forged graph. Precisely, the GNS scales the eigenvalues of the original spectrum and constructs the forged graph with scaled eigenvalues and original eigenvectors. This approach maximizes the preservation of spectrum information to guarantee data availability. Meanwhile, it provides high robustness towards deanonymous attacks. The experimental results show that when SGF discards only 10% of the spectrum, the forged graph has high data availability. At this time, if the distance vector deanonymity algorithm is used to attack the forged graph, almost 100% of the nodes can be identified, while when achieving the same availability, only about 20% of the nodes in the forged graph obtained from GNS can be identified. Moreover, our method is better than SGF in capturing the real graph's structure in terms of modularity, the number of partitions, and average clustering.

1. Introduction

In recent years, the security and privacy of data have received extensive attention. How to protect data privacy while maintaining data availability is a hot research topic. Xiong et al. [1] propose a privacy and availability data-clustering scheme based on the k -means algorithm and differential privacy. Aldossary and Allen [2] discuss the privacy and availability of data in cloud storage and analyzes the current risks and feasible solutions. However, these methods have only been studied for data in Euclidean space. For non-Euclidean data, such as graph data, Day et al. [3] proposed a graph differential privacy method that some nodes or edges can be removed to provide the security of the query. According to [4], the deletion of a node may cause the cascading failure of multiple nodes, thus affecting the

structural characteristics of the network. Pu et al. [5] also mentioned that the deletion of edges would weaken the effectiveness of the graph data for downstream tasks. Therefore, some scholars have proposed that forge random graphs with specific properties to protect the privacy of graph data. This provides structural features of graph data while protecting its privacy. This method is widely used in the following ways: privacy-preserving social network publishing, generating specific data sets for model training, and relationship anonymity. The realization of these purposes requires that the generated graph have certain anonymity while meeting some structural metrics, maintaining the structural features.

There are many previous works done in this area, Milo et al. [6] generated a graph that only satisfies a given degree sequence. Calvert et al. [7] proposed two methods to

generate networks with targeted topology. They mainly focus on the aspect of hierarchy and locality present in the network. In [8], an algorithm that generates the synthetic graph with a target joint degree matrix is proposed. In INFOCOM 2018, Baldesi et al. [9] propose an algorithm called spectral graph forge (SGF) for generating random graphs that preserves the community structure from a real network of interest. It uses a low-rank approximation of the modularity matrix to generate forged graphs that match the target modularity within the user-selectable degree of accuracy. The modularity is also known as an essential metric of community structure, used for community detection and network analysis. As compared with two excellent previous works [10, 11], the SGF shows better performance in many metrics: modularity, partition number, degree sequence, and average clustering.

However, the low-rank approximation module in the SGF algorithm focuses more on the principal components. Its parameter α selected by the user determine how much the spectrum will be used. The closer α is to 1; the less spectrum will be used. This kind of method improves the algorithm's efficiency while keeping the general features of the original graph. Nevertheless, according to reference [12], even when a suitable local scaling affinity matrix is given, clustering cannot be obtained by the first k th eigenvectors of the matrix. Moreover, for a sparse graph, eigenvectors are more important when clustering [13]. It also can be seen from the experimental results in SGF that the clustering of the real graph can be well preserved only when α is above 0.9, that is, using more than 90% spectrum. Therefore, we can naturally find that more spectrum is needed to meet high data availability.

Meanwhile, the anonymity of the SGF algorithm decreases with the increase of α . As we analyzed, every component of the spectrum is useful. Reducing the α will inevitably lead to the loss of information. In other words, the SGF algorithm needs to make a trade-off between performance and anonymity. Therefore, is there an algorithm to maintain the original spectrum as much as possible while also providing high anonymity? Solving this problem has become the motivation of this paper.

Therefore, we propose graph nonlinear scaling (GNS). GNS scales eigenvalues of the original spectrum and leverages the scaled eigenvalue and original eigenvectors to generate a forged graph. Considering both data availability and anonymity, we list four heuristic rules to guarantee that the new spectrum has similar structural properties to the real one while has high anonymity. In summary, this nonlinear scaling method preserves all information carried by the eigenvectors without leaking the privacy of the original graph.

In the experimental part, the effectiveness of the GNS algorithm is evaluated from three aspects. The first one is the correctness of the scaling rules. And the effectiveness of different functions is compared through three metrics on a real-world data set. Results show that our scaling rules can preserve the properties of the original graph completely. The second one is the applicability of the GNS. We calculated three typical metrics on different kinds of data set, which are

modularity, number of partitions, and average clustering. Compared with the SGF, our method has better performance in clustering coefficient and modularity due to all eigenvectors are used. The last one is the anonymity ability. Publishing the graph generated by a real one usually tends to disclose the privacy information in the real graph. Attackers often use a deanonymity attack to explore the relationship between two graphs and identify the nodes. Therefore, we use the ability to resist the deanonymity attack to evaluate the algorithm's anonymity. Experimental results show that the GNS maintains the original spectrum structure and has a strong resistance to a deanonymous attack.

2. Preliminaries

In this section, we present some background knowledge on random graph generation and introduce the role of each component in the spectrum.

2.1. Notation

Graph. A graph is a pair $G = (V, E)$, where V is a set whose elements are called vertex (singular: vertex) and E is a set of paired vertices, whose elements are called edges.

Adjacency matrix. An adjacency matrix is a square matrix used to represent a finite graph. The elements of the matrix indicate whether pairs of vertices are adjacent or not in the graph. In the special case of a finite simple graph, the adjacency matrix is a (0,1) matrix with zeros on its diagonal. If the graph is undirected (i.e., all of its edges are bidirectional), the adjacency matrix is symmetric.

Spectrum. The spectrum of a matrix is the set of its eigenvalues. More generally, if $T: V \rightarrow V$ is a linear operator over any finite-dimensional vector space, its spectrum is the set of scalars λ such that $T - \lambda I$ is not invertible. The whole spectrum provides valuable information about a matrix.

Principal component. It is the dominant eigenvalue, which is the largest in absolute value. It is used in many applications, such as PageRank and PCA.

Deanonymous attack. Deanonymization attacks attempt to identify the nodes in a graph, exploiting similarities between the two graphs and potentially auxiliary information.

Algebraic growth. Algebraic growth is a change that has a constant rate.

Exponential growth. Exponential growth is a specific way that a quantity may increase over time. It occurs when the instantaneous rate of change (i.e., the derivative) of a quantity with respect to time is proportional to the quantity itself.

2.2. Role of Spectrum. When analyzing the utility of a graph for networks, the metrics usually used are degree [14], eigenvectors [15], eigenvalues [16], clustering coefficient

[17, 18], spectral radius [19], etc. The spectrum is closely related to these graph utility metrics. In this paper, we focus on the spectrum of networks. It mainly includes the eigenvalues of the adjacency matrix and Laplace matrix of the network.

There are two important eigenvalues for the analysis of spectrum, which are the largest eigenvalue λ_1 of adjacency matrix A and the second-largest eigenvalue ν_2 of Laplace matrix L . The eigenvalues of A encode information about the cycles of a network as well as its diameter [16]; the largest eigenvalue λ_1 is related to some metrics such as the maximum degree and the number of communities. In [20], the authors study how the virus propagates in real networks. In this paper, the general epidemic threshold for an arbitrary graph is proposed. It is proved that under a reasonable approximation, the epidemic threshold for a network is closely related to λ_1 . The eigenvalues of the Laplace matrix encode tree structural information about the original network. The second-largest eigenvalue ν_2 is closely related to the community partition of the original network. In [21], $1 - \nu_2$ is the lower bound of the normal cut of graphs. For the corresponding eigenvectors, it also can help acquire clusters or communities in the network. When the eigenvectors are combined with other available network statistics (e.g., node degree), they can be used to describe various network properties.

Many studies have shown that smaller eigenvalues in the spectrum are also critical. In [22], the eigenvectors associated with small eigenvalues carry smoothly varying signals, encouraging neighbour nodes to share similar values, so that small eigenvalues are referred to as low-frequency components in graphs. As pointed out by [23], the low-frequency components in graphs help preserve intracluster information in the network and reflect the local characteristics of the graph. Similarly, reference [21] proposed that for a network structure, the importance of a node group is entirely determined by the smallest eigenvalue of its Laplacian primary submatrix, which can be used to identify nodes (groups) that have a systemic effect on the whole situation and control such nodes (groups) to achieve the best overall effect at the lowest cost.

Spectrum plays a pivotal role in some popular researches at present. The PCA method is a typical method of operating on eigenvalues, which is widely used in image processing [24] and pattern recognition [25]. Karhunen–Loève transform is a method for eigenvalues in image processing, which is equivalent to the PCA method when the matrix of the K–L transform is a covariance matrix. In the field of quantum mechanics, especially in atomic physics and molecular physics, the atomic orbitals and molecular orbits in the Hartree–Fock equation can be defined as the eigenvectors of the Fock operator [26, 27]. The corresponding eigenvalues can be explained by the Koopmans theorem as the ionization potential. In the field of social networks, the computation of eigenvalues and eigenvectors is very extensive. Researchers not only can study the network properties based on the spectrum but also reconstruct the network based on the spectrum of the published networks [9].

3. Our Method

In this section, we will elaborate on our nonlinear scaling method and prove that the scaling spectrum can provide both data availability and anonymity.

3.1. How to Preserve Eigenvectors

Problem. *: For a real symmetric matrix A , we have $A \cdot X = X \cdot \Lambda$, where Λ is the diagonal matrix composed of eigenvalues, $X = (\alpha_1, \dots, \alpha_n)$, and α_i is the corresponding eigenvector of eigenvalue $\lambda_i, i = 1, \dots, n$. The goal is to generate a matrix B , which has the same eigenvectors as those of A .

Matrix A is diagonalizable because it is a real symmetry matrix. Then, we have

$$X^{-1}AX = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{pmatrix} = \Lambda. \quad (1)$$

Lemma 1 (see [28]). *Let λ be the eigenvalue of matrix A , and X is the corresponding eigenvector of λ . λ^m are the eigenvalues of A^m , and X is the corresponding eigenvector of λ^m . $A^m x = \lambda^m x$, where $m \in \mathbb{N}$.*

Proof (Mathematical induction). If λ is the eigenvalue of a matrix, X is the corresponding eigenvectors of λ . The following formula can be obtained. When $m = 1$, $A^m x = \lambda^m x$ established.

$$Ax = \lambda x. \quad (2)$$

Let $A^n x = \lambda^n x$, when $m = n$, then there is the following formula:

$$\begin{aligned} A^{n+1}x &= A(A^n)x \\ &= A(\lambda^n x) \\ &= \lambda^n (Ax) \\ &= \lambda^n (\lambda x) \\ &= \lambda^{n+1}x. \end{aligned} \quad (3)$$

From the above, it can be concluded that, for $m \in \mathbb{N}$, the proposition is established; λ^m is the eigenvalue of matrix A^m , and X is the eigenvector corresponding to λ^m . There is

$$A^m X = \Lambda^m X. \quad (4)$$

□

Theorem 1 (eigenvectors' preserving). *For a real symmetric matrix A of order n , if B is a polynomial matrix of it, matrices A and B have the same eigenvectors.*

Proof. $B = f(A)$; when B is the polynomial matrix of A , $f(x)$ is the Lagrange polynomial as follows:

$$f(x) = ax^m + bx^{m-1} + \dots + c. \quad (5)$$

The lemma shows that $A^m X = \Lambda^m X$; there is the following formula:

$$\begin{aligned} & (aA^m + bA^{m-1} + \dots + c)X \\ &= aA^m X + bA^{m-1} X + \dots + cX \\ &= a\Lambda^m X + b\Lambda^{m-1} X + \dots + cX \\ &= f(\Lambda)X. \end{aligned} \quad (6)$$

From formulas (7) and (8) is established. When B is a polynomial matrix of matrix A , it has the same eigenvectors as matrix A .

$$aA^m + bA^{m-1} + \dots + c = f(A) = B, \quad (7)$$

$$B \cdot X = f(\Lambda) \cdot X. \quad (8)$$

To sum up, the problem * can be solved as follows: firstly, the spectral decomposition of matrix A is performed, $A = X \cdot \Lambda \cdot X^{-1}$. Then the Lagrange polynomial $f(x)$ is used to scale Λ to $f(\Lambda)$, noted as Λ' . Finally, a new matrix B is constructed by the following formula. In this case, the eigenvectors of matrices B and A are the same.

$$B = X^{-1} \Lambda' X. \quad (9) \quad \square$$

3.2. How to Scale. In the last section, the correctness of the nonlinear scaling method is proved; a polynomial scaling will preserve all eigenvectors of matrix A . However, the eigenvector preserving theorem does not tell us what kind of polynomial should be used. The scaling rules will be given with specific analysis in this section to guarantee data availability and anonymity.

For a matrix A , its spectrum is a space in which the eigenvectors are the basis vectors, and the eigenvalues refer to the stretch lengths in each direction. Therefore, the Lagrange polynomials used for scale should not change the original spectrum's structural features, such as the corresponding relationship between eigenvectors and eigenvalues. Only in this case can the algorithm guarantee the data availability of the forged graph. Correspondingly, considering the privacy protection of graph data, the deanonymity attack should not be neglected. When the forgery graph is attacked, the number of identified nodes can reflect the anonymity ability of the algorithm. All of these puts forward higher requirements for the selection of polynomials in the scaling process.

Let us take a toy for example to explain in detail: Suppose the original spectrum with three base vectors $\alpha_1 = [1, 2, 3]$, $\alpha_2 = [4, -2, -0.5]$, and $\alpha_3 = [3, 5, 4]$, each vector corresponding to the stretch length of $\lambda_1 = 18$, $\lambda_2 = 7$, and $\lambda_3 = 0.6$. Our scaling goal is to ensure data availability while also maintaining high privacy.

3.2.1. Principal Keeping. For data availability, ensuring the original spectrum's feature is the first priority. More precisely, the importance of components before and after mapping should be consistent. It can be seen from Figure 1,

the graph on the left is the original spectrum with three base vectors α_1, α_2 , and α_3 . The corresponding stretch length is 18, 7, and 0.6, respectively; α_1 is the principal component; and α_3 is the smallest component. In the right graph, the base vectors remain unchanged; with stretch lengths scaled by $f(x) = \cos(x)$, they become 0.66, 0.75, and 0.82; α_3 becomes the main component while α_1 is the smallest one. It is clear that the features of the spectrums have been significantly changed. Therefore, the first rule to choose a scaling function is that the importance of each component must be guaranteed to remain unchanged.

3.2.2. Algebraic Growth. Figure 2 shows the results after mapping by three functions with different properties, all of which conform to the above order-keeping rules. Compared with $f(x) = e^x$ and $f(x) = \log(x)$, the spectrum scaled by $f(x) = 0.1x^2 + x$ is the closest to the original spectrum. The reason is that both $f(x) = e^x$ and $f(x) = \log(x)$ scale the eigenvalues in an exponential way, while Lagrange polynomial guarantees the algebraic growth of the eigenvalues so that the spectrums' features are not greatly changed. In particular, we can see that the exponential growth of $f(x) = e^x$ makes the principal component α_1 grow rapidly, leading to the neglect of other components in the whole space.

3.2.3. Nonlinear Scaling. For a deanonymity attack, the far the distance between matrices A and B , the less node will be identified. In this perspective, although the first-order polynomial is the best form to keep the algebraic growth, it only enlarges the values in equal proportion and does not change any structure features. At this point, if the adversary uses a strong deanonymity attack, most of the nodes will be identified. Therefore, the scaling function should be nonlinear. That is, the Lagrange polynomials cannot just be linear first-order functions.

3.2.4. Distribution Dependency. Adaptive parameters are necessary. The range of eigenvalues of different spectrums is very different. For the sake of getting better performance, adaptive parameters are needed to adjust the changing trend of function.

To sum up, we give four heuristic rules to select the optimal scaling function. For the toy example, we give a concrete function and analyze it. First of all, to not change the order of the original eigenvalues and make sure it scales at the algebraic level, we use $f(x) = ax^2 + bx$, which is a monotone increasing within the range of λ_i . Secondly, we can adjust a and b to make the trend of function more suitable for the current eigenvalue distribution to maintain the proportion of each component that does not change in essence. Therefore, we give the following scaling function: $f(x) = 0.05x^2 + x$ for this toy example. Figure 3 shows that the characteristics of the original space are well preserved after being scaled.

Figure 4 describes in detail how to apply the above rules to design the scaling function.

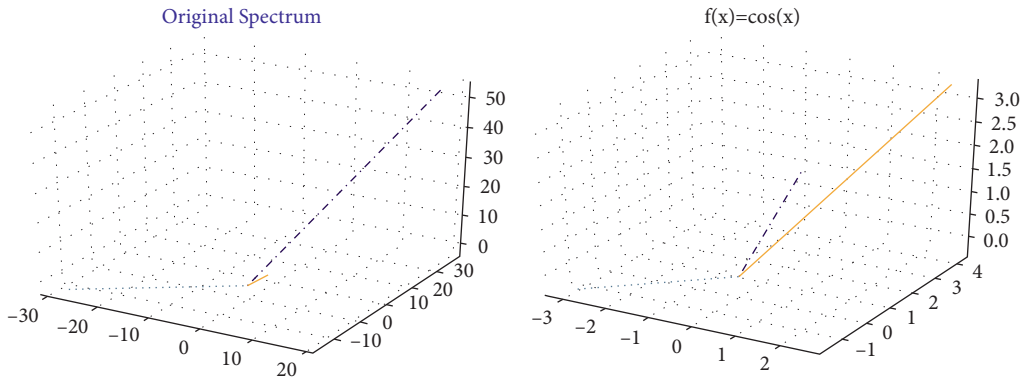


FIGURE 1: Scaling by the nonmonotonic function $f(x) = \cos(x)$.

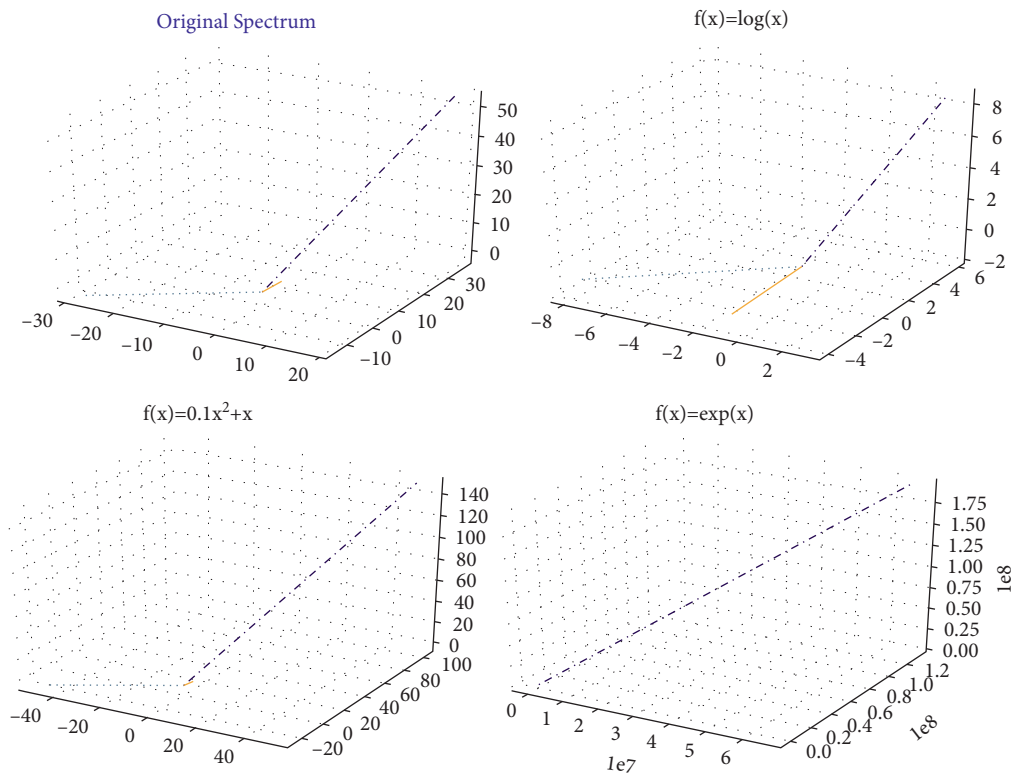


FIGURE 2: Scaling by functions of different forms.

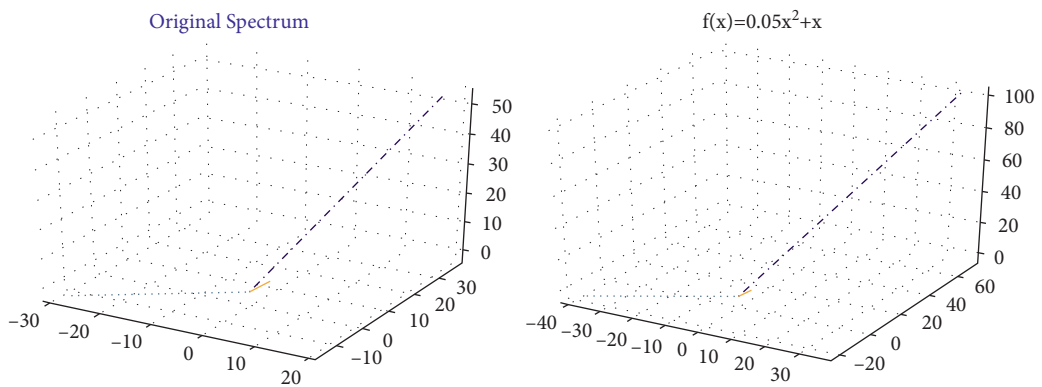


FIGURE 3: Scaling by example functions.

Algorithm 1: Design the scaling function

Input: The diagonal matrix Λ composed of eigenvalues of matrix M
Output: The suitable scaling function f for the Λ .

1. Calculate the distribution φ of Λ
2. if $\varphi > 0$ then
3. do choose a nonlinear and monotonic increasing function in this distribution
4. else if $\varphi \in \mathbb{R}$ then
5. do choose a nonlinear function which decrease in the range less than 0 and increase in the range bigger than 0.
6. while the scaled result of $f(x)$ is not similar with the original spectrum then
7. do adjust the parameters λ for f
8. Return f

FIGURE 4: Algorithm 1 (design the scaling function).

Input: The diagonal matrix Λ composed of eigenvalues of matrix M
Output: The suitable scaling function f for the Λ

- (1) Calculate the distribution φ of Λ
- (2) **if** $\varphi > 0$ **then**
- (3) **do** choose a nonlinear and monotonic increasing function in this distribution
- (4) **else if** $\varphi \in \mathbb{R}$ **then**
- (5) **do** choose a nonlinear function that decreases in the range less than 0 and increases in the range bigger than 0
- (6) **while** the scaled result of $f(x)$ is not similar to the original spectrum **then**
- (7) **do** adjust the parameters λ for f
- (8) **Return** f

ALGORITHM 1: Design the scaling function

3.3. Efficiency of the Scaling Rules. In the last section, we intuitively show the impact of different scaling functions through a toy example's spectrum image. This section will explain the effect of different functions through three metrics on a real-world data set. The first metric is modularity, which can well reflect the conservation level of the community structure. The second is the clustering coefficient, which shows the change degree of local features. At last, we measured the degree series correlation. It is a vital perspective to reflect the network topology. The scaling function for this data set designed by Algorithm 1 is $f(x) = 0.01(x + 30)^2 + 0.2x$. It perfectly keeps the importance order of each component in this spectrum while the scaling presents an algebraic growth.

For these three important indexes, we take the ratio before and after scaling as the evaluation standard. As shown in Figure 5, the best performance occurs when the scaling function is $f(x)$. Modularity ratio and average clustering ratio are away from 1 with the function's order increase. In Algorithm 1, the distribution parameters are added to adjust the change trend of the function, which makes $f(x)$ the most suitable for the spectrum. A series of rules in Algorithm 1 ensure that the spectrum before and after scaling will not change essentially.

3.4. Graph Nonlinear Scaling Algorithm (GNS). Based on the analysis above, we summarize the method in this paper as follows. As shown in Figure 6, the algorithm takes matrix M

as the input. First, spectrum decomposition on matrix M is performed, where V is the eigenvectors and Λ represents the eigenvalues. The second step is selecting the scaling function $f(x)$, which is the key to this algorithm. As shown in the figure, the scaled Λ is expressed as Λ' . And the new matrix M' is constructed by V and Λ' . This method is conducive to preserving the structural features of the real spectrum.

Next, we will further demonstrate the GNS algorithm in the form of pseudocode in Figure 7. Algorithm 2 is the overall process for obtaining the target matrix, where Algorithm 1 is used to design the scaling function.

4. Experiments

In this section, we evaluate the performance of GNS on three representative data sets. We start by describing the data sets used in this experiment, followed by the results on three metrics. And we compared the results with state-of-the-art method SGF. It shows that our method is better than SGF in those metrics with a high correlation with eigenvectors. For those metrics that have a loose connection with eigenvectors, our results are almost the same as those of the SGF method. Finally, we apply the state-of-the-art deanonymization attack: the Distance Vector attack [29], to assess the anonymity of GNS.

4.1. Evaluation Setup. Three data sets will be used in this experiment. The first one is the karate club data set, which is the classic network in the field of social networks. Second



FIGURE 5: Performance on different functions with three metrics.

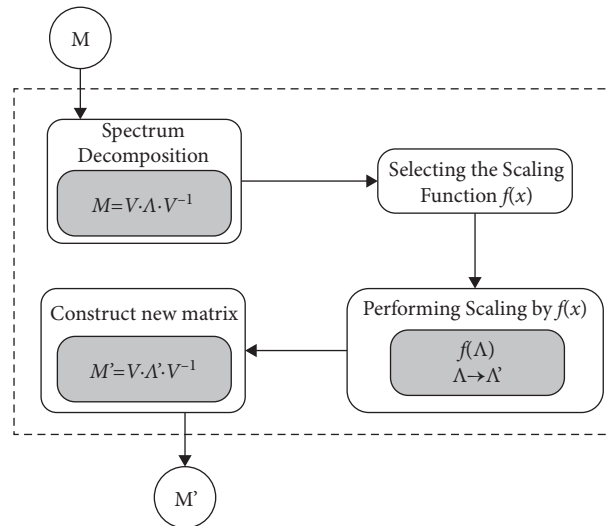


FIGURE 6: The pipeline of GNS.

Algorithm 2: Graph non-linearly scaling

- Input: The adjacency matrix of matrix M
 Output: The adjacency matrix of matrix with the same eigenvectors of M .
1. $V, \Lambda =$ similar diagonalization (M)
 2. $f(x) =$ Design the scaling function (Λ)
 3. $\Lambda' = f(\Lambda)$
 4. $M' = V \cdot \Lambda' \cdot V^{-1}$
 5. Return M'

FIGURE 7: Algorithm 2 (graph nonlinearly scaling).

and third data sets are the Facebook and Twitter ego networks, all from McAuley and Leskovec [30]. The karate data set can reflect the effect of our algorithm on the classic social network. The Facebook data set is used to be a powerful way to show that our algorithm does surpass SGF. The third data set proves the general validity of our method in a large-scale social network.

In this experiment, for a more convincing demonstration that our approach is indeed superior to the SGF algorithm, we only modify the low-rank approximation

module of SGF according to the idea of the control variable method, and the other modules are consistent. To be more specific, we also take the modularity matrix as the input of GNS and process Bernoulli samples on the output matrix to obtain the final graph.

4.1.1. Karate Club Network. This network is a social network constructed by Zachary, a sociologist, by observing an American University karate club. The network consists of 34 nodes and 78 sides. The individual node represents a club member, while the side represents the friendship between the members. The karate club network has become a classic data set in the detection of the complex network community structure.

4.1.2. Facebook Ego Network. This data set is the user data of Facebook collected from the App side, including its attributes, social circles, and ego network. There are 4,039 users and 88,234 connections. It consists of a group of ego networks derived from the Facebook social network, including

Input: The adjacency matrix of matrix M

Output: The adjacency matrix of the matrix with the same eigenvectors of M

- (1) $V, \Lambda =$ similar diagonalization (M)
- (2) $f(x) =$ **Design the scaling function** (Λ)
- (3) $\Lambda' = f(\Lambda)$
- (4) $M' = V \cdot \Lambda \cdot V^{-1}$
- (5) **Return** M'

ALGORITHM 2: Graph nonlinearly scaling

10 different networks, ranging from 52 to 1,034 nodes. In this experiment, we took the edges from all ego nets combined as a whole graph.

4.1.3. Twitter Ego Network. The Twitter data set was crawled from public sources, which includes node features (profiles), circles, and ego networks. There are 81,306 users and 1,768,149 connections. Twitter data set includes 973 ego networks. Edges from the top 100 ego networks were combined as a directed graph in this experiment.

4.2. Results on Global and Local Metrics. To verify the effectiveness of GNS, we calculated three metrics on three kinds of data sets. At the same time, the performance of the SGF algorithm is compared. The second data set was the same one used in the SGF paper. Experimental results show that our algorithm has better performance in clustering and modularity due to all eigenvectors are preserved.

4.2.1. Modularity. As the definition in [31], modularity is the fraction of the edges that fall within the given group minus the expected fraction if the edges were distributed at random. We evaluated it on three data sets. The calculation of modularity uses the famous algorithm proposed by Lefebvre et al. [32]. Consider m_0 for the modularity of the input graph and m_1 for the forged graph. m_0/m_1 is used to judge the difference of modularity before and after processing; the more it closes to 1, the graph's global features are captured better.

As shown in Figure 8, the GNS algorithm's effect is almost the same as that of SGF using 90% spectrum; the modularity ratios on these three data sets are about 1. When SGF uses only a 40% spectrum, the forged graph does not precisely keep the real graph's structural features. Its modularity ratio is relatively far away from 1 compared with the GNS.

Meanwhile, it is worth noting that SGF ($\alpha = 0.9$) keeps 90% original spectrum leading to an inevitable problem: the distinguishability between forged and original graphs is too small. In contrast, the GNS algorithm uses Lagrange polynomials to scale the eigenvalues in the original spectrum. That ensures the GNS does not directly use the original spectrum to create a new graph, guaranteeing its anonymity. This is also reflected in the subsequent deanonymization attack experiments.

4.2.2. Number of Partitions. As we targeted global features in this experiment, the number of partitions is also a necessary global metric to measure. It shows how many communities in the real graph are preserved. In Figure 9, the partition ratio is calculated by n_0/n_1 , where n_0 represents the partition number detected in the original graph and n_1 is that in the forged graph. Ideally, the value is 1. The result is similar to that of the modularity ratio. The performance of GNS on the three data sets is close to that of SGF ($\alpha = 0.9$), and both of them surpass SGF ($\alpha = 0.4$) due to more spectrum information is used.

4.2.3. Average Clustering. As a local property of networks, average clustering represents the extent of triadic closure. The average clustering ratio between the input graph and output was examined. The target value is 1. As shown in Figure 10, the experimental results are in line with our reasoning. The GNS algorithm using more spectrum information performs way better than SGF ($\alpha = 0.4$), And it is slightly better than SGF ($\alpha = 0.9$).

4.3. Deanonymity Experiment. According to the analysis, the forged graph's eigenvectors are consistent with those of the real graph, and the eigenvalues are scaled by a Lagrange polynomial. Let the real eigenvalues as the plaintext and the scaled eigenvalues as the ciphertext. The application scenario of publishing the forged graph is considered here. In this scenario, the adversary can only obtain the forged graph. Therefore, the real eigenvalue cannot be deduced.

However, the nodes may be transparent for the attacker if they use a strong deanonymity attack. As introduced, a deanonymization attack will exploit the similarities between two graphs to identify nodes with only a few seeds. In our evaluation, the state-of-the-art deanonymization attack: The distance vector (DV) attack is used. About 5% of nodes are set as the ground truth seed to feed in the DV attack. At each time, we record the percentage of the identified node to verify the privacy protection ability. The experiment results on Facebook data set are shown in Figure 11.

As shown in Figure 11, we compared the ability to resist the deanonymization attack of GNS and SGF algorithms under the same modularity ratio. $|MR - 1|$ is the absolute value between the modularity ratio and 1 where the modularity ratio is noted as MR . Based on this, we divide the values into four boxes, as shown in Figure 11, each box represents a level of data availability. For SGF, it can be seen that as the value is

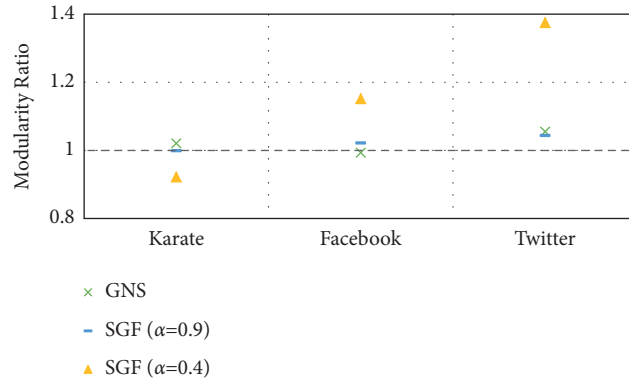


FIGURE 8: The modularity ratio on the karate, Facebook, and Twitter data sets.

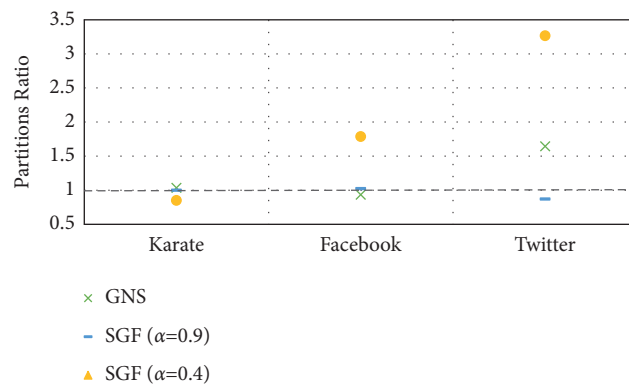


FIGURE 9: The partitions ratio on karate, Facebook, and Twitter data sets.

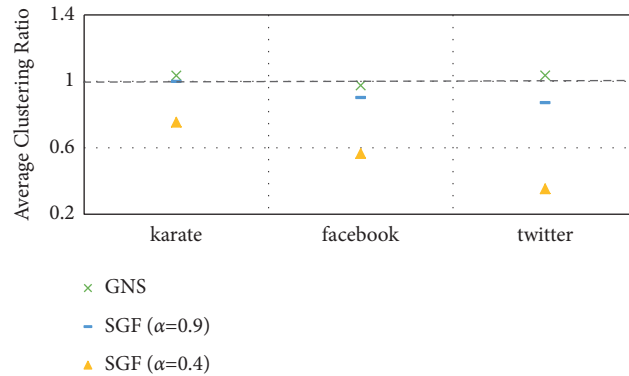


FIGURE 10: The average clustering ratio on karate, Facebook, and Twitter data sets.

closer to 0, the more global features being captured, the number of identified nodes is also increasing rapidly. Especially when the value is between 0.00 and 0.04, the percentage points are close to 100%. Only when the SGF algorithm makes a compromise by discarding part of the spectrum, the percentage of identified nodes will be below 40%, whereas its modularity is far lower than expected. The trend line of SGF presents a nearly exponential growth. On the contrary, GNS has only about 40% nodes, which are recognized when the distance is between 0.00 and 0.04. The growth trend is nearly an algebraic growth. From this, we

can conclude that compared to SGF, the GNS algorithm has a stronger ability to resist deanonymity attacks. Furthermore, high data availability can still be guaranteed. In other words, when we release the GNS-processed graph, the spectrum of the forged graph is similar to the original one. More importantly, it is difficult for the adversary to reveal too much information from the forged image, which ensures the privacy of the original graph.

Table 1 shows the performance of different methods on Facebook data set, comparing the data availability and anonymity. As shown in the table, when the modularity ratio

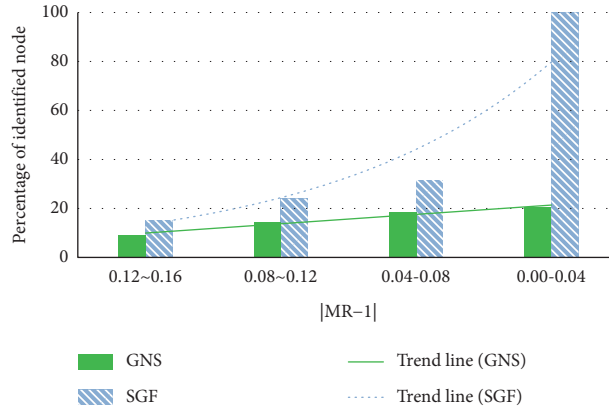


FIGURE 11: The percentage of identified nodes of SMG and SGF compared under the same modularity, MR is the modularity ratio. The distance vector attack is used.

TABLE 1: Comparison of availability and anonymity of data obtained by different methods on Facebook.

Method	Modularity ratio	Percentage of the identified node by distance vector attack (%)
GNS	0.993	20.6
SGF ($\alpha = 0.9$)	1.022	100
SGF ($\alpha = 0.4$)	1.151	15.2

approaches 1, for the GNS algorithm, only 20.6% of the nodes in the forged graph are identified and provide data availability and anonymity at the same time. When α equals 0.4, the SGF algorithm can achieve a similar degree of anonymity as GNS, while its modularity ratio is 1.151 at this time, which means the algorithm does not capture the global structure characteristics very well, the data availability is relatively low when it has high anonymity.

5. Conclusion

This paper presents a GNS algorithm for graph generation and graph anonymization. The forged graph has a similar spectrum to the original one, like a perfect stand-in of the real graph. More precisely, the new spectrum is constructed by the scaled eigenvalues and original eigenvectors. Therefore, the formed graph maintains the structure of the original graph very well. Moreover, the scaling function designed by rules guarantees that the new graph does not disclose the private information of the real one. GNS is suitable for any situation where eigenvectors are needed. This paper mainly studies the applications in the graph area. Compared with the state-of-the-art algorithm spectral graph forge (SGF), our results not only retain the data availability but also outperform the SGF algorithm in anonymity.

Data Availability

The social network 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.

References

- [1] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2018.
- [2] S. Aldossary and W. Allen, "Data security, privacy, availability and integrity in cloud computing: issues and current solutions," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 4, pp. 485–498, 2016.
- [3] W. Y. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *Proceedings of the 2016 International Conference on Management of Data*, pp. 123–138, San Francisco, CA, USA, July 2016.
- [4] C. Hong, J. Zhang, W.-B. Du, J. M. Sallan, and O. Lordan, "Cascading failures with local load redistribution in interdependent Watts-Strogatz networks," *International Journal of Modern Physics C*, vol. 27, no. 11, Article ID 1650131, 2016.
- [5] C. Pu, K. Wang, and Y. Xia, "Robustness of link prediction under network attacks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 8, pp. 1472–1476, 2019.
- [6] R. Milo, N. Kashtan, S. Itzkovitz, M. Newman, and U. Alon, "On the uniform generation of random graphs with prescribed degree sequences," 2003, <https://arxiv.org/abs/cond-mat/0312028>.
- [7] K. L. Calvert, M. B. Doar, and E. W. Zegura, "Modeling internet topology," *IEEE Communications Magazine*, vol. 35, no. 6, pp. 160–163, 1997.
- [8] M. Gjoka, B. Tillman, and A. Markopoulou, "Construction of simple graphs with a target joint degree matrix and beyond," in *Proceeding of the 2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 1553–1561, Hong Kong, China, May 2015.
- [9] L. Baldesi, C. T. Butts, and A. Markopoulou, "Spectral graph forge: graph generation targeting modularity," in *Proceeding of the IEEE INFOCOM 2018-IEEE Conference on Computer*

- Communications*, pp. 1727–1735, Honolulu, HI, USA, April 2018.
- [10] T. A. B. Snijders and K. Nowicki, “Estimation and prediction for stochastic block models for graphs with latent block structure,” *Journal of Classification*, vol. 14, no. 1, pp. 75–100, 1997.
- [11] S. Trajanovski, F. A. Kuipers, J. Martín-Hernández, and P. Van Mieghem, “Generating graphs that approach a prescribed modularity,” *Computer Communications*, vol. 36, no. 4, 2013.
- [12] W. Ye, S. Goebel, C. Plant, and C. Böhm, “FUSE: full spectral clustering,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1985–1994, San Francisco, CA, USA, August 2016.
- [13] A. Jamakovic and S. Uhlig, “On the relationship between the algebraic connectivity and graph’s robustness to node and link failures,” in *Proceeding of the 3rd EuroNGI Conference on Next Generation Internet Networks*, Trondheim, Norway, May 2007.
- [14] X. Ying and X. Wu, “Randomizing social networks: a spectrum preserving approach,” in *Proceedings of the 2008 SIAM International Conference on Data Mining*, pp. 739–750, Society for Industrial and Applied Mathematics, Atlanta, GA, USA.
- [15] K. Liu and E. Terzi, “Towards identity anonymization on graphs,” in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of data*, pp. 93–106, ACM, Vancouver, Canada, June 12 2008.
- [16] L. Wu, X. Ying, and X. Wu, “Reconstruction from randomized graph via low rank approximation,” in *Proceedings of the SIAM International Conference on Data Mining*, pp. 60–71, Society for Industrial and Applied Mathematics, Columbus, OH, USA, May 2010.
- [17] L. Zou, L. Chen, and M. T. Özsu, “k-automorphism,” *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 946–957, 2009.
- [18] M. Brautbar and M. Kearns, “A clustering coefficient network formation game,” in *Proceeding of the International Symposium on Algorithmic Game Theory*, pp. 224–235, Athens, Greece, October 2011.
- [19] M. S. Birman and M. Z. Solomjak, *Spectral Theory of Self-Adjoint Operators in Hilbert Space*, Springer Science & Business Media, Berlin, Germany, 2012.
- [20] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, “Epidemic spreading in real networks: an eigenvalue viewpoint, Reliable Distributed Systems,” in *Proceedings of the 22nd International Symposium on*, pp. 25–34, Florence, Italy, October 2003.
- [21] L. Hui, X. Xu, G. Chen, and X. Xu, “Optimizing pinning control of complex dynamical networks based on spectral properties of grounded Laplacian matrices,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 99, pp. 1–11, 2018.
- [22] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, “Learning structural node embeddings via diffusion wavelets,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1320–1329, London United Kingdom, August 23 2018.
- [23] T. Maehara, “Revisiting graph neural networks: all we have is low-pass filters,” 2019, <https://arxiv.org/abs/1905.09550>.
- [24] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision, Cengage Learning*, Springer, Berlin, Germany, 2014.
- [25] N. M. Nasrabadi, “Pattern recognition and machine learning,” *Journal of Electronic Imaging*, vol. 16, no. 4, Article ID 049901, 2007.
- [26] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure theory*, Courier Corporation, Chelmsford, MA, USA, 2012.
- [27] B. N. Khoromskij, V. Khoromskaia, and H.-J. Flad, “Numerical solution of the Hartree-Fock equation in multilevel tensor-structured format,” *SIAM Journal on Scientific Computing*, vol. 33, no. 1, pp. 45–65, 2011.
- [28] J. Steven, *Leon Linear Algebra with Applications*, Bargain Smart Plug, London, UK, 7th edition, 2006.
- [29] S. Ji, W. Li, P. Mittal, X. Hu, and R. A. Beyah, “SecGraph: a uniform and open-source evaluation system for graph data anonymization and de-anonymization,” in *Proceedings of the 24th USENIX Conference on Security Symposium*, pp. 303–318, Washington, DC, USA, August 2015.
- [30] J. J. McAuley and J. Leskovec, “Learning to discover social circles in ego networks,” *News in Physiological Sciences*, vol. 2012, pp. 548–556, 2012.
- [31] M. E. J. Newman, “Modularity and community structure in networks, 2006 APS March Meeting,” *American Physical Society*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [32] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and experiment*, vol. 2008, no. 10, Article ID P10008, 2008.

Research Article

Edge Computing Assisted an Efficient Privacy Protection Layered Data Aggregation Scheme for IIoT

Rong Ma, Tao Feng , and Junli Fang

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

Correspondence should be addressed to Tao Feng; fengt@lut.cn

Received 7 May 2021; Revised 3 August 2021; Accepted 20 August 2021; Published 29 September 2021

Academic Editor: Qing Yang

Copyright © 2021 Rong Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The emergence of edge computing has improved the real time and efficiency of the Industrial Internet of Things. In order to achieve safe and efficient data collection and application in the Industrial Internet of Things, a lot of computing and bandwidth resources are usually sacrificed. From the perspective of low computing and communication overhead, this paper proposes an efficient privacy protection layered data aggregation scheme for edge computing assisted IIoT by combining the Chinese Remainder Theorem (CRT), improved Paillier homomorphic algorithm, and hash chain technology (edge computing assisted an efficient privacy protection layered data aggregation scheme for IIoT, EE-PPDA). In EE-PPDA, first, a layered aggregation architecture based on edge computing is designed. Edge nodes and cloud are responsible for local aggregation and global aggregation, respectively, which effectively reduces the amount of data transmission. At the same time, EE-PPDA achieves data confidentiality through improved Paillier encryption, ensuring that neither attackers nor semitrusted nodes (e.g., edge nodes and clouds) can know the private data of a single device, and it can resist by simply using hash chains to resist tampering and pollution attacks ensure data integrity. Second, according to the CRT, the cloud can obtain the fine-grained aggregation results of subregions from the global aggregation results, thereby providing fine-grained data services. In addition, the EE-PPDA scheme also supports fault tolerance. Even if some IIoT devices or communication links fail, the cloud can still decrypt incomplete aggregated ciphertexts and obtain the expected aggregation results. Finally, the performance evaluation shows that the proposed EE-PPDA scheme has less calculation and communication costs.

1. Introduction

With the increasing popularity of IoT in the industrial field, IIoT, as an important application of the Internet of Things in the industry, has received more and more attention from researchers. IIoT is dedicated to interconnecting things in industrial scenarios, such as machines, sensors, and actuators [1], as well as sampling, processing, and applying real-time data in industrial environments, which promotes the conversion of traditional industries to smart industries. Since devices and sensors are usually resource-constrained, the traditional IIoT architecture integrates cloud computing models, sending all data collected by local devices to the cloud for processing and storage to reduce the computing and storage costs of local devices [2]. However, with the rapid deployment of IIoT devices, more and more data are

frequently sent to remote clouds, which not only causes huge communication costs but also brings huge processing and storage pressure to the cloud. Therefore, it is not practical to rely solely on the cloud computing model for delay-sensitive IIoT applications. In this case, the edge computing model is introduced as a supplement to cloud computing [3] to achieve efficient local data processing in IIoT; that is, user terminals can migrate their computing and storage tasks to the local edge of the network edge node [4], thereby reducing the processing pressure on the cloud, realizing low-latency data processing, and significantly reducing communication overhead.

In IIoT, large amounts of perception data collected by industrial equipment and regularly transmitted to the cloud usually contain sensitive information [5, 6]. Therefore, in recent years, reducing the amount of transmitted data and

protecting the privacy and security of the data have attracted a lot of attention. Data aggregation is seen as an effective method to reduce communication overhead and protect data privacy. For example, edge nodes can perform aggregation operations on the received data and then deliver a single aggregation result to the cloud, thereby significantly reducing the amount of data transmission, and the data privacy of a single device is leaked [7]. Although data aggregation can achieve a great performance improvement, the aggregation operation is usually performed by an untrusted third party, so privacy and security (confidentiality and integrity) are still threatened. For example, curious entities (such as edge nodes and clouds) can observe private content in received data packets.

In order to provide fine-grained data services on the cloud while protecting data privacy, confidentiality, and integrity, this paper proposes an efficient privacy protection layered data aggregation scheme for edge computing assisted IIoT. The main contributions are summarized in the following points:

- (1) The first major contribution is the design of a layered aggregation architecture based on edge computing, which enables data aggregation to be implemented on the local edge nodes and the cloud separately, which significantly reduces the amount of data transferred from the edge nodes to the cloud.
- (2) The second contribution is that edge nodes use a simple hash chain mechanism to resist tampering and pollution attacks, while also preventing the leakage of individual device privacy information at semitrusted nodes and resisting eavesdropping attacks on all communication links in the IIoT.
- (3) The third contribution is that the cloud can recover the aggregate results of all subregions and the entire region from a single global aggregated ciphertext to support fine-grained data services. At the same time, when the IIoT device or transmission channel fails, the cloud can still decrypt the aggregated ciphertext smoothly; that is, the proposed scheme supports fault tolerance.

This remainder of the article is organized as follows: Section 2 covers the work of the edge computing and data aggregation scheme for IIoT. The system model and adversary model of the proposed privacy protection data aggregation scheme are described in Section 3. In Section 4, we describe the efficient privacy protection layered data aggregation scheme. Section 5 analyzes the proposed program in terms of safety and performance, respectively. Section 6 summarizes the full text.

2. Related Work

Recently, many methods to protect cloud/edge system data security have been proposed, such as certificateless signature [8] and blockchain [9]. There are also many schemes that use homomorphic encryption to achieve secure data aggregation [10]. For example, Lu et al. [11] designed an efficient and

privacy-protected aggregation scheme in the smart grid. The scheme uses a super-increasing sequence to integrate multidimensional data into a one-dimensional form and then uses the Paillier algorithm to aggregate the encrypted data. This reduction significantly improves communication efficiency and better meets the real-time requirements of communication. Chen et al. [12] introduced a novel multifunctional data aggregation scheme that allows the gateway to perform multifunctional aggregation, and the control center can calculate various statistical information (variance, one-way analysis of variance, etc.) in a privacy-protected manner and be flexible and provide diversified services locally. At the same time, by increasing the acceptable noise to resist the differential attack [13], Li et al. [14] constructed an effective privacy protection demand response scheme. By combining homomorphic encryption and key update technology, the solution can provide privacy protection, confidentiality, and key update functions. In addition, Li et al. [15] proposed a privacy protection dual-function aggregation scheme based on lattice encryption technology. The data control center in the smart grid can calculate the mean and variance of all users' power consumption and protect user privacy to prevent eavesdropping. Wang et al. [16] designed an anonymous aggregation scheme for edge-assisted cloud computing systems. This scheme reduces bandwidth consumption by using intermediate fog nodes to perform homomorphic aggregation and protects identity privacy through anonymity mechanisms. However, the above solutions can only achieve privacy protection against external attackers and cannot prevent privacy leakage caused by internal threats. For example, a semitrusted or compromised cloud control center can obtain individual device data.

In order to overcome the above shortcomings, in literature [17], the authors designed a privacy-protected data aggregation scheme based on untrusted aggregators, which enables each user to encrypt data with different keys to prevent the aggregator from infringing on data privacy. In addition, the scheme also uses differential privacy technology to resist differential attacks. Ni et al. [18] proposed a security-enhanced data aggregation scheme based on Paillier encryption, in which a trapdoor hash function is used to implement data authentication to protect the confidentiality and integrity of data and prevent malicious aggregation. In addition, Chen et al. [19] designed a fault-tolerant data aggregation scheme using homomorphic Paillier encryption. This solution can protect personal user data from attacks from gateways, control centers, and powerful attackers that can destroy the control center, while supporting fault tolerance. Kamil et al. [20] designed a privacy aggregation scheme suitable for smart grids based on the elliptic curve encryption algorithm, which can not only safely resist internal attacks but also solve a series of security challenges. Zhang et al. [21] proposed a novel space-time aggregation scheme, in which the time dimension aggregation is performed on the user side, and the gateway is responsible for the spatial aggregation of the entire community. This scheme realizes privacy protection by resisting internal and external collusion attacks. However, the above solutions can only

provide a global aggregation result for the control center and cannot meet the more fine-grained requirements of the cloud. For example, the cloud needs to know the aggregation results of multiple specific subregions.

In order to solve the above problems, Lu et al. [22] proposed a novel privacy protection subset aggregation scheme to meet the needs of the control center to obtain more fine-grained aggregation results. This scheme divides the entire user residence into two subsets according to the set threshold and then obtains the total energy consumption and the number of users in each subset by using the composite order group. At the same time, the data privacy of individual users is protected at the curious gateway and control center. Lu et al. extended the work in [22] to support data integrity authentication and proposed a subset aggregation scheme based on data integrity [23]. This scheme is based on a novel hash chain construction mechanism to complete the verification of the integrity of the aggregated data. Literature [24] proposed a privacy-protected multi-subset data aggregation scheme, which can protect the privacy of users while calculating the number of users and summarizing the total power consumption of each subset. However, this scheme lacks a verification mechanism to ensure the integrity of the received data and does not support fault tolerance. In addition, Knirsch et al. [25] proposed a fault-tolerant and efficient scheme to aggregate data on different groups. The solution is based on CRT, Shamir's secret sharing, and Paillier algorithm to formulate a novel aggregation protocol to support efficient and fault-tolerant group aggregation with privacy protection, as well as the dynamic joining and leaving of households. However, this solution is not fault-tolerant. When any smart meter fails, it will not be able to recover the global aggregation result. At the same time, both literatures [25] lack a data integrity authentication scheme.

The above schemes can all produce certain privacy protection data aggregation effects, but there still remain the following unresolved problems: (1) Data aggregation operations are usually performed by untrusted third parties, so there are privacy and security risks. While resisting external attackers, we also need to guard against internal attackers. (2) The cloud can recover the aggregate results of all subregions and the entire region from a single global aggregated ciphertext to support fine-grained data services. (3) When the IIoT device or transmission channel fails, the cloud can still decrypt the aggregated ciphertext smoothly; that is, the proposed scheme supports fault tolerance.

3. Problem Description

3.1. System Model. In the IIoT network scenario based on edge computing, a layered data aggregation system model is constructed. The model includes three layers, sensing layer, edge layer, and cloud layer, and mainly includes five entities: IIoT device, edge node (EN), Industrial Cloud (IC), Trusted Management Authority (TMA), and user. The detailed relationship between these entities is shown in Figure 1.

In the sensing layer, IIoT devices are divided into multiple subareas based on geographical distribution. Each

IIoT device has sensing, processing, and communication functions and is regarded as a data source. Their main responsibility is to collect sensing data in designated areas in real time and periodically forward their encrypted data to the industrial cloud through edge nodes. The purpose is to monitor specific areas and protect the privacy of sensitive data at the same time.

In edge layer, each subarea is managed by an adjacent edge node, and the edge node is an intermediate device between the IIoT device and the cloud. The edge node is mainly responsible for two tasks. The first task is data authentication: when the edge node receives sensing data from the IIoT device, in order to ensure the authenticity and integrity of the data, the edge node will perform authentication operations on the received data. If the received data has not been tampered with or is not contaminated data injected by an active attacker, the edge node will accept the data; otherwise, it will be deleted. The second task is data aggregation: after the edge node authenticates all the received data, it aggregates all the valid encrypted data into a number and generates a local report to send to the industrial cloud, which greatly reduces the amount of communication between edge nodes and the cloud while reducing the processing burden on the cloud.

The cloud layer contains an IC as the data management center of the system. IC is responsible for collecting data of all IIoT devices forwarded through edge nodes and performing global aggregation operations on the received local aggregated data to track aggregate statistics at any time. At the same time, IC can provide fine-grained services, that is, provide users with statistical information of designated subregions or global regions when they receive their requests.

As regards the user, for legitimate users, if they need to know the statistics of a specific subregion or global region, they can send a request to the cloud. Subsequently, according to the requirements in the user's request, the cloud returns the corresponding statistical information to the user.

Regarding TMA, in EE-PPDA, it is assumed that a fully trusted TMA only participates in the system initialization phase, and its responsibility is to initialize system parameters and keys and publish public parameters and key distribution to IIoT devices, edge nodes, and ICs.

3.2. Adversary Model. This article is mainly concerned with the security, integrity, and privacy protection in the process of data generation and transmission. Assuming that the industrial cloud and edge nodes in the network model are both "honest and curious" entities, this means that they honestly implement security protocols but at the same time remain curious about the device's sensing data.

In our adversary model, we consider a strong attacker *A* whose goal is to perceive as much of the user's personal privacy data as possible. "Strong" means that attacker *A* not only can listen to all the communication data in our system model but also can initiate the following attacks:

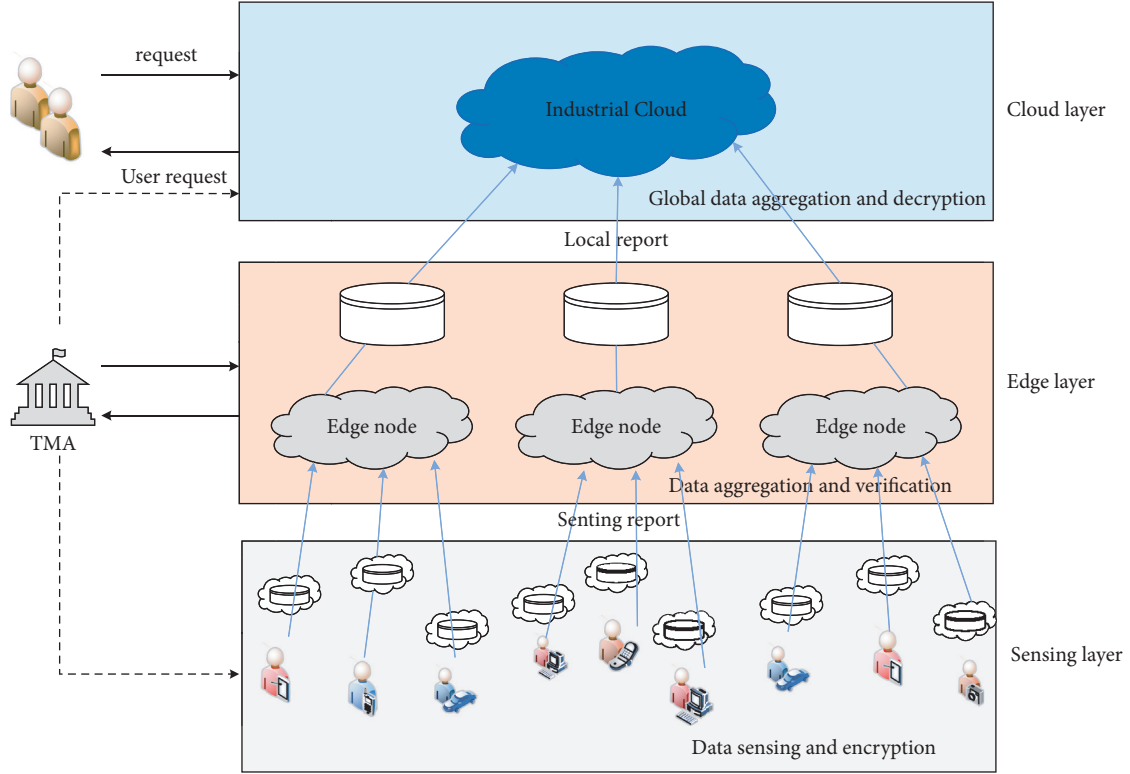


FIGURE 1: System model.

A may tamper with the transmitted data for malicious purposes or directly inject contaminated data. Therefore, the intermediate edge node should have the ability to detect and delete erroneous data locally.

A can eavesdrop on all communication channels to steal the transmitted perception data, which will lead to the leakage of private information.

In addition, a practical application scenario is also considered; that is, there is an IIoT device or a communication channel failure, which may cause the cloud to fail to decrypt the received aggregated ciphertext.

4. Efficient Privacy Protection Layered Data Aggregation Scheme

This section proposes an efficient privacy protection layered data aggregation scheme for IIoT. This scheme integrates the concept of layered aggregation, improved Paillier encryption, the Chinese remainder theorem, and hash chain technology to achieve efficient and fine-grained aggregation statistics decryption without exposing personal privacy and low-cost integrity authentication. The scheme mainly includes four parts: system initialization, data collection and encryption, local data processing, and global data aggregation and decryption. The details are as follows.

4.1. System Initialization. First, set two security parameters (μ, l) in the IIoT system, and then TMA randomly selects two large prime numbers Q_1 and Q_2 ; $|Q_1| = |Q_2| = \mu$. At the same time, calculate the public and private keys of

homomorphic Paillier encryption ($N = Q_1Q_2$, $g = 1 + N$), and define a function as $L(x) = x - 1/N$. Assuming that there are k subregions in the sensing layer and n sensing devices in each subregion, TMA selects k relatively prime positive integers p_1, p_2, \dots, p_k , $|p_i| = l$ to calculate coefficient a_i of each subregion. The process is as follows:

$$\left\{ \begin{array}{l} P = \prod_{i=1}^k p_i, \\ P_i = \frac{P}{p_i}, \\ T_i \equiv P_i^{-1} \pmod{p_i}, \\ a_i = T_i \cdot P_i. \end{array} \right. \quad (1)$$

Subsequently, TMA uses a pseudorandom number generator to generate kn uncorrelated random numbers $\{s_{11}, \dots, s_{1n}, \dots, s_{k1}, \dots, s_{kn}\}$, which are assigned to corresponding sensing devices as private keys. At the same time, the private key s_0 of the industrial cloud (IC) is calculated according to the following equation and sent to the IC:

$$s_0 + \sum_{i=1}^k \sum_{j=1}^n s_{ij} \equiv 0 \pmod{\lambda}. \quad (2)$$

In addition, generate a set of pseudorandom numbers $\{I_{11}, \dots, I_{1n}, \dots, I_{k1}, \dots, I_{kn}\}$ to construct a set of hash

chain heads $\{H_{11,0}, \dots, H_{1n,0}, \dots, H_{k1,0}, \dots, H_{kn,0}\}$, and each hash head is attached with a TMA signature σ . Then it is sent to the corresponding IIoT sensing devices and edge nodes. In addition, TMA selects a cyclic group G and two secure encryption hash functions: $h: \{0, 1\}^* \rightarrow Z_N^*$ and $H: \{0, 1\}^* \rightarrow G$. Finally, TMA chooses a random number k_i as the shared key between the edge node edge_i and IC and publishes the system public parameters $\{G, L(x), N, a_i, p_i: i = 1, 2, \dots, k, h, H\}$.

4.2. Data Collection and Encryption.

- (1) Collection of industrial data: Each IIoT sensing device continuously collects real-time sensing data and periodically sends the collected data to the IC through the edge node. Suppose that there are k subregions A_i in the sensing layer, satisfying the condition $A_i \cap A_j = \emptyset, i = 1, 2, \dots, k, i \neq j$. Each subarea A_i is governed by an adjacent edge node

$$h(t_\tau)^{s_{ij}} \in Z_N^*,$$

$$m'_{ij,\tau} = m_{ij,\tau} \cdot a_i,$$

$$c_{ij,\tau} = g^{m_{ij,\tau'}} \cdot r^N \bmod N^2 = (1 + N)^{m_{ij,\tau'}} \cdot h(t_\tau)^{s_{ij} \cdot N} \bmod N^2$$

$$(1 + N)^m = 1 + \sum_{i=1}^m \binom{m}{i} N^i \rightarrow (1 + N \cdot m_{ij,\tau'}) \cdot h(t_\tau)^{s_{ij} \cdot N} \bmod N^2. \quad (3)$$

In addition, in order to provide evidence of the integrity of the received data at the edge node to ensure that the data has not been tampered with or contaminated by an attacker, a hash chain with one-way characteristics is used to calculate the current hash chain value $H_{ij,\tau}$ of the ciphertext $c_{ij,\tau}$:

$$H_{ij,\tau} = H(c_{ij,\tau}) \oplus H_{ij,\tau-1}. \quad (4)$$

Finally, the encrypted sensing report $(c_{ij,\tau}, H_{ij,\tau})$ is sent to the upper edge node edge_i , waiting for further aggregation processing.

4.3. Local Data Processing.

- (1) When the edge node edge_i receives the encrypted sensing report $(c_{ij,\tau}, H_{ij,\tau})$ sent by all the sensing devices in the subarea under its jurisdiction in the time slot t_τ , it first passes the hash chain value $H_{ij,\tau}$ in the inspection report. The correctness of hash chain value verifies the integrity of all received data in turn. The specific process is as follows: edge_i calculates the hash chain value $H'_{ij,\tau} = H(c_{ij,\tau}) \oplus H_{ij,\tau-1}$ for verification based on the ciphertext $c_{ij,\tau}$ and checks whether the equation $H'_{ij,\tau} = H_{ij,\tau}$ holds. If it is true, the verification is passed, and edge_i receives $c_{ij,\tau}$ and stores $H_{ij,\tau}$ for the next integrity verification.

edge_i and contains n IIoT sensing devices $IID_{ij}, j = 1, 2, \dots, n$. At the same time, we assume that the reporting period of the IIoT sensing device is $\Gamma = \{t_1, t_1, \dots, t_{\text{MAX}}\}$, and the raw perception data collected by IID_{ij} at time $t_\tau \in \Gamma$ is denoted as $m_{ij,\tau} \in Z_N$.

- (2) Sensing data encryption: Because the data collected by each IIoT sensing device always contains sensitive and private information, and there are active attackers and eavesdroppers in the communication channel between the sensing device and edge nodes, in order to prevent the privacy data of individual sensing devices from being contaminated or eavesdropped by attackers, each sensing device IID_{ij} needs to perform the following encryption operations to obtain its ciphertext $c_{ij,\tau}$ before forwarding its data $m_{ij,\tau}$ to the upper edge node:

- (2) When all verified ciphertexts $c_{ij,\tau}, j = 1, 2, \dots, n$, are obtained, edge_i uses the additive homomorphism of Paillier encrypted ciphertexts to aggregate all ciphertexts without decryption. Get the aggregation result $C_{i,\tau}$ of subregion A_i under jurisdiction:

$$C_{i,\tau} = \prod_{j=1}^n c_{ij,\tau} = \left(1 + N \cdot \sum_{j=1}^n m'_{ij,\tau}\right) \cdot h(t_\tau)^{N \cdot \sum_{j=1}^n s_{ij}} \bmod N^2. \quad (5)$$

- (3) In order to ensure the integrity of the aggregated ciphertext $C_{i,\tau}$ of the subarea, edge_i calculates the verification code $H_{i,\tau} = H(C_{i,\tau} \| k_i)$ through the shared secret key k_i with the IC and provides verification evidence for the IC. Finally, edge_i sends its local report $(C_{i,\tau}, H_{i,\tau})$ to the IC.

4.4. Global Data Aggregation and Decryption.

- (1) After the cloud center receives the local reports $(C_{i,\tau}, H_{i,\tau}), 1 \leq i \leq k$, of k edge nodes, it first verifies the integrity of the aggregated ciphertext $C_{i,\tau}$ of all subregions in turn. The specific process is as follows: IC based on the previous one Hash chain value $H_{i,\tau-1}$ calculates $H'_{i,\tau} = H(C_{i,\tau}) \oplus H_{i,\tau-1}$ to verify whether the equation $H'_{i,\tau} = H_{i,\tau}$ is correct. If the equation is

correct, the verification is passed and the IC accepts $C_{i,\tau}$.

- (2) In order to simplify the key management of the IC while enhancing the privacy protection of the individual perception device data, the system only allocates a unique key s_0 to the IC, so that the IC cannot directly decrypt the aggregated ciphertext of each subarea. In order to restore the aggregated statistical values of the desired subregion, IC must first aggregate all subregions aggregated ciphertext through the following calculation to obtain a global aggregation result C_τ :

$$C_\tau = \prod_{i=1}^k c_{i,\tau} = \left(1 + N \cdot \sum_{i=1}^k \sum_{j=1}^n m'_{ij,\tau} \right) \cdot h(t_\tau)^{N \cdot \left(\sum_{i=1}^k \sum_{j=1}^n s_{ij} \right)} \bmod N^2. \quad (6)$$

Next, IC can decrypt and obtain the statistical value of each subarea and the global statistical value (e.g., the sum and the average value) by performing the following steps.

Step 1: IC uses its key s_0 to eliminate the term containing $h(t_\tau)$ in the expression of C_τ and obtain value B after simplification:

$$\begin{aligned} B &= C_\tau \cdot h(t_\tau)^{s_0}, \\ &= (1 + N)^{\sum_{i=1}^k \sum_{j=1}^n m'_{ij,\tau}} \cdot h(t_\tau)^{N \cdot \left(\sum_{i=1}^k \sum_{j=1}^n s_{ij} + s_0 \right)} \bmod N^2 \\ &\xrightarrow{s_0 + \sum_{i=1}^k \sum_{j=1}^n s_{ij} \equiv 0 \bmod \lambda \Rightarrow s_0 + \sum_{i=1}^k \sum_{j=1}^n s_{ij} = \phi \lambda, \text{ where } \phi \in \mathbb{Z}_N^*} \\ &= (1 + N)^{\sum_{i=1}^k \sum_{j=1}^n m'_{ij,\tau}} \cdot h(t_\tau)^{N \cdot \phi \lambda} \bmod N^2 \\ &= (1 + N)^{\sum_{i=1}^k \sum_{j=1}^n m'_{ij,\tau}} \bmod N^2 \\ &= \left(1 + N \cdot \sum_{i=1}^k \sum_{j=1}^n m'_{ij,\tau} \right) \bmod N^2 \end{aligned} \quad (7)$$

Step 2: According to value B , IC can decrypt to obtain a pseudoglobal aggregate value W :

$$\begin{aligned} W &= \frac{(A-1)}{N \bmod N^2}, \\ &= \sum_{i=1}^k \sum_{j=1}^n m'_{ij,\tau} m_{ij,\tau} \quad (8) \\ &= \sum_{i=1}^k a_i \sum_{j=1}^n \end{aligned}$$

Step 3: In order to obtain the total aggregation result of the global area, IC first needs to calculate the aggregation statistics of each subarea. Based on the known system parameters p_i , $i = 1, 2, \dots, k$, IC can obtain the statistics and $D_{i,\tau}$ of each subarea through the Chinese remainder theorem:

$$\begin{aligned} D'_\tau &= W \bmod P, \\ D_{i,\tau} &= \sum_{j=1}^n m_{ij,\tau} = D'_\tau \bmod p_i. \end{aligned} \quad (9)$$

At the same time, the corresponding mean value $E_{i,\tau}$ of each subregion can also be obtained:

$$E_{i,\tau} = \frac{D_{i,\tau}}{n}. \quad (10)$$

Finally, the global statistics sum D_τ and the corresponding mean value E_τ of k subregions can be obtained:

$$\begin{aligned} D_\tau &= \sum_{i=1}^k D_{i,\tau}, \\ E_\tau &= \frac{D_\tau}{kn}. \end{aligned} \quad (11)$$

4.5. Fault Tolerance. Consider a practical scenario. Some devices in a subarea fail at a certain point in time, and the edge node cannot receive its report, causing the edge node and the cloud to receive incomplete aggregation results. Since the cloud only has one key s_0 , obtaining incomplete aggregated ciphertext will cause the above-mentioned decryption process to fail to be successfully performed, and the cloud will not be able to correctly decrypt the aggregated ciphertext.

Since each edge node holds n hash chains, these hash chains are used to verify the sensing reports of n different devices at different points in time, so edge nodes can find damage by inspecting unverified hash chain devices. Let $A'_t \subset A_t$ denote the collection of faulty equipment, and let C'_t denote the incomplete aggregation result received by edge $_i$ at time t_τ . In order to obtain information $h'(t_\tau)$ related to the devices in the fault set A'_t , edge $_i$ sends a loss report (A'_t, t_τ) to the TMA. Since the TMA manages the keys of all devices, the report is received (A'_t, t_τ) , and TMA can use the private key of the device involved in A'_t to calculate $h'(t_\tau)$:

$$h'(t_\tau) = h(t_\tau)^{\sum_{IID_{ij} \in A'_t} s_{ij}}. \quad (12)$$

The missing information is returned $h'(t_\tau)$ to edge $_i$. After receiving $h'(t_\tau)$, edge $_i$ combines it with C'_t to obtain the decryptable ciphertext C_τ through the following calculation:

$$C_\tau = C'_t \cdot h'(t_\tau) = \left(1 + N \cdot \sum_{IID_{ij} \in A_t/A'_t} m'_{ij,\tau} \right) \text{mod} N^2. \quad (13)$$

Then, according to equations (8)–(12), the cloud can still decrypt the incomplete aggregate ciphertext and obtain the expected aggregate statistical value.

5. Security and Performance Evaluation

5.1. Security Analysis. According to the attacker model defined in the problem description, this section will evaluate the privacy, confidentiality, and integrity of the device-sensing data.

5.1.1. Confidentiality and Privacy. For confidentiality, the ciphertext form of the sensing data $m_{ij,\tau}$ of each device IID_{ij} is $c_{ij,\tau} = (1 + N \cdot m'_{ij,\tau}) \cdot h(t_\tau)^{s_{ij} \cdot N} \text{mod} N^2$. If $h(t_\tau)^{s_{ij}}$ is regarded as a random number, the converted ciphertext form $c_{ij,\tau}$ can regard $c_{ij,\tau} = (1 + N \cdot m'_{ij,\tau}) \cdot r_{ij}^N \text{mod} N^2$ as the encryption result of the Paillier algorithm. Similarly, the aggregation result of subarea A_t and global area A is also a valid Paillier encryption result. Since the Paillier encryption algorithm is semantically safe against selective plaintext attacks [26], EE-PPDA can resist eavesdropping attacks and ensure the confidentiality of the original sensing data and aggregated results. At the same time, except that the authorized IC can successfully decrypt the aggregation results of each subarea and the entire area, other unauthorized entities (such as edge nodes) cannot obtain the plaintext of the aggregation results.

For privacy, neither semitrusted aggregators (edge nodes and cloud) nor eavesdroppers can obtain the perception data of a single device. When a semitrusted edge node receives all perception reports from its subarea, it will not be possible for the edge node to recover any perception data of any IIoT device because it cannot obtain the decryption private key. After all the ciphertexts are aggregated, because the aggregated result is semantically secure, the edge node still cannot infer any real information from the encrypted aggregated result. For a semitrusted IC, although it can use its private key s_0 to decrypt and read the aggregated plaintext of each subarea, it cannot observe the sensing data of a single device from the aggregated plaintext. In addition, based on the above confidentiality analysis, even if an eavesdropper can obtain the ciphertext transmitted on all communication links, it still cannot infer the original sensing data of a single IIoT device. Summarizing the above analysis results, it can be concluded that the proposed EE-PPDA scheme protects the privacy of the original data of a single IIoT device.

5.1.2. Integrity. In the transmission link between the IIoT device and the edge node, an attacker may tamper with the transmitted data or directly inject polluted data. In order to ensure the validity of the data received in the edge node, the hash chain technology is used on the edge node to achieve integrity authentication. At each transmission time point, the sensing report $(c_{ij,\tau}, H_{ij,\tau})$ of each IIoT device contains a new hash chain value, which can be calculated, where it is the previous hash chain value. Based on the one-way characteristic of the hash chain, it is not feasible for an attacker to obtain from it, so it is difficult for an attacker to launch a successful tampering attack. When the edge node receives the sensing report, if it is verified in the previous time period, it can effectively verify the integrity of the data through calculation. If it is equal, it means that it has not been tampered with or is not the tainted data injected during the communication. Therefore, EE-PPDA can effectively protect data integrity to resist malicious attacks by attackers.

5.2. Performance Evaluation. This section will evaluate the proposed EE-PPDA scheme from two aspects: the computing overhead of IIoT devices, edge nodes, and ICs and the amount of data transmission. IT is compared with three other schemes: the SEDA scheme proposed in [18], the LPDA-EC scheme in [27], and the APPA scheme in [28]. These three schemes all use the standard Paillier algorithm, and the ciphertext form is $c = g^m \cdot r^N \text{mod} N^2$. The simulation experiment runs on a computer configured with Intel Core i5-8250U@1.60 GHz CPU, 8 G RAM.

5.2.1. Computational Overhead. Let the symbols C_E , C_M , C_H , C_{XOR} , C_e , C_p , and C_m denote an exponential operation on $Z_{N^2}^*$, a $Z_{N^2}^*$ multiplication operation, a hash operation, an XOR operation, and an exponential operation on the cyclic group G bilinear pairing and multiplication on G , respectively. As compared with the time-consuming bilinear pairing C_p operation, the calculation time of C_M , C_H , and

TABLE 1: Computational complexity comparison.

	EE-PPDA	SEDA [18]	LPDA-EC [27]	APPA [28]
IIoT device	$C_E + C_M + C_H + C_{XOR}$	$2C_E + 3C_e$	$2C_E + 3C_m$	$2C_E + C_M + C_H$
EN	$(n-1)C_M + nC_{XOR} + (n+1)C_H$	$(n-1)C_M + (n+1)C_m$	$(n-1)C_M + 4C_e$	$(n+2)(C_M + C_H)$
IC	$C_H + C_E$	$2C_P + 2C_E + (n+2)C_e$	$2C_P + 2C_E$	$C_E + C_M + C_H$

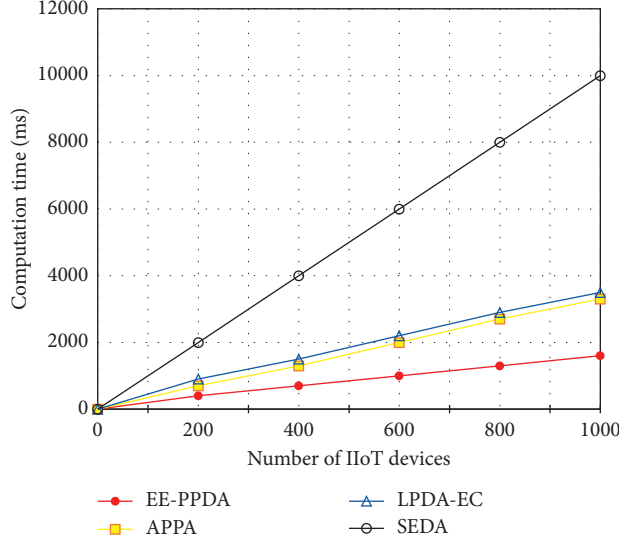


FIGURE 2: Computational cost comparison.

C_{XOR} and the operation time after decryption are negligible, so the computational overhead caused by these operations can be ignored. Based on the MIRACL and PBC libraries, an experiment was carried out to estimate the time cost of each operation, in which parameters μ and G were set to 512 bits and 160 bits, respectively. From the perspective of computational complexity, bilinear pairing operations have the highest computational complexity among these operations, followed by exponentiation and multiplication. Our experimental results also confirm this conclusion. The final experimental results show that the calculation time of C_P is close to 18.0 ms, C_e and C_E are about 1.70 ms and 1.60 ms, respectively, and smallest C_m is close to 0.07 ms.

In Table 1, the computational overheads of the four schemes at IIoT devices, edge nodes, and IC are listed in detail. In EE-PPDA, the calculation required for an IIoT device IID_{ij} to generate a perception report $(c_{ij,\tau}, H_{ij,\tau})$ is $C_E + C_M + C_H + C_{XOR}$, and C_E occupies the largest computational cost. Therefore, compared to the amount of calculation required by the other three schemes, $2C_E + 3C_e$, $2C_E + 3C_m$, and $2C_E + C_M + C_H$, EE-PPDA reduces the computational overhead by nearly half on the device side.

At edge nodes, if low-calculation operations (such as authentication of a single ciphertext) are ignored, the EE-PPDA, SEDA, and LPDA-EC schemes only need to perform $(n-1)C_M$ operations with a small amount of calculation. It can aggregate n ciphertexts, and the APPA scheme requires $(n+1)$ times. Due to the low time-consuming operation of C_M , it can be said that the computational costs of these four schemes at edge nodes are almost the same. At the IC, the

EE-PPDA scheme only needs $C_H + C_E$ operations to verify the received reports and decrypt the aggregation results, which is slightly less than the $C_E + C_M + C_H$ operations required in the APPA scheme. However, the SEDA and LPDA-EC schemes require $2C_P + 2C_E + (n+2)C_e$ and $2C_P + 2C_E$ operations, respectively, both of which include time-consuming C_P operations. As we all know, the computational cost of C_P is significantly higher than operating C_E . Therefore, the EE-PPDA scheme greatly reduces the computational cost of the IC. Combining the above analysis results, it can be concluded that the proposed EE-PPDA scheme achieves lightweight security and privacy protection.

In order to compare the calculation cost more intuitively, the execution time of the above mechanism is calculated, and the curve of the total calculation time as the number of IIoT devices increases is depicted in Figure 2. Obviously, compared with the other three schemes, the proposed EE-PPDA scheme significantly reduces the calculation time. Especially when more IIoT devices are added, more calculations will be saved by the EE-PPDA scheme.

5.2.2. Data Transfer Volume. In the EE-PPDA scheme, data transmission includes two parts: device-to-edge communication (device-to-EN) and edge-to-IC (EN-to-IC) communication. In the device-to-EN phase, the IIoT device sends its sensing report $(c_{ij,\tau}, H_{ij,\tau})$ to the upper edge node $edge_i$, and the size of the report is $S_{ij} = 2048 + 160$ bits. Therefore, the total amount of data transmission during device-to-edge communication is $S_{DF} = n \cdot S_{ij}$ bits. Next, in

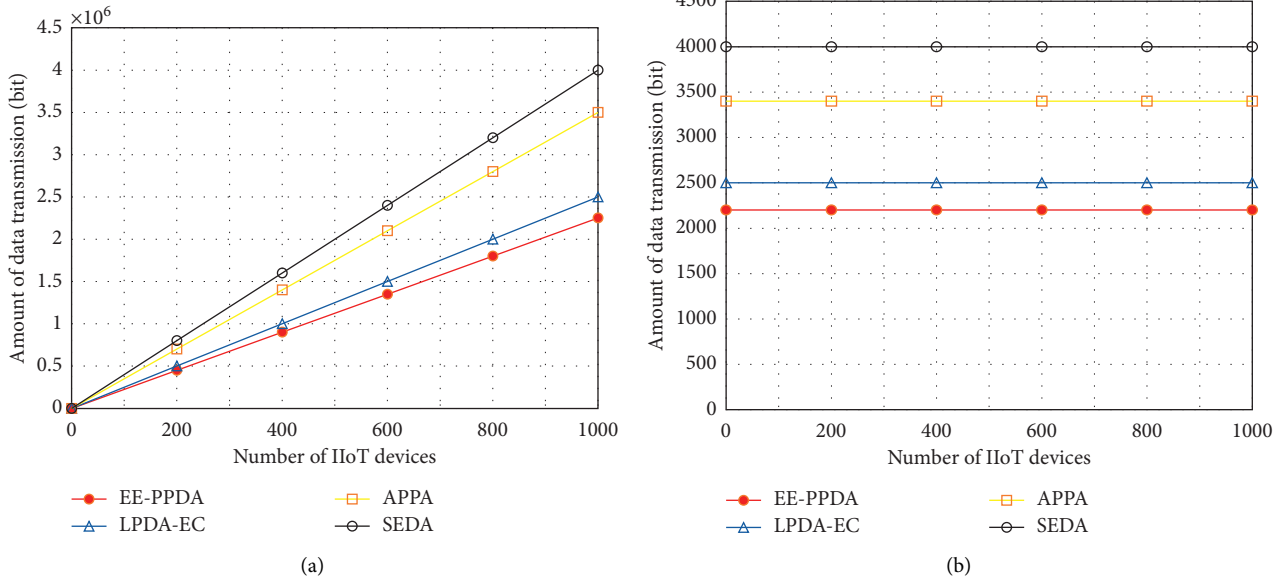


FIGURE 3: (a) Device-to-EN data transfer volume comparison. (b) EN-to-IC data transfer volume comparison.

the local data processing stage, since each edge node aggregates n ciphertexts into one and generates an aggregate report $(C_{i,r}, H_{i,r})$ and sends it to the IC, the amount of data transmission from edge node to IC is significantly reduced. Specifically, the amount of data transfer in the EN-to-IC phase is reduced from $(2048 + 160) \cdot n$ bits to $S_{FC} = 2048 + 160$ bits. Figure 3(a) shows the comparison results of the data transmission volume of the four schemes in the device-to-edge phase. It is obvious that the proposed EE-PPDA scheme achieves the slowest growth rate, and among the four schemes keep the data transfer volume to a minimum. This shows that the EE-PPDA scheme effectively reduces the amount of data communication in the device-to-edge process. From Figure 3(b), it can be found that the increase in the number of IIoT devices will not lead to an increase in the data transmission volume in the EN-to-IC phase, which is attributed to the aggregation operation of the edge nodes. At the same time, the EE-PPDA scheme still achieves the least amount of data transfer among the four schemes in the EN-to-IC phase. Combining Figures 3(a) and 3(b), it can be seen that EE-PPDA can significantly reduce communication overhead and bandwidth consumption.

From the above security and performance analysis results, it can be seen that the proposed EE-PPDA scheme is an efficient and secure data aggregation scheme. These security and performance advantages are very suitable for actual IIoT scenarios.

6. Conclusions

This paper proposes a hierarchical data aggregation scheme with efficient privacy protection in edge computing assisted IIoT, referred to as EE-PPDA. By adopting an improved homomorphic Paillier algorithm and a simple hash chain mechanism, EE-PPDA can provide effective protection for data privacy,

confidentiality, and integrity at the same time. In particular, the data privacy of a single device is also protected in semitrusted edge nodes and the cloud. At the same time, the CRT-based hierarchical aggregation design enables the cloud to provide fine-grained data services by obtaining aggregation results in smaller subregions. Finally, the experimental results further prove the advantages of the scheme in terms of calculation and communication costs. In future work, consider integrating data space-time compression, network resource optimization theory, and machine learning methods into the solution in this paper to build a more efficient and smarter data aggregation solution. At the same time, the hierarchical aggregation scheme proposed in this paper provides a fault-tolerant mechanism for data loss to ensure the normal operation of the system. However, data loss will affect the final data analysis results. How to reconstruct the lost data can be considered as a future research direction.

Data Availability

The experimental data used to support the results of this study can be obtained from the corresponding authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. 62162039 and 61762060) and Foundation for the Key Research and Development Program of Gansu Province, China (Grant no.20YF3GA016).

References

- [1] P. Borovska and M. Gugutkov, "The intersection of IoT ecosystem security and blockchain technology in the context of industry 4.0," *THERMOPHYSICAL BASIS OF ENERGY TECHNOLOGIES (TBET 2020)*, pp. 10–14, 2021.
- [2] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
- [3] K. Sha, T. A. Yang, W. Wei, and S. Davari, "A survey of edge computing-based designs for IoT security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 195–202, 2020.
- [4] G. Alandjani, "Leveraging vulnerabilities in sensor based IOT edge computing networks[]," *International Journal of Future Generation Communication and Networking*, vol. 14, no. 1, pp. 11–20, 2021.
- [5] B. Zhao, X. Liu, W.-N. Chen, W. Liang, X. Zhang, and R. H. Deng, "PRICE: privacy and reliability-aware real-time incentive system for crowdsensing," *IEEE Internet of Things Journal*, no. 99, p. 1, 2021.
- [6] B. Zhao, S. Tang, X. Liu, X. Zhang, and W.-N. Chen, "IronM: privacy-preserving reliability estimation of heterogeneous data for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5159–5170, 2020.
- [7] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.
- [8] Y. Zhang, R. Deng, D. Zheng, J. Li, P. Wu, and J. Cao, "Efficient and robust certificateless signature for data crowdsensing in cloud-assisted industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5099–5108, 2019.
- [9] K. P. Yu, L. Tan, M. Aloqaily, H. Yang, and Y. Jararweh, "Blockchain-enhanced data sharing with traceable and direct revocation in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, p. 11, 2021.
- [10] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.
- [11] R. Lu, X. Liang, X. Lin, and X. Shen, "EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.
- [12] L. Chen, R. Lu, Z. Cao, K. AlHarbi, and X. Lin, "MuDA: multifunctional data aggregation in privacy-preserving smart grid communications," *Peer-to-Peer Networking and Applications*, vol. 8, no. 5, pp. 777–792, 2015.
- [13] B. Yang, X. Cao, X. Li, Q. Zhang, and L. Qian, "Mobile-edge-computing-based hierarchical machine learning tasks distribution for IIoT," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2169–2180, 2019.
- [14] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2053–2064, 2013.
- [15] C. Li, R. Lu, H. Li, L. Chen, and J. Chen, "PDA: a privacy-preserving dual-functional aggregation scheme for smart grid communications," *Security and Communication Networks*, vol. 8, no. 15, pp. 2494–2506, 2015.
- [16] H. Wang, Z. Wang, and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 712–719, 2018.
- [17] E. Shi, H. T. H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," *Network and Distributed System Security Symposium (NDSS)*, vol. 2, pp. 1–17, 2011.
- [18] J. Ni, K. Alharbi, X. Lin, and X. Shen, "Security-enhanced data aggregation against malicious gateways in smart grid," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, San Diego, CA, USA, December 2015.
- [19] L. Chen, R. Lu, and Z. Cao, "PDAFT: a privacy-preserving data aggregation scheme with fault tolerance for smart grid communications," *Peer-to-Peer Networking and Applications*, vol. 8, no. 6, pp. 1122–1132, 2015.
- [20] I. A. Kamil, S. O. Sunday, and O. Ogundoyin, "A privacy-aware data aggregation scheme for smart grid based on elliptic curve cryptography with provable security against internal attacks," *International Journal of Information Security and Privacy*, vol. 13, no. 4, pp. 109–138, 2019.
- [21] L. Zhang, J. Zhang, and Y. H. Hu, "A privacy-preserving distributed smart metering temporal and spatial aggregation scheme," *IEEE Access*, vol. 7, pp. 28372–28382, 2019.
- [22] R. Lu, K. Alharbi, X. Lin, and C. Huang, "A novel privacy-preserving set aggregation scheme for smart grid communications," in *Proceedings of the IEEE global communications conference (GLOBECOM)*, pp. 1–6, San Diego, CA, USA, December 2015.
- [23] M. Tahir, A. Khan, A. Hameed, M. Alam, M. K. Khan, and F. Jabeen, "Towards a set aggregation-based data integrity scheme for smart grids," *Annals of Telecommunications*, vol. 72, no. 9–10, pp. 551–561, 2017.
- [24] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: privacy-preserving multisubset data aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 462–471, 2017.
- [25] F. Knirsch, D. Engel, and Z. Erkin, "A fault-tolerant and efficient scheme for data aggregation over groups in the smart grid," in *Proceedings of the IEEE Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, 2017.
- [26] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, 1999.
- [27] J. Zhang, Y. Zhao, J. Wu, and B. Chen, "LPDA-EC: a light-weight privacy-preserving data aggregation scheme for edge computing," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 98–106, Chengdu, China, October 2018.
- [28] Z. Guan, Y. Zhang, L. Wu et al., "APPA: an anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *Journal of Network and Computer Applications*, vol. 125, pp. 82–92, 2019.

Research Article

Private Data Aggregation Based on Fog-Assisted Authentication for Mobile Crowd Sensing

Ruyan Wang,^{1,2,3} Shiqi Zhang ,^{1,2,3} Zhigang Yang,^{1,2,3} Puning Zhang,^{1,2,3} Dapeng Wu,^{1,2,3} Yongling Lu,⁴ and Alexander Fedotov⁵

¹School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

²Advanced Network and Intelligent Connection Technology Key Laboratory of Chongqing Education Commission of China, Chongqing 400065, China

³Chongqing Key Laboratory of Ubiquitous Sensing and Networking, Chongqing 400065, China

⁴State Grid Jiangsu Electric Power Company Ltd. Research Institute, Nanjing 211103, China

⁵Peter the Great St. Petersburg Polytechnic University, Polytechnicheskaya, 29, St.Petersburg 195251, Russia

Correspondence should be addressed to Shiqi Zhang; 17780734752@163.com

Received 8 May 2021; Accepted 28 August 2021; Published 22 September 2021

Academic Editor: James Ying

Copyright © 2021 Ruyan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In mobile crowd sensing (MCS), the cloud as a single sensing platform undertakes a large number of communication tasks, leading to the reduction of sensing task execution efficiency and the risk of loss and leakage of users' private data. In this paper, we propose a spatial ciphertext aggregation scheme with collaborative verification of fog nodes. Firstly, the cloud and fog collaboration architecture is constructed. Fog nodes are introduced for data validation and slices transmission, reducing computing cost on the sensing platform. Secondly, a multipath transmission method of slice data is proposed, in which the user identity and data are transmitted anonymously by the secret sharing method, and the data integrity is guaranteed by hash chain authentication. Finally, a spatial data aggregation method based on privacy protection is presented. The ciphertext aggregation calculation of the sensing platform is realized through Paillier homomorphic encryption, and the problem of insufficient data coverage in the sensing region is solved by the position-based weight interpolation method. The security analysis demonstrates that the scheme can achieve the expected security goal. The simulation results show the feasibility and effectiveness of the proposed scheme.

1. Introduction

With the rapid development of mobile communication technology and the popularity of various wearable mobile devices, mobile users can collect various data anytime and anywhere. Mobile crowd sensing (MCS) is an emerging perception model. Mobile users collect sensing data for specific tasks through sensors (e.g., cameras and temperature sensors) that are embedded in the phone or wearable device. Then, the data is uploaded to sensing platforms by wireless sensing technologies (e.g., wireless networks and Bluetooth). After the task is completed, mobile users get paid from the platform [1, 2]. While receiving the sensing data, the sensing platform is responsible for evaluating and aggregating

sensing data. Data aggregation often mines the raw data for more useful information. For example, the average air quality index obtained by aggregation can reflect the local air quality condition more intuitively; the average travel speed of public transportation on a road can reflect the congestion of that road. After processing the uploaded data, the platform transmits the uploaded data to the task initiator and completes the sensing task. With low deployment cost and large coverage area, MCS can be applied in areas such as traffic congestion prediction [3, 4], industrial IoT [5–7], traffic detection [8, 9], smart medical [10, 11], environmental detection [12], and social networking [13, 14].

However, MCS faces some serious problems in privacy, security, and communication in the above applications.

Firstly, the sensing data collected by MCS often involves the user's location data that contains abundant personal information. If an attacker obtains the user's geographic location from the perceived data, the user's activity range can be inferred [15, 16]. To protect sensitive information of mobile users, most studies encrypt or add noise to the sensing data, such as local differential privacy [17, 18]. However, the sensing platform cannot aggregate the encrypted data, which reduces the usability of the sensing data. Secondly, when transmitting sensing data through wireless networks, the sensing data is easily exposed to channel monitors, making it more easily attacked, stolen, and tampered with. Existing studies mostly carry out tamper-proof authentication of perceived data by generating hash abstract or hash chain [19, 20] or provide an identity authentication system [21] to prevent attackers from malicious submission of false data. However, there is still a risk that the generated hash value will be intercepted by the attacker. In addition, when the number of sensing terminals is too large, the frequent data verification by the sensing platform will bring huge communication and computing costs and reduce the efficiency of the sensing platform. Finally, mobile users are randomly distributed in various locations in the city, and the sensing data collected and uploaded are discrete. These discrete distributions of sensing data are not conducive to the overall evaluation of the sensing area, so to obtain the sample values of unknown locations, they are generally obtained by interpolation algorithms related to the location of the sensing data, but they often reveal the specific location of the mobile users and leak user privacy.

Targeting at the above problems, this paper proposes a spatial ciphertext aggregation scheme with collaborative verification of fog nodes. Inspired by the significant advantages of fog nodes [22, 23], we use fog nodes for data validation and slice transmission to alleviate the communication and computation costs of the sensing platform. Shamir secret sharing is used to transmit the sensing data and user identity information to the fog nodes in the form of slices, which ensures the integrity of the sensing data and the privacy security of the user identity and then combines the one-way hash function to complete data authentication, and finally, the sensing platform recovers the encrypted data and user identity information to complete other operations. The scheme also ensures the aggregated computation of the sensed data in encrypted form, while the prediction of the sample values of unknown locations is realized in combination with the geographic interpolation algorithm, which enables the overall data evaluation of the sensing area. The main contributions of this paper are as follows:

- (1) A novel cloud and fog collaboration architecture is constructed. Fog nodes are introduced to assist the sensing platform considering its characteristics of low delay, multiple distribution, and certain computing capacity, realizing data verification and slice reception, and reducing the communication and computing costs of the sensing platform.

- (2) A multipath transmission method of slice data is put forward. Sensing data and user identity information are sliced and transmitted through Shamir secret sharing. Then, a reasonable secret threshold t is set according to the number of fog nodes to realize anonymous transmission of user identity, and hash chain authentication is adopted to achieve a trade-off between privacy protection and data integrity.
- (3) A spatial data aggregation method based on privacy protection is advanced. The ciphertext aggregation calculation of the sensing platform is realized through Paillier homomorphic encryption, and the problem of insufficient data coverage in the sensing region is solved by the position-based weight interpolation method.

The remainder of this paper is organized as follows. The related works are introduced in Section 2. Section 3 describes the preliminary knowledge of Paillier encryption protocol, secret sharing, and inverse distance weighted. The system model is introduced in Section 4. Then, Section 5 introduces the spatial secret aggregation scheme with collaborative verification of fog nodes. And, the security analysis and simulation results are described in detail in Section 6. Finally, Section 7 summarizes the paper.

2. Related Work

The privacy protection issues in the MCS system mainly focus on privacy task allocation, data collection, and data aggregation. Relevant researchers have published the following research results on these issues.

Based on fog-assisted computing, a Privacy-Aware Task Allocation and Data Aggregation (PTAA) scheme was proposed by using bilinear pairing and homomorphic encryption technology in literature [24]. The scheme took advantage of the fog nodes to assist the sensing platform to assign tasks and used the transport independent protocol and the secure two-party aggregation protocol to realize the privacy task assignment and data aggregation, reducing the burden of the sensing platform. Ni et al. [25] proposed a Fog-Assisted Secure Data Deduplication (Fo-SDD) scheme. By designing a BLS-oblivious pseudorandom function, it enabled fog nodes to delete deduplicated data, while protecting privacy, ensuring data confidentiality, and improving communication efficiency. The scheme also achieved anonymization of user identity during data collection by further extending Fo-SDD. Basudan et al. [26] proposed a Certificateless Aggregate Signcryption (CLASC) scheme to enhance security in data transmission of vehicular crowd sensing based on the road surface condition monitoring system with fog computing, which ensured data privacy security using lower computation cost. However, the above scheme does not consider the risk of interception of sensing data during transmission, and a malicious attacker may intercept the transmission data in the open transmission network, resulting in the loss of sensing data and affecting the sensing task to be performed.

Concerning data collection and aggregation, Chen et al. [27] put forward a data privacy protection method for untrusted servers. The collected data was divided into multiple slices based on the number of adjacent participants, and then, the data slices were forwarded to the adjacent participants. When the number of slices reached a threshold, all slice carriers sent data slices directly to the server. However, this method simply distributed the data slices randomly to the neighboring nodes. When data slices were transmitted, attackers can easily collect data slices, leading to an increased probability of data leakage. In literature [28], a privacy-preserving data aggregation scheme was designed using data slicing and blending techniques, which supports additive aggregation. Data slices were distributed to neighboring participants; thus, the participants' sensing data was hidden. Li and Cao [29] presented a new mobile sensing protocol to obtain the sum of time-series data, which uses homomorphic encryption and a novel key management scheme based on efficient HMAC to achieve additive ciphertext aggregation of sensed data. However, the protocol required additional communication to handle dynamic user access. But the above literature did not consider the case where the participants collude with the server to leak privacy. Fan et al. [30] came up with a novel privacy-aware and trustworthy sum aggregation protocol for mobile sensing, which protected the data privacy of benign users even when multiple users conspire against each other, but there was still a risk of losing the submitted data.

In other studies in the area of MCS security, Agir et al. [31] proposed a user-adaptive location privacy protection scheme, which generated multiple noises by setting a personal privacy threshold and a user-defined privacy protection level. Then, the user's privacy security was guaranteed combined with the spatial steganography unit. However, this solution was computationally expensive and lacked effective privacy level criteria. Gisdakis et al. [32] used Security Assertion Markup Language (SAML) and Transport Layer Security (TLS) protocols to establish trust between entities, and then, Private Information Retrieval (PIR) techniques were adopted to ensure privacy in communication. Based on the Merkle tree, the privacy protection mechanism in literature [33] was presented, which can authenticate participants anonymously without the trusted third party. However, the above schemes did not consider the case that malicious attackers submit false data, which may interfere with the final results.

3. Preliminaries

3.1. Paillier Encryption Protocol. The Paillier Cryptosystem is a modular, public-key encryption scheme, created by Pascal Paillier [34]. The security of this homomorphic encryption scheme is based on determining the n th-order residue class problem. In the following, we will review the specific process of the program:

3.1.1. Key Generation. To construct the key, one must choose two large primes p and q , and then, compute $n = pq$, $\lambda = \text{lcm}[(p-1)(q-1)]$, where $\text{lcm}(p, q)$ is calculated as the

least common multiple of p and q . Then, select a semi-random, nonzero value $g \in Z_n^*$ such that $k = L(g^\lambda \bmod n^2)$, where $L(u) = u - 1/n$. It is said that g is semi-random since k generated by g needs to satisfy $\text{gcd}(k, n) = 1$, and then, calculate $\mu = k^{01} \bmod n$.

The public key Pk is (n, g) , and the private key Sk is (λ, μ) .

3.1.2. Encryption. For the plaintext m , select the random parameter $r \in Z_n^*$. Then, the ciphertext

$$\begin{aligned} c &= E(m) \\ &= g^m \cdot r^n \bmod n^2. \end{aligned} \quad (1)$$

3.1.3. Decryption. The Paillier decryption function:

$$m = L(c^\lambda \bmod n^2) \cdot \mu \bmod n. \quad (2)$$

3.1.4. Homomorphic Properties. An encryption function with the homomorphic property is an encryption function where two plaintexts m_1 and m_2 satisfy $C(E(m_1), E(m_2)) = E(m_1 \oplus m_2)$, where C is an operation on the ciphertext domain. When \oplus represents addition, the encryption is said to be additive homomorphic encryption; when \otimes represents multiplication, the encryption is said to be multiplicative homomorphic encryption. Homomorphic properties of the Paillier encryption algorithm:

$$D(E(m_1) \cdot E(m_2) \bmod n^2) \equiv m_1 + m_2 \bmod n. \quad (3)$$

3.2. Shamir Secret Sharing Algorithm. The secret sharing algorithm was proposed by Shamir in 1979 based on Lagrange interpolation, which allows n participants to share a secret value s , but the secret value s can be recovered by any t participants, and less than t participants cannot get any information about s . The above t is called the threshold, and a secret sharing with n participants and a threshold of t is denoted as (t, n) -secret sharing. The formal definition of Shamir secret sharing is as follows.

3.2.1. Related Parameters. The finite domain F_q is chosen, the secret value $s \in F_q$, t is the threshold, the set of participants is $U = \{u_1, u_2, \dots, u_n\}$, the identity of each participant is u_i , and $u_i \in F_q$ is not equal to zero.

3.2.2. Slicing and Distribution. Randomly choose a $t-1$ degree polynomial $f(x)$ on F_q ; $f(x)$ is shown below:

$$f(x) = s + a_1x^1 + a_2x^2 + \dots + a_{t-1}x^{t-1} \bmod q, \quad (4)$$

where $a_1, a_2, \dots, a_{t-1} \in F_q$ in $f(x)$. Then, all secret slices are calculated based on participant identity:

$$y_i = f(u_i). \quad (5)$$

Finally, the computed slices are secretly distributed to the corresponding participant u_i .

3.2.3. *Secret Recovery.* When there are no less than t participants providing secret slices, one can use u_i and y_i to recover $f(x)$, and hence the $t-1$ degree polynomial $f(x)$ can be easily obtained by using the equation as follows:

$$f(x) = \sum_{i=1}^t y_i \prod_{j=1, j \neq i}^t \frac{x - u_j}{u_i - u_j} \text{mod } q. \quad (6)$$

After that, the secret value s is recovered by substituting $x = 0$ into $f(x)$.

3.3. *Inverse Distance Weighted.* Inverse distance weighted (IDW) is a weighted average interpolation method that can be interpolated in an exact or smooth manner. It uses the distance between the interpolation point and the sample point as the weight for the weighted average, and the closer the sample point is to the interpolation point, the greater the weight given to it. Suppose that the predicted location is (x_0, y_0) , the predicted value is z , the perceived user location is (x_i, y_i) , the perceived data is m_i , and the number of participating users is n . Calculate z according to the following steps:

- (1) Calculate the Euclidean distance for each point:

$$(x_i - x_0)^2 + (y_i - y_0)^2 = d_i. \quad (7)$$

- (2) Calculate the distance weights for each point:

$$w_i = \frac{d_i^{-1}}{\sum_{i=1}^n d_i^{-1}}. \quad (8)$$

- (3) Calculate the value of the unknown point:

$$\begin{aligned} z &= \sum_{i=1}^n w_i m_i \\ &= \frac{d_1^{-1} m_1 + d_2^{-1} m_2 + \dots + d_n^{-1} m_n}{\sum_{i=1}^n d_i^{-1}}. \end{aligned} \quad (9)$$

4. System Model

4.1. *System Model.* As shown in Figure 1, the spatial ciphertext aggregation system with collaborative verification of fog nodes consist of sensing platform, task initiator, fog nodes, mobile users, and authority center.

4.1.1. *Task Initiator.* Task initiators are users of the MCS services. The task initiator is responsible for issuing a specific task, and each task has the clear data type requirement. A task initiator could be an individual or organization that lacks an ability to perform a certain computing or data collection task.

4.1.2. *Sensing Platform.* The sensing platform could be played by an organization or a corporation that provides a platform for crowdsourcing. It accepts service requests from task initiator, deals with the requests, selects proper mobile users, and assigns relevant tasks to them.

4.1.3. *Fog Nodes.* The fog nodes act as a relay between the sensing platform and the mobile user, undertaking data verification and the reception and distribution of data slices.

4.1.4. *Mobile Users.* Referring to mobile users with sensing devices, mobile users collect data and calculate spatially relevant statistical information as required by the task. After encrypting the data, the sensing data and identity data are sliced according to the number of fog nodes deployed. Finally, the slices are sent to the fog nodes along with the authenticated hash digest value.

4.1.5. *Authority Center.* It is responsible for generating and distributing key materials to data requestors and MCS servers. In this system, the authority center distributes the generated public key and the parameters required for data slicing to mobile users for data encryption and slicing and distributes the private key to task initiator so that they can download the aggregated encrypted data from the sensing platform and get the specified task data.

4.2. *Security Model.* In the architecture of this paper, we assume that the authority center is fully trusted and that the authority center cannot be attacked by any attackers and that it manages the distribution of keys and other parameters. Task initiator, sensing platform, fog nodes, and mobile users are all honest but curious, and each part will follow the rules to perform its own task, but will also infer information about others based on the data it holds. And, external security threats come from malicious attackers; in general, attackers may listen to communication channels and intercept encrypted sensing data, spatial data, etc.

4.3. *Design Objective.* Based on the above security model and system architecture, we propose the following design goals:

4.3.1. *Privacy.* During the task execution, the specific location and sensing data of the mobile user are encrypted, and the fog nodes and sensing platform do not know the specific location and sensing data of the mobile user. In the data aggregation phase, the aggregated data is still stored in the encrypted form in the sensing platform, and only the task initiator can access it through the private key.

4.3.2. *Security.* The encrypted sensing data and user identity information are distributed to the fog nodes in a slicing manner so that an attacker cannot obtain the specific sensing data and user identity information even if he intercepts part

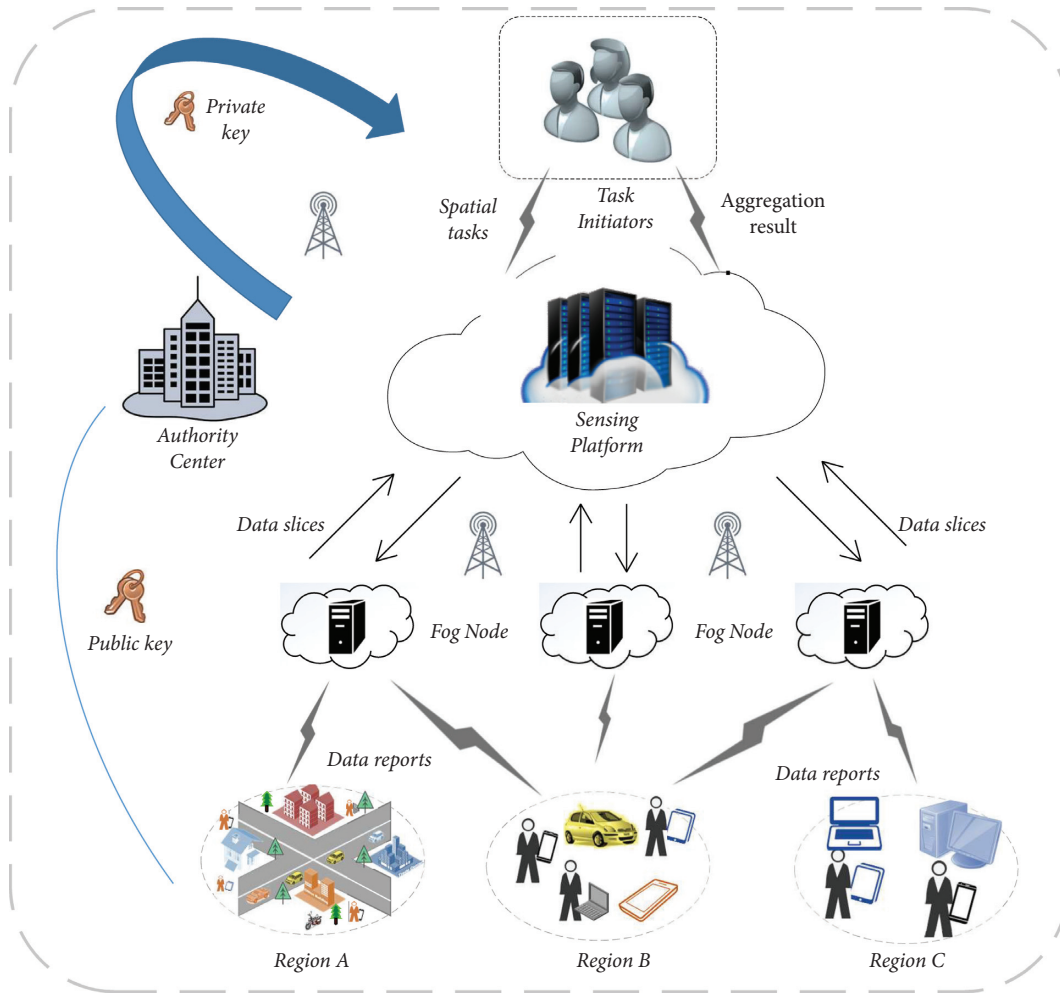


FIGURE 1: System architecture.

of the data slices. And, the data slices come with a hash digest, so an attacker cannot interfere with sensing data recovery by tampering with some of the data slices. For the internal perception system, the fog nodes only undertake the function of receiving and forwarding in pieces, and the user identity information can only be obtained after secret recovery by the sensing platform, which ensures the privacy and security of the user identity.

4.3.3. *Efficiency.* Fog nodes take on the verification of sensing data, reducing the communication and computation cost of the sensing platform.

5. Spatial Ciphertext Aggregation Scheme with Collaborative Verification of Fog Nodes

In this section, we propose a spatial secret aggregation scheme with collaborative verification of fog nodes, which consists of five phases: system initialization, mobile user data report generation, data validation and slices reception, secret recovery and data aggregation, and data decryption and result acquisition.

5.1. *Overview.* Task initiator initiates spatially relevant task requests to obtain overall sensing data for a region. After receiving the task request, the sensing platform assigns the task to the mobile users. Then, the authority center configures the system parameters, distributing the public key and fog nodes identity to the mobile users and the private key to the task initiator. Mobile users collect data according to the requirements of task. Because the specific locations of mobile users within the sensing area are discrete, the uploaded sensing data has limited coverage. And, mobile users need to calculate spatially relevant statistical information to get sample values of some unknown locations in combination with geographic interpolation that make the uploaded data in the area more holistic.

This paper focuses on describing the computation of sample values for unknown locations by data aggregation using homomorphic encryption and geographic interpolation. In this process, in order to hide the mobile users' location data and identity information and to protect the privacy of the sensing data, mobile users encrypt data with public keys, slice the data and identity information based on the number of fog nodes, and then use one-way hash functions to generate hash chain for data authentication.

Mobile users distribute data, identity information slices, and authentication information to the corresponding fog nodes. Afterward, the fog nodes verify its data integrity and transmit the data and identity information slices to the sensing platform after the verification is completed. The sensing platform receives the data slices and performs secret recovery to get the mobile users' encrypted sensing data and the users' original identity information. The sensing platform completes the incentive or other operations based on the identity information and then performs ciphertext data aggregation. After aggregation is completed, the task initiator downloads the aggregated data via the private key to obtain the aggregated results.

5.2. System Initialization. In our system model, consider mobile users as $P = \{p_1, p_2, p_3, \dots, p_n\}$, mobile user location as $\{x_i, y_i\}$, sensing data as m_i , identity information as p_i , spatially relevant statistical information as D_i , unknown locations as (x_o, y_o) , fog nodes as $U = \{u_1, u_2, u_3, \dots, u_k\}$, each fog node identity as u_j , and hash function as h . At the beginning of the sensing task, the authority center randomly selects two large prime numbers p and q , calculates $n = pq$ according to the predefined calculation principle, and satisfies $\gcd[L(g^\lambda \bmod n^2), n] = 1$. The public key (n, g) is transmitted to the mobile users, and the secret sharing-related parameters and the fog node identity u_j are also sent to the mobile users together. Then, the authority center computes $\lambda = \text{lcm}[(p-1), (q-1)]$ and $\mu = L(g^\lambda \bmod n^2)^{01} \bmod n$ and transfers the private key (μ, λ) to the task initiator.

5.3. Location-Aware Inverse Distance Weighted Ciphertext Aggregation Protocol. As shown in Figure 2, m_i represents the sensing data collected by mobile user p_i at its location, and d_i represents the Euclidean distance between the mobile user and the unknown location. At the beginning of the sensing task, the sensing platform broadcasts the coordinates of the unknown location and the mobile user computes the Euclidean distance d_i between itself and the unknown location. Then, the mobile user encrypts $d_i^{-1}m_i$ and d_i^{-1} to get C_{i1} and C_{i2} . The sensing platform receives encrypted data from n mobile users and uses homomorphic encryption properties to obtain sensing data aggregation results with the ciphertext form. Then, the task initiator uses the private key transmitted by AC to decrypt and finally gets the aggregated result with plaintext form $d_1^{-1}m_1 + d_2^{-1}m_2 + d_3^{-1}m_3 + \dots + d_n^{-1}m_n$ and $d_1^{-1} + d_2^{-1} + d_3^{-1} + \dots + d_n^{-1}$. Based on the knowledge in the Preliminaries section, the sample value z for the unknown location can be calculated.

5.4. Mobile User Data Report Generation. This phase is divided into three main steps: sensing data acquisition and spatial data calculation, data encryption, and data transmission.

Step 1. Sensing data acquisition and spatial data calculation: each mobile user p_i collects sensing data m_i as required by the task and calculates spatial data based on its own location:

$$\frac{1}{(x_i - x_o)^2 + (y_i - y_o)^2} = d_i. \quad (10)$$

Due to the properties of Paillier homomorphic encryption, data transformation of d_i is required to obtain spatially relevant statistical information for encryption:

$$D_i = \lceil d_i \cdot 10^k \rceil. \quad (11)$$

where k varies with the sensing area range to ensure that D_i is an integer and $\lceil \cdot \rceil$ is the rounding symbol.

Step 2. Data encryption: for each mobile user p_i , after sensing data collection and computing spatially relevant statistical information are performed, data encryption is performed using the received public key (n, g) :

$$\begin{aligned} c_{i1} &= E(D_i m_i) \\ &= g^{D_i m_i} \cdot r^n \bmod n^2, \\ c_{i2} &= E(D_i) \\ &= g^{D_i} \cdot r^n \bmod n^2. \end{aligned} \quad (12)$$

where c_{i1} and c_{i2} denote the ciphertext information obtained by the user after encrypting $D_i m_i$ and D_i .

Step 3. Data transmission: before performing data forwarding, authority center (AC) counts the number of working fog nodes in the current sensing area, sets a maximum number of slices M_{\max} , and queries the historical data forwarding success rate of fog nodes in the area. After that, AC makes a trade-off between privacy of the transmitted data and efficiency of the sensing task completion. If this sensing task requires higher privacy of the transmitted data, AC selects the threshold t based on the maximum number of slices M_{\max} . On the contrary, if the sensing task needs to be completed efficiently and the privacy requirement of the transmitted data is lower, AC prioritizes the fog nodes with a high success rate of historical forwarded data and generates a threshold t based on the number of these fog nodes. After that, the AC sends the fog node identity, threshold t , and other data slicing related parameters to the mobile user and the sensing platform. Mobile user p_i splits two copies of data c_{i1} and c_{i2} and own identity information p_i into k slices according to the number of fog nodes, while setting a suitable threshold value t . Mobile user p_i slices data and identity information according to the fog nodes' identity $U = \{u_1, u_2, u_3, \dots, u_k\}$ distributed by the authority center:

$$\begin{aligned} f_1^i(x) &= c_{i1} + a_1 x^1 + a_2 x^2 + \dots + a_{t-1} x^{t-1} \bmod q, \\ f_2^i(x) &= x_{i2} + a_1 x^1 + a_2 x^2 + \dots + a_{t-1} x^{t-1} \bmod q, \\ f_3^i(x) &= p_i + a_1 x^1 + a_2 x^2 + \dots + a_{t-1} x^{t-1} \bmod q. \end{aligned} \quad (13)$$

The mobile user p_i gets the data and identity information slices generated by the identity identifiers of the

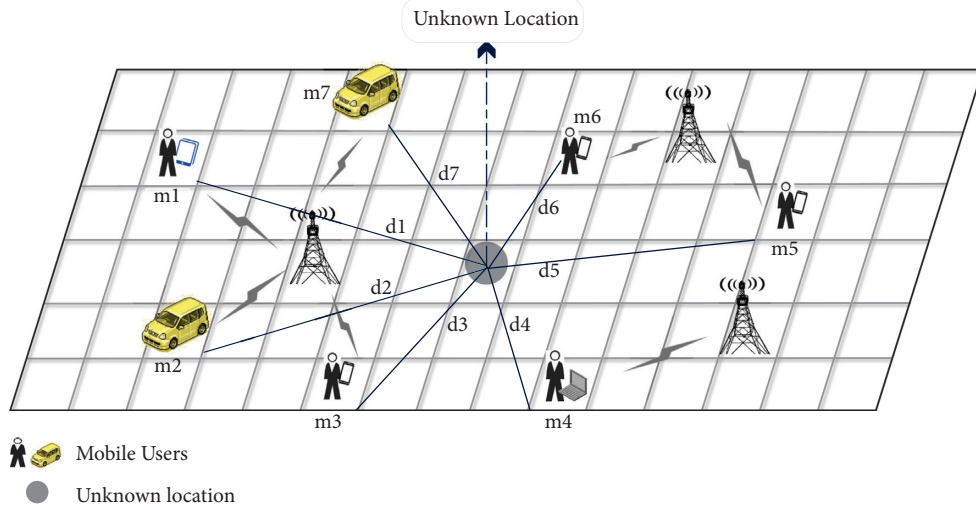


FIGURE 2: Location-aware inverse distance weighted ciphertext aggregation protocol.

k fog nodes, respectively. $f^i(u_j)$ denotes the slice obtained by the mobile user p_i through the fog node identity u_j , and n and k are the number of mobile users and fog nodes, respectively, and the following are the slices generated by the data c_{i1} and c_{i2} and identity information p_i of user p_i , respectively:

$$\begin{aligned} &f_1^i(u_1), f_1^i(u_2), f_1^i(u_3), \dots, f_1^i(u_k), \\ &f_2^i(u_1), f_2^i(u_2), f_2^i(u_3), \dots, f_2^i(u_k), \\ &f_3^i(u_1), f_3^i(u_2), f_3^i(u_3), \dots, f_3^i(u_k). \end{aligned} \quad (14)$$

As shown in Figure 3, the mobile user p_i generates data slices, connects the data slice $f^i(u_j)$ with the hash digest value h_{j01}^i generated by the previous data slice $f^i(u_{j01})$ to generate a new hash digest value h_j^i , and points to the next data slice $f^i(u_{j+1})$ until the final generation of the end of the hash chain h_k^i .

Finally, the mobile user p_i sends the k data slices $f_1^i(u_j)$ and $f_2^i(u_j)$ ($1 \leq j \leq k$) generated from data c_{i1} and c_{i2} along with the corresponding hash digest values and k identity information slices $f_3^i(u_j)$ ($1 \leq j \leq k$) to the k corresponding fog nodes.

5.5. Data Validation and Slices' Reception. In this phase, mobile users send their encrypted data slices with authentication information and identity information slices to the fog nodes. Then, fog nodes will first verify the integrity of the encrypted data. As shown in Figure 4, after receiving the data slice $f^i(u_j)$ corresponding to mobile user p_i , fog node u_j uses the hash digest h_{j-1}^i sent by the previous fog node u_{j01} , connects to generate h_j^i , and transmits it to the next fog node u_{j+1} . Finally, the last fog node u_k compares the two generated hash chain tails h_{k1}^i and h_{k2}^i with the received h_{k1}^i and h_{k2}^i , and if the results are consistent, the verification is successful. In the above process, there is a certain probability that the data slices are stolen by the attacker, and the fog nodes whose data

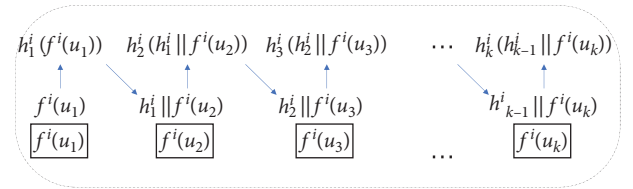


FIGURE 3: Hash chain.

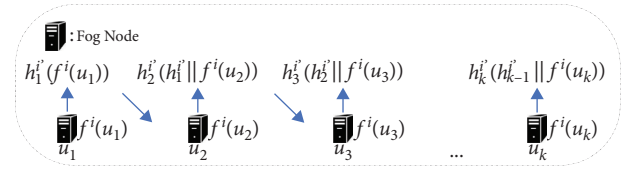


FIGURE 4: Fog nodes' collaborative data validation.

slices are intercepted by the attacker cannot compute the hash digest to complete collaborative authentication. At this time, if the number of remaining adjacent fog nodes are greater than t , the data slicing can still be collaboratively verified to ensure the integrity and authenticity of the transmitted data. If collaborative verification fails, fog node u_j compares the hash digest h_j received by itself with the computed h_j^i to locate the location of the fog node with the wrong data slice. As for the users' identity information slicing, the fog nodes undertake the function of relaying and forwarding to ensure the anonymous transmission of users' identity information. The k identity information slices $f_3^i(u_j)$ ($1 \leq j \leq k$) of user p_i are stored on the corresponding k fog nodes and transmitted to the sensing platform together after the encrypted data slices are successfully verified.

5.6. Secret Recovery and Data Aggregation. The fog nodes send the received users' identity information slices and the verified data slices to the sensing platform, which first performs secret recovery:

$$f_c^i(x) = \sum_{j=1}^t f_c^i(u_j) \prod_{L=1, L \neq j}^t \frac{x - u_L}{u_j - u_L} \text{mod } q \quad (c = 1, 2, 3). \quad (15)$$

Substituting $x = 0$ into the above equation, we get

$$\begin{aligned} f_1^i(0) &= c_{i1} \\ &= E(D_i m_i), \\ f_2^i(0) &= c_{i2} \\ &= E(D_i), \\ f_3^i(0) &= p_i. \end{aligned} \quad (16)$$

$$\begin{aligned} D_1 m_1 + D_2 m_2 + \dots + D_n m_n \leftarrow F_1 &= E(D_1 m_1) E(D_2 m_2) \dots E(D_n m_n) \text{mod } n^2, \\ D_1 + D_2 + \dots + D_n \leftarrow F_2 &= E(D_1) E(D_2) \dots E(D_n) \text{mod } n^2. \end{aligned} \quad (17)$$

5.7. Data Decryption and Result Acquisition. The task initiator decrypts the aggregation result using the received private key (μ, λ) and then computes $z = Z_1/Z_2$ to obtain the sample value z of the unknown location:

$$\begin{aligned} D_1 m_1 + D_2 m_2 + \dots + D_n m_n &= L(F_1^\lambda \text{mod } n^2) \cdot \mu \text{mod } n = Z_1, \\ D_1 + D_2 + \dots + D_n &= L(F_2^\lambda \text{mod } n^2) \cdot \mu \text{mod } n = Z_2. \end{aligned} \quad (18)$$

6. Performance Evaluation

In this section, we first analyze how the spatial ciphertext aggregation scheme with collaborative verification of fog nodes achieves the given design goals and then experimentally demonstrate the performance of this scheme in terms of communication efficiency and computation cost.

6.1. Security Analysis

6.1.1. Data Privacy and Security. In the data collection phase, the mobile user encrypts the sensing data and spatial data using the public key sent by the authority center, and the encrypted data is transmitted to the fog nodes in the form of data slices. Data verification phase, fog nodes, or other malicious attackers who intercept the data are unable to infer the plaintext message m_i from the ciphertext C_i . In the data aggregation phase, the data slices received by the sensing platform are recovered in ciphertext, and the sensing platform performs data aggregation on the received ciphertext data. After data aggregation, the aggregated results are still stored in the sensing platform in ciphertext, which only the task initiator can get by decrypting with private key. And, the sensing platform cannot get the plaintext data in the aggregation process. In general, only the task initiator can get the final result in plaintext during the above process,

The sensing platform recovers the encrypted data c_{i1} and c_{i2} of the user p_i and the identity information p_i . Then, the sensing platform uses the received identity information to achieve the incentive mechanism or performs other necessary system operations. Afterward, using the homomorphic encryption property of Paillier, the sensing platform starts ciphertext aggregation of the received encrypted data from all users:

while the fog nodes or the sensing platform can only process the ciphertext. The security of Paillier homomorphic encryption technology ensures that the sensing data can withstand internal and external privacy threats of the MCS system.

6.1.2. Data Integrity and Identity Privacy Security. For mobile users, the identity information and encrypted sensing data are divided into k slices based on the number of fog nodes. Each slice is generated based on the corresponding fog node identity, and a suitable recovery threshold t is set. When the data slice is sent to the corresponding fog node, the mobile user generates the corresponding hash chain according to the method in Section 5 and sends it to the corresponding fog node together with the data slices. Therefore, even if a malicious attacker intercepts a part of the data slices, according to the secret sharing feature in Section 3, as long as the number of remaining slices is greater than t , the sensing platform is still able to recover the encrypted data. Although some malicious attackers intercept the data slices and re-send forged messages pretending to be legitimate participants, all fog nodes will collaboratively authenticate based on the received hash chain, which guarantees the accuracy of the data source. The users' identity information are also stored in the form of slices on the fog nodes, and a single fog node cannot know the real identity of the user, less than t fog nodes also cannot collude to launch the real identity of the user, and only the sensing platform can recover to get the users' identity, to achieve the user identity anonymous transmission. After the sensing platform recovers the identity information, it completes the incentive or other system operations according to the user's identity. In this scheme, Shamir secret sharing guarantees the anonymous transmission of user identity, and combining with hash chain message authentication guarantees the integrity of data.

6.2. *Experiment.* We performed the simulation in Python 3.8, and the scenarios and related configuration parameters involved are as follows.

In the simulations, we consider a scenario in which the task initiator requests the overall air index in a region. We set the number of mobile users to 10 ~ 100 with a growth step of 10 and the number of tasks participated by each mobile user to 10 ~ 50 with a growth step of 10. Mobile user p_i randomly generates sensing data distributed in $[100, 1000]$, and the coordinates of the location of each mobile user are set to $[(x_i, y_i) | 0 \leq x_i \leq 100, 0 \leq y_i \leq 100]$. The number of fog nodes is set to 10 ~ 100, and the growth step is 30. For Paillier homomorphic encryption, we set the number of key bits to 32 ~ 256 bits to meet the security requirements of different data lengths, respectively, but it will bring some computation cost accordingly. All system simulations are simulated on a PC (CPU: Core i5-9400F @ 2.90 GHz and RAM: 8 GB).

The performance metrics include the computation cost of data encryption, data slicing, data recovery and aggregation, and data decryption. Then, we evaluate the impact of the number of mobile users, the number of fog nodes, the secret threshold t , the number of tasks per user, and the key length on the above parts.

6.2.1. *Costs of Data Encryption.* The computation cost per mobile user in the encryption phase as the number of tasks grows is given in Figure 5 to demonstrate the efficiency of data submission by mobile users. Since mobile users are located in a lightweight computing scenario, the key length of 32 ~ 256 bits can fully fulfill the data encryption requirements in this scenario, and this scheme can fulfill the privacy protection requirements of mobile users with a small increase in computing cost.

To simulate the encryption environment with different data lengths, we also give the computation cost with different key lengths. From the figure, we can see that the computation cost increases as the number of tasks per mobile user grows, which is because mobile users cannot process multiple tasks in parallel, and when the number of tasks is too large, mobile users consume a lot of computation time. At the same time, with the same number of tasks, the encryption cost varies greatly with different key lengths, so it is necessary to choose the appropriate number of key bits according to different encryption environments to fulfill the security requirements in different scenarios.

6.2.2. *Cost of Validation and Aggregation.* The computation cost of the fog nodes and the sensing platform is demonstrated in Figure 6. From the figure, it can be seen that the fog nodes undertake part of the computation tasks of the sensing platform and reduce the computation cost of the sensing platform, which is consistent with the design goal of this scheme.

In Figure 6, the fog nodes take on the task of data verification, and since each fog node receives data slices generated by each mobile user based on the identity of that fog node, the number of slices processed by each fog node increases as the number of mobile users grows, and

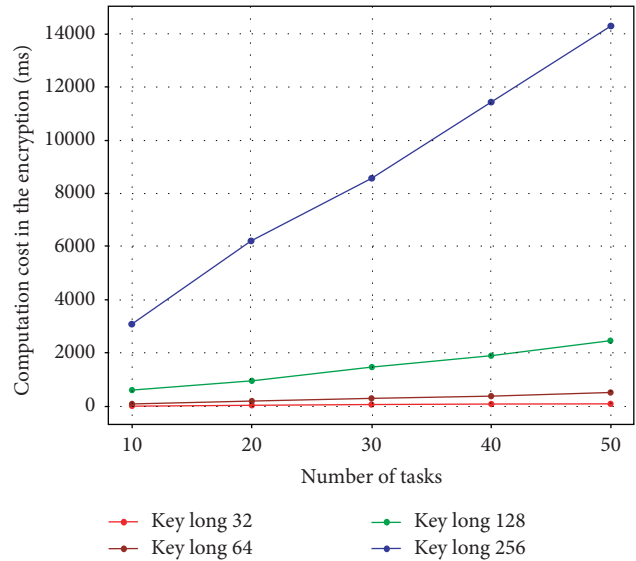


FIGURE 5: The computation cost in encryption.

the computation cost increases. And, the growth of the number of fog nodes will lead to a longer hash chain, increasing the time for collaborative verification. But the corresponding secret sharing threshold can also be increased, which can improve the security of sensing data transmission. We assume that the data is divided into n slices and the threshold is $t \leq n$, which means that the attacker can recover the sensing data by stealing t data slices, and if n is increased and t is increased accordingly, the data slices that the attacker needs to steal will increase accordingly, and the difficulty of stealing will also increase, reducing the risk of sensing data being stolen. Since the sensing platform takes on the task of data slicing recovery and ciphertext aggregation, the computation cost will be higher than fog nodes that only perform authentication. While increasing the secret recovery threshold t affects the data recovery time, the number of mobile users affects the ciphertext aggregation time, and from the four subplots in Figure 6, we can find that the computation cost of the sensing platform increases with the number of mobile users and the threshold.

6.2.3. *Data Accuracy.* Since this paper combines homomorphic encryption with IDW, the additive homomorphic property is used to compute the sample value of the unknown location. The inverse of the distance between each mobile user and the unknown location is rounded, which leads to a difference between the calculated results and those calculated using IDW. This is the main reason for the error. So, we use the relative error to express the difference between the sample values of unknown locations obtained using this scheme and the real sample values of unknown locations. The relative error can well reflect the degree of data reliability, where Z_t denotes the sample value of the unknown location obtained after the t th encryption and aggregation using this scheme, while Z_t' denotes the sample value of the unknown location obtained by the t th direct aggregation

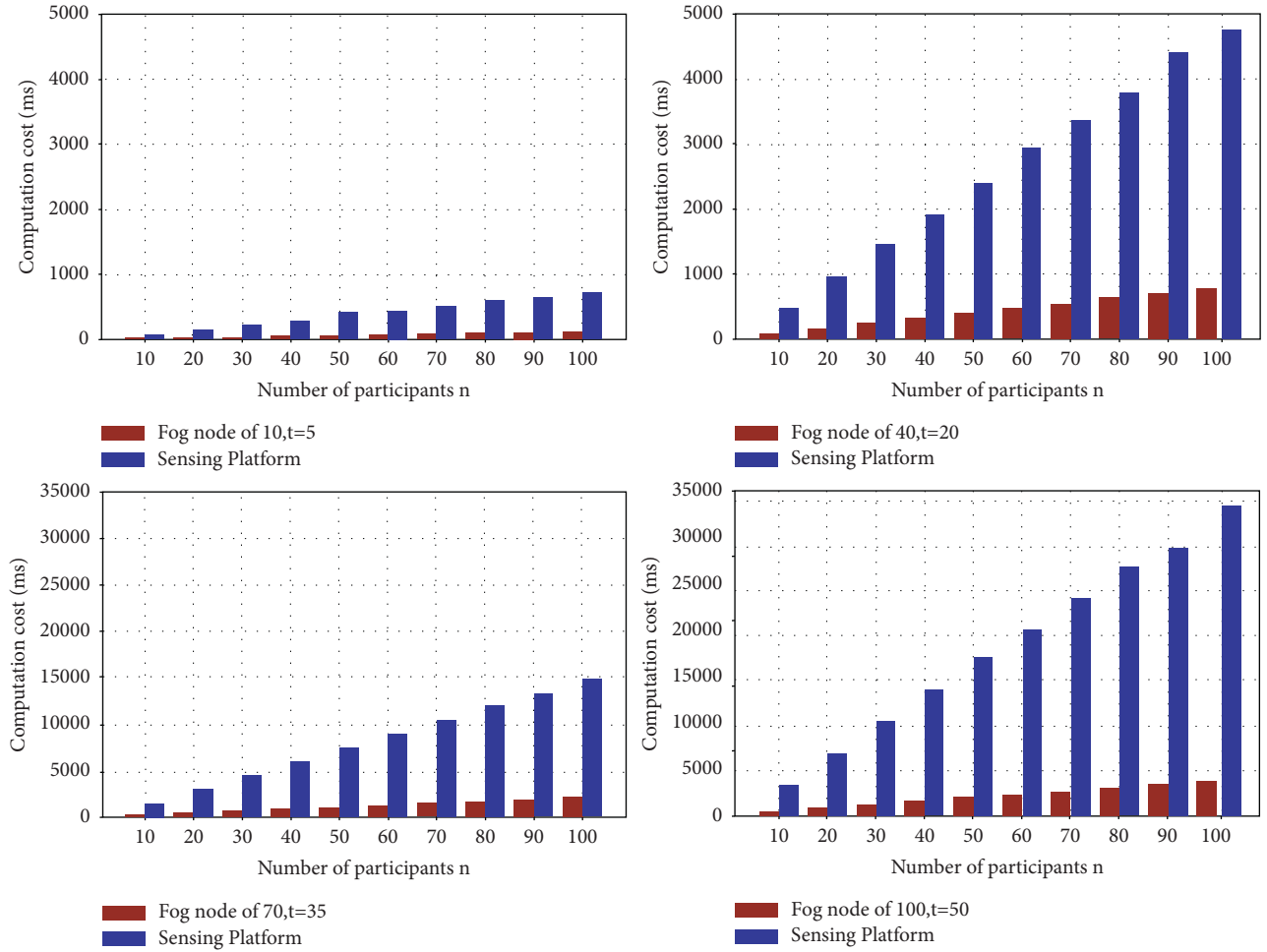


FIGURE 6: The computation cost in validation and aggregation.

without encryption, δ denotes the relative error, Δ denotes the accuracy, and the scheme will be run 1000 times to get the average relative error. The error in this scheme comes from the data error caused by rounding the data due to encryption when the mobile user calculates the spatially relevant statistical information D_i related to its location:

$$\delta = \frac{1}{n} \sum_{t=1}^n \left| \frac{Z_t - Z'_t}{Z'_t} \right|, \quad (19)$$

$$\Delta = (10\delta) \times 100\%.$$

We represent in Figure 7 the accuracy of the data obtained when different numbers of mobile users are involved in the task. The figure shows that the results obtained using our scheme are in general agreement with the real values and that our scheme is able to trade-off

privacy security in data transmission and encrypted data aggregation with a fairly small loss of accuracy.

6.2.4. Cost of Data Decryption. Figure 8 shows the computation cost of the task initiator to obtain the sensed data. Since the task initiator decrypts the data directly at the sensing platform using the private key, the key length is the main factor affecting the decryption time.

Overall, the computation cost paid by mobile users and task initiators in this scheme is much lower than that of fog nodes and sensing platform, and mobile users only need to pay a small computation cost to fulfill their own requirements for privacy protection. Therefore, this scheme can fulfill the requirements of mobile users and task requestors with limited computation power and achieve lightweight task participation.

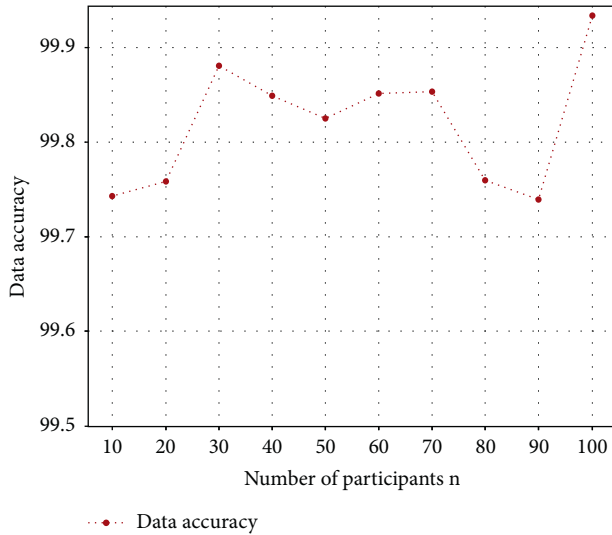


FIGURE 7: Data accuracy.

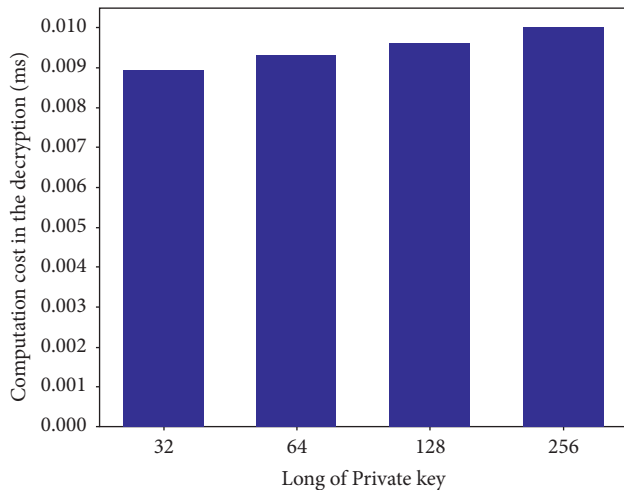


FIGURE 8: The computation cost in decryption.

7. Conclusion

In this paper, we propose a spatial ciphertext aggregation scheme with collaborative verification of fog nodes. Firstly, a cloud and fog collaboration architecture is constructed, where fog nodes are introduced to undertake the functions of data verification and slice reception, which reduces the computational cost of the sensing platform. Secondly, a multipath transmission method of slice data is advanced to realize the anonymous transmission of user identities. Then, combined with hash chain authentication, the integrity and authenticity of the sensing data are ensured. Finally, a privacy-protected spatial data aggregation method is presented. The interpolation method is adopted to predict the sample values of unknown locations in the sensing area, and the Paillier homomorphic encryption is used to ensure the privacy of the perceived data in this process. Security analysis and simulation results show that the solution can protect user privacy and security and reduce the computational cost of the sensing platform.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61901071, 61871062, 61771082, and U20A20157), General Project of Natural Science Foundation of Chongqing (cstc2019jcyj-msxmX0303), Science and Natural Science Foundation of Chongqing, China (cstc2020jcyj-zdxmX0024), University Innovation Research Group of Chongqing (CXQT20017), and Program for Innovation Team Building at Institutions of Higher Education in Chongqing (CXTDX201601020).

References

- [1] W. Feng, Z. Yan, H. Zhang, K. Zeng, Y. Xiao, and Y. T. Hou, "A survey on security, privacy, and trust in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2971–2992, 2017.
- [2] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, "A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 7, 2019.
- [3] L. Xiao, T. Chen, C. Xie, H. Dai, and H. V. Poor, "Mobile crowdsensing games in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1535–1545, 2017.
- [4] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
- [5] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in iiot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2019.
- [6] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An ai-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of iot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2019.
- [7] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5g wireless communications: a deep learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 227–236, 2018.
- [8] A. Thiagarajan, L. Ravindranath, K. LaCurts et al., "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings Of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 85–98, Berkeley CA, USA, November 2009.
- [9] Z. Zhang, P. Zhang, D. Liu, and S. Sun, "Srsm-based adaptive relay selection for d2d communications," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2323–2332, 2017.
- [10] C. De Capua, A. Meduri, and R. Morello, "A smart ecg measurement system based on web-service-oriented architecture for telemedicine applications," *IEEE Transactions on*

- Instrumentation and Measurement*, vol. 59, no. 10, pp. 2530–2538, 2010.
- [11] R. Morello, C. De Capua, and A. Meduri, “A wireless measurement system for estimation of human exposure to vibration during the use of handheld percussion machines,” *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 10, pp. 2513–2521, 2010.
- [12] M. A. Alsheikh, Y. Jiao, D. Niyato, P. Wang, D. Leong, and Z. Han, “The accuracy-privacy trade-off of mobile crowdsensing,” *IEEE Communications Magazine*, vol. 55, no. 6, pp. 132–139, 2017.
- [13] H. Amintoosi and S. S. Kanhere, “A reputation framework for social participatory sensing systems,” *Mobile Networks and Applications*, vol. 19, no. 1, pp. 88–100, 2014.
- [14] C. Luo, S. Guo, S. Guo, L. T. Yang, G. Min, and X. Xie, “Green communication in energy renewable wireless mesh networks: routing, rate control, and power allocation,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3211–3220, 2014.
- [15] G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B. Y. Zhao, “Defending against sybil devices in crowdsourced mapping services,” in *Proceedings Of the 14th Annual International Conference On Mobile Systems, Applications, and Services*, pp. 179–191, Singapore, June 2016.
- [16] Z. Yang, R. Wang, D. Wu, and D. Luo, “Utm: a trajectory privacy evaluating model for online health monitoring,” *Digital Communications and Networks*, vol. 7, 2020.
- [17] H. Sun, B. Dong, H. Wang, T. Yu, and Z. Qin, “Truth inference on sparse crowdsourcing data with local differential privacy,” in *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*, pp. 488–497, IEEE, Seattle, WA, USA, December 2018.
- [18] N. Wang, X. Xiao, Y. Yang et al., “Collecting and analyzing multidimensional data with local differential privacy,” in *Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 638–649, IEEE, Macao, China, April 2019.
- [19] B. Zhao, S. Tang, X. Liu, X. Zhang, and W.-N. Chen, “Ironm: privacy-preserving reliability estimation of heterogeneous data for mobile crowdsensing,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5159–5170, 2020.
- [20] Y. Tian, Z. Wang, J. Xiong, and J. Ma, “A blockchain-based secure key management scheme with trustworthiness in dwsns,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
- [21] D. Wu, Z. Yang, B. Yang, R. Wang, and P. Zhang, “From centralized management to edge collaboration: a privacy-preserving task assignment framework for mobile crowd sensing,” *IEEE Internet of Things Journal*, vol. 8, 2020.
- [22] D. Wu, R. Bao, Z. Li, H. Wang, H. Zhang, and R. Wang, “Edge-cloud collaboration enabled video service enhancement: a hybrid human-artificial intelligence scheme,” *IEEE Transactions on Multimedia*, vol. 23, 2021.
- [23] D. Wu, X. Han, Z. Yang, and R. Wang, “Exploiting transfer learning for emotion recognition under cloud-edge-client collaborations,” *IEEE Journal on Selected Areas in Communications*, vol. 39, 2020.
- [24] H. Wu, L. Wang, and G. Xue, “Privacy-aware task allocation and data aggregation in fog-assisted spatial crowdsourcing,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 589–602, 2019.
- [25] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. S. Shen, “Providing task allocation and secure deduplication for mobile crowdsensing via fog computing,” *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 581–594, 2018.
- [26] S. Basudan, X. Lin, and K. Sankaranarayanan, “A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing,” *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 772–782, 2017.
- [27] J. Chen, H. Ma, and D. Zhao, “Private data aggregation with integrity assurance and fault tolerance for mobile crowdsensing,” *Wireless Networks*, vol. 23, no. 1, pp. 131–144, 2017.
- [28] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, “Prisense: privacy-preserving data aggregation in people-centric urban sensing systems,” in *Proceedings of the 2010 Proceedings IEEE INFOCOM*, pp. 1–9, IEEE, San Diego, CA, USA, March 2010.
- [29] Q. Li and G. Cao, “Efficient and privacy-preserving data aggregation in mobile sensing,” in *Proceedings of the 2012 20th IEEE International Conference On Network Protocols (ICNP)*, pp. 1–10, IEEE, Austin, TX, USA, November 2012.
- [30] J. Fan, Q. Li, and G. Cao, “Privacy-aware and trustworthy data aggregation in mobile sensing,” in *Proceedings of the 2015 IEEE Conference on Communications and Network Security (CNS)*, pp. 31–39, IEEE, Florence, Italy, September 2015.
- [31] B. Agir, T. G. Papaioannou, R. Narendula, K. Aberer, and J.-P. Hubaux, “User-side adaptive protection of location privacy in participatory sensing,” *GeoInformatica*, vol. 18, no. 1, pp. 165–191, 2014.
- [32] S. Gisdakis, T. Giannetsos, and P. Papadimitratos, “Security, privacy, and incentive provision for mobile crowd sensing systems,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 839–853, 2016.
- [33] Q. Li and G. Cao, “Providing efficient privacy-aware incentives for mobile sensing,” in *Proceedings of the 2014 IEEE 34th International Conference On Distributed Computing Systems*, pp. 208–217, IEEE, Madrid, Spain, July 2014.
- [34] M. O’Keefe, “The paillier cryptosystem,” *Mathematics Department*, vol. 18, pp. 1–16, 2008.

Research Article

TASC-MADM: Task Assignment in Spatial Crowdsourcing Based on Multiattribute Decision-Making

Yunhui Li ¹, Liang Chang ², Long Li ², Xuguang Bao ², and Tianlong Gu ³

¹School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China

²Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

³College of Information Science and Technology, Jinan University, Guangzhou 510000, China

Correspondence should be addressed to Tianlong Gu; gutianlong@jnu.edu.cn

Received 7 May 2021; Accepted 9 August 2021; Published 21 August 2021

Academic Editor: Qing Yang

Copyright © 2021 Yunhui Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The methodology, formulating a reasonable task assignment to find the most suitable workers for a task and achieving the desired objectives, is the most fundamental challenge in spatial crowdsourcing. Many task assignment approaches have been proposed to improve the quality of crowdsourcing results and the number of task assignment and to limit the budget and the travel cost. However, these approaches have two shortcomings: (1) these approaches are commonly based on the attributes influencing the result of task assignment. However, different tasks may have different preferences for individual attributes; (2) the performance and efficiency of these approaches are expected to be improved further. To address the above issues, we proposed a task assignment approach in spatial crowdsourcing based on multiattribute decision-making (TASC-MADM), with the dual objectives of improving the performance as well as the efficiency. Specifically, the proposed approach jointly considers the attributes on the quality of the worker and the distance between the worker and the task, as well as the influence differences caused by the task's attribute preference. Furthermore, it can be extended flexibly to scenarios with more attributes. We tested the proposed approach in a real-world dataset and a synthetic dataset. The proposed TASC-MADM approach was compared with the RB-TPSC and the Budget-TASC algorithm using the real dataset and the synthetic dataset; the TASC-MADM approach yields better performance than the other two algorithms in the task assignment rate and the CPU cost.

1. Introduction

Spatial crowdsourcing, first introduced by Kazemi and Shahabiin [1], refers to an economic and efficient solution to participation in completing tasks, such as sensing tasks [2, 3]. The popularity of mobile devices and advanced Internet technologies have made it a popular trend in performing spatial tasks [4, 5]. Unlike conventional crowdsourcing, spatial crowdsourcing requires a worker to travel to a given location to perform a given task [6]. Examples of spatial crowdsourcing, such as environmental conditions and monitoring traffic flow at selected locations [7, 8], crowdsourcing news reporting tasks [9], and natural disaster response [10], are likely to have spatial requirements that cannot be fulfilled remotely and require physical arrival at the task's location. Spatial crowdsourcing is becoming a

compelling paradigm for recruiting workers to perform the tasks. However, due to the openness of crowdsourcing, there are some core issues: (1) how to guarantee the quality of crowdsourcing results and the number of tasks completed; (2) how to control the cost, such as the budget used and the travel cost; and (3) how to ensure the efficiency of task completed. All three core issues in spatial crowdsourcing are involved in task assignment. Thus, the task assignment is considered as the most fundamental challenge in spatial crowdsourcing [11].

The task assignment approach is mainly based on some attributes affecting the performance of task assignments [12], such as the distances between workers and tasks, and the qualities of workers. On one hand, previous research has shown that a task's distance from a worker affects the crowdsourcing outcomes [13, 14]. Tasks that are further

away from workers are less likely to be completed because workers tend to complete the nearby tasks. Some task assignment algorithms consider the locations of the tasks and workers to maximize the total number of assigned tasks [1, 14, 15]. On the other hand, the quality of a worker is considered to positively affect the crowdsourcing result's quality [16]. Some works evaluate a worker's quality by using distance and reputation and then propose a task allocation approach that balances the result's quality and the budget utilization rate [16, 17]. The existing task assignment method is based on the attributes that influence the results but ignores the different preferences of a task for different attributes. For example, monitoring traffic flow has strict requirements on the location of a worker, monitoring climate may accept workers working in a slightly larger area such as a city, and reporting news has limitations on both the location and quality of workers. Therefore, different tasks may have different attribute preferences.

Studies on spatial task assignment aim to allocate the task to suitable workers for different objectives, such as maximizing the total amount of assigned tasks [1, 18], minimizing the total travel cost of the allocated workers [16], and maximizing the overall quality of crowdsourcing results under the budget constraint [19]. Unfortunately, these objectives conflict with each other; optimizing multiple goals simultaneously is especially difficult. Consider the following examples: (a) increasing the total amount of tasks assigned is potentially achieved by relaxing the constraints on workers, such as increasing the accepted task region in which the task is allowed to be performed and lowering the threshold of workers' credibility. However, these methods may lead to increased travel costs or decreased quality of results. (b) One way to reduce the uncertainty of crowdsourced data is to ask multiple workers to complete the same task and then aggregate the responses of those workers to get the result of the task. However, asking multiple workers to complete the same task will increase the payment and the latency. Thus, a task assignment solution must involve a trade-off among various objectives.

In short, (1) the existing task assignment approach does not fully consider the task's attribute preference; (2) a specific task assignment approach usually achieves a certain objective, but fails to achieve several conflicting goals. To address the two problems, we propose a flexible and efficient task assignment approach in spatial crowdsourcing based on multiattribute decision-making (TASC-MADM), which takes into account the distance attribute and the reputation attribute simultaneously, as well as tasks' different preferences of attributes. Our goal is to trade off the quality of the result and the task allocation rate under the budget constraint of the task.

In this paper, we advance the key contributions of our research as follows:

- (i) Unlike existing work that simply focuses on some critical attributes while ignoring the preferences of different tasks, in this paper, we collectively consider the impact of distance attribute and worker quality attribute on the crowdsourcing result, as well as tasks' different preferences for attributes.

- (ii) We formulate the problem of task assignment in spatial crowdsourcing as a multiattribute decision-making (TASC-MADM) problem and propose a novel algorithm solving this problem. The linear weighted-evaluation method is used to rating the candidate workers comprehensively, which enables a task to select the appropriate worker according to its preference for attributes. Besides, the proposed approach allows achieving the objectives of maximizing the total amount of assigned tasks or the quality of outcome by setting attribute weights.
- (iii) Although our algorithm is simple, it performs well. Besides, it can be flexibly extended to the situation where any number of attributes affects the crowdsourcing result.

The rest of the paper is organized as follows. Section 2 presents related works on task assignment of spatial crowdsourcing. Section 3 formally defines problems involved in TASC-MADM. Section 4 describes the proposed TASC-MADM approach. The performance evaluation and discussion of the TASC-MADM approach are conducted in Section 5. Finally, Section 6 concludes the work and suggests some directions for future studies.

2. Related Work

Task assignment, i.e., the intelligent matching of tasks with the most appropriate workers, is a fundamental challenge of crowdsourcing [20–22]. Although there have been several studies on conventional crowdsourcing task allocation, they cannot be directly applied to spatial crowdsourcing, because the location of the spatial task and that of the workers are vital for the result of the spatial task assignment.

Research on spatial crowdsourcing task allocation is still in the early stage. In the spatial task assignment area, existing studies have mainly concentrated on exploiting the attributes of the tasks and workers. These attributes usually indicate the distance between the location of the workers and tasks, the capacity (i.e., the maximum number of tasks that a worker is willing/able to complete), and the quality of the workers [21, 23], etc. Kazemi and Shahabi [1] utilized the spatial region R (i.e., a rectangle region in which the worker accepts tasks) and the capacity of the workers $\max T$ to assign each worker to his nearby tasks. The greedy (GR) algorithm is presented to maximize the task assignment at each time instance. However, the greedy strategy cannot solve the global optimization problem. To solve this problem, heuristics are used to maximize the overall assignments. Hence, they proposed the second strategy: the least location entropy priority (LLEP) strategy. A location located in an area with few workers has low entropy. Conversely, a location located in a worker-density area has high entropy. Obviously, tasks with smaller location entropy are less likely to be completed by workers. In the heuristic, a higher priority is given to tasks located in areas with smaller location entropy. Furthermore, travel cost is a critical issue for spatial crowdsourcing. High travel costs may prevent workers from participating in the task and result in high costs for task requesters. Hence, they

proposed the third strategy: the nearest neighbor priority (NNP) strategy. Workers are assigned to tasks closer to them in preference, which aims at maximizing the overall finished tasks while reducing the workers' travel cost whenever possible. The research of [1] aims to maximize the number of task assignments while keeping the travel cost minimized, but they assume that the worker does not reject tasks assigned to them and trusts the workers to be reliable. Hassan and Curry [24] consider the situation where the worker can reject a task, propose a contextual bandit algorithm learning the possibility of task accepted by a worker, to assign a worker with high possibility based on the spatial locations of workers and tasks, and aim to maximize the total amount of successful assignments.

In addition to the locations of the tasks and workers and the capacity of workers, the quality of workers is another important attribute affecting the result of task allocation. Some works incorporate the quality of workers in the assignment process with the aim of controlling the quality [25] and cost for all completed tasks [16, 19, 26]. In traditional crowdsourcing, worker quality can be modeled by the worker's reputation [26–28], which may be a rating of the worker's past works, or an evaluation of the worker's knowledge, ability, confidence in completing tasks successfully, etc. A worker with a higher reputation is generally perceived to be better at his work. Cheng et al. [29] considered workers' confidence in completing tasks successfully and proposed the reliable-diversity-based spatial crowdsourcing approach. The approach Budget-TASC [16] considers the number of workers in the task assignment and thinks that the distance of a worker from the task negatively influences the quality of the crowdsourcing result [17, 18]; the reliability of the workers is given by a reputation function discounted by the distance. Then, the task is assigned to the worker with the highest reliability, to maximize the desired quality of results obtained, while the total budget is limited. However, the task assignment rate of spatial crowdsourcing tasks is not considered. RB-TPSC [17] presents a task package assignment algorithm with aim of maximizing the desired quality of the results from selected workers under a limited budget, improving the number over all spatial crowdsourcing tasks. Besides, Zhao et al. [30] thought that the quality of task accomplishment is mostly related to the worker's preference for the task category.

In this paper, to improve the performance of task assignment, we present a novel, efficient, and flexible approach by jointly considering multiattributes and preferences.

3. Problem Definition

In this section, we introduce some basic concepts of spatial crowdsourcing task assignment and then give the formal definitions. For the convenience of the following description, we firstly list the symbols used in this paper, see Table 1.

We consider there are a set of workers $W = \{w_1, w_2, w_i, \dots, w_m\}$ and a set of tasks $T = \{t_1, t_2, t_j, \dots, t_n\}$. Subscripts i and j are the worker ID and task ID, respectively. A worker w_i is represented as a tuple of the form $\langle l^i, r^i, q^i \rangle$, where $l^i = \langle \text{lon}^i, \text{lat}^i \rangle$ is the

location of the work i , lon^i and lat^i are the longitude and latitude of the worker i , respectively, and r^i represents the reputation of the worker i and q^i is the task quota of the worker i . A spatial crowdsourcing task t_j is represented as a tuple of the form, i.e. $t_j = \langle l^j, R^j, B^j \rangle$, where $l^j = \langle \text{lon}^j, \text{lat}^j \rangle$ is the location of the task j , which is represented by a longitude-latitude coordinate, lon^j and lat^j are the longitude and latitude of the task j , respectively, and $B^j \in \mathbb{R}^+$ is the limited budget of the task j .

Definition 1 (decision matrix for a task). Given a set of tasks T and a set of workers W . Let W^j be the set of workers within the region of the radius R of the task $t_j \in T$ and f be the number of attributes that are considered when assigning tasks. To facilitate the description of algorithms subsequently, the task and the worker are attached to the matrix columns. Then, the decision matrix of the task t_j is shaped as $DM_{n \times (f+2)}^j$, where $n = |W^j|$.

For example, the worker $w_i \in W^j$; the attributes involved are the distance and workers' reputation; then, the item $s^j = \langle j, i, d_{ji}, r^i \rangle$ is included in DM^j and d_{ji} represents the distance between t_j and w_i .

Setting the radius value of a task is one of the central aspects of task assignment. A very low radius value would result in a low task completion rate because of the lack of enough workers. In contrast, a very high radius value would have no practical significance because of the unavailability of workers willing to travel a long distance to perform a task. Previous studies suggest that the most acceptable distance for the workers is 0–2 km [1, 7, 13]. In practice, some workers may be tempted by the larger budget to perform remote tasks. In this paper, we assume (1) a task with a higher budget can select workers from a wider region and (2) some workers are willing to travel further for the higher rewards. So, the radius R^j is positively affected by the budget B^j and negatively affected by the extra allowance per kilometer β . Let a worker's accepted baseline distance be γ in the condition of a baseline payment P for a task. When a task's budget is less than the baseline payment P , the task is not likely to be accepted by any worker; at this situation, the radius is represented by a negative number, because no workers locate within a region of a task's negative radius. Then, the method in [16] is slightly changed to compute R^j :

$$R^j = \begin{cases} \gamma + \frac{B^j - P}{\beta}, & B^j \geq P, \\ -1, & B^j < P. \end{cases} \quad (1)$$

Identifying the value of the parameter P is not our emphasis. We focus on proposing a flexible spatial task assignment method that considers multiple attributes affecting the result in the task assignment and different requirement preferences for each attribute. Hence, we set P as the lowest budget among the budgets of all tasks.

Our goal is to select the worker with the highest rating of combined distance and reputation for a task. However, if the above decision matrix is directly used to determine the task

TABLE 1: List of symbols.

Symbol	Meaning
W	The set of workers, $W = \{w_1, w_2, \dots, w_i, \dots, w_m\}$
T	The set of tasks, $T = \{t_1, t_2, \dots, t_j, \dots, t_n\}$
i	A worker's number, $i \in \{1, 2, \dots, m\}$
j	A spatial task's number, $j \in \{1, 2, \dots, n\}$
l^i	The location of the worker, i
l^j	The location of the task, j
B^j	The budget of the task, j
R^j	The radius of the task, j
r^i	The reputation of the worker, i
d_{ji}	The distance from the task j to the worker, i
β	The extra allowance per kilometer
γ	Worker's accepted baseline distance
P	The baseline payment of a task
DM^j	The decision matrix of the task, j
E^j	The extra allowance for the task, j
p^j	The reward for the task, j
a^i	The amount of task assigned to the worker, i
q^i	The task quota of the worker, i
S	The result of task assignment

assignment, there are two problems. Firstly, the orders of magnitude of attributes are usually different, owing to the different natures of attributes. If the original value is directly used for rating the items, the role of the attribute with the higher value in the comprehensive rating will be highlighted, and the role of the attribute with the lower value will be relatively weakened. Secondly, the distance is a cost-type attribute and the reputation is a benefit-type attribute, which means the distance and the reputation have different influence trend on ratings. Therefore, in order to ensure the reliability of the rating results, it is necessary to normalize the original data. We adopt the linear proportional transformation method to normalize the distance and reputation, as shown in the following equations:

$$d_{ji} = \begin{cases} \frac{\max_{i \in W^j} d_{ji} - d_{ji}}{\max_{i \in W^j} d_{ji} - \min_{i \in W^j} d_{ji}}, & \max_{i \in W^j} d_{ji} \neq \min_{i \in W^j} d_{ji}, \\ 1, & \text{otherwise,} \end{cases} \quad (2)$$

$$r^i = \begin{cases} \frac{r^i - \min_{i \in W^j} r^i}{\max_{i \in W^j} r^i - \min_{i \in W^j} r^i}, & \max_{i \in W^j} r^i \neq \min_{i \in W^j} r^i, \\ 1, & \text{otherwise.} \end{cases} \quad (3)$$

Definition 2 (reward for a task). When a task is completed, the requester must offer rewards to the corresponding worker. Let E^j represent the extra allowance for the task j when it is completed by a worker without γ . Then, the reward of the task j is expressed in the following equation:

$$p^j = P + E^j, \quad (4)$$

where E^j is related to the parameter β , d_{ji} , γ . The farther a worker travels, the more extra allowance he should get. If the worker w_i completes the task t_j within the accepted baseline distance γ , then E^j equals to 0; otherwise, E^j is proportional to the extra distance and the extra allowance, so E^j is computed in the following equation [16]:

$$E^j = \begin{cases} \beta(d_{ji} - \gamma), & d_{ji} > \gamma, \\ 0, & d_{ji} \leq \gamma. \end{cases} \quad (5)$$

The parameter d_{ji} mentioned above involves the location of the task and the worker. We computed d_{ji} from the task t_j to the worker w_i by the Haversine formula [31]:

$$d_{ji} = R \times 2 \times \arcsin \left(\left(\sin^2 \left(\frac{\text{lat}^j - \text{lat}^i}{2} \right) + \cos(\text{lat}^j) \times \cos(\text{lat}^i) \times \sin^2 \left(\frac{\text{lon}^j - \text{lon}^i}{2} \right) \right)^{1/2} \right), \quad (6)$$

where R refers to the earth's radius [12].

Definition 3 (TASC-MADM problem). Given a set of tasks T and a set of workers W , we assume that each task is

assigned to the optimal worker. Let $S = \{s^1, s^2, \dots, s^j, \dots, s^n\}$ represent the selected workers for all tasks, $s^j = \langle j, i, d_{ji}, r^i \rangle$, $s^j \in DM^j$, and TASC-MADM is to find s^j such that the following linear combination is optimized:

$$\max_{i \in W^j}(\text{score}_{ji}) = \max_{i \in W^j}(w_0 \times d_{ji} + w_1 \times r^i), \quad (7)$$

where j and i are the matching task-worker pair; and $w_v \in [0, 1]$ is the attribute importance parameter, $\sum_{v \in \{0,1\}} w_v = 1$.

The weights are usually determined objectively or subjectively. The entropy method is generally used to objectively get weighting of every attribute [32]. However, we hope that the task assignment operation can set the attribute weight to achieve different optimization goals. So, the weight values are set according to the requirement preference of the task for attributes. If a task has more requirement preference for distance more than that for reputation, it can set $w_0 > 0.5$. Otherwise, $w_0 < 0.5$. The setting, $w_0 = 0.5$, implies the following situation: among the workers with the same reputation, the worker with a shorter distance has the priority to be selected. Similarly, among the workers with the same distance, the worker with a higher reputation has the priority to be selected.

3.1. Complexity Analysis. One task assignment of a single task is to find the best worker among m workers; it needs to repeat the assignment $n \times m$ times to complete all task assignments, so the problem is solvable in polynomial time.

4. Assignment Protocol

We assume that the workers querying the tasks are willing to accept the tasks. Thus, assigning a task to a worker means selecting the best worker with the highest comprehensive rating of distance and reputation. In this section, we will elaborate on our spatial task assignment algorithm.

4.1. Preparing Decision Matrix. Preparing the task's decision matrix involves two steps. Firstly, we obtain the decision matrix DM^j for the task j . Each item of the decision matrix represents a candidate worker. In this paper, the attributes, affecting the result of a task assignment, include the reputation of a worker and the distance from a task to a worker. The workers within the radius of the task j are a part of the task's candidates, excluding the workers who are assigned tasks more than their quota (Line 2–4). Secondly, normalize the decision matrix DM^j by equations (2) and (3) (Line 5). The pseudocode, obtaining and normalizing the decision matrix DM^j for the task j , is given by Algorithm 1.

The computational complexity of Algorithm 1 depends on the loop operation and the normalization operation. The complexities of these two operations are $O(m)$ and $O(|W^j|)$, respectively. Since $|W^j|$ is usually much less than $|m|$, the total computational complexity is $O(m)$.

4.2. TASC-MADM Approach. As mentioned in Section 3, a spatial task should be assigned to the worker with the highest rating, which optimizes the linear combination of the distance and the reputation.

Algorithm 2 is the pseudocode of the allocation method, namely, TASC-MADM algorithm, which inputs a set of workers W , a set of tasks T , and the parameters P, β, γ and returns the best assignment result S , containing task-and-worker assignment with the highest rating.

Initially, S is set to empty (Line 1), $a^i_{i=\{1,2,\dots,m\}} = 0$ (i.e., each worker has been assigned zero times) (Line 2). Next, for each task t_j , calculate the radius and the distance to the workers and obtain and normalize the decision matrix DM^j by calling Algorithm 1 (Line 6). Next, if the decision matrix has more than zero items, compute the scores of items (Line 9) and, simultaneously, compute the reward p^j paid by the task t_j to the worker w_i (Line 10). For subsequent easy operation, the item's score and the reward are associated with other information including the task number, the worker number, the distance, and the reputation (Line 11). Next, we sort the items descending by scores (Line 12). Intuitively, the item with a higher rank indicates a better assignment than other assignments. Finally, the task t_j is assigned to the topmost worker (Line 13–17).

The assignment iterates for n rounds (Line 3) and finally returns the assignment result of all tasks (Line 18). In each iteration, the computational complexity is $O(m)$, so the total computational complexity is $O(n \times m)$.

5. Experiment Evaluation

In this section, we tested the performance of our approach on both real and synthetic data.

5.1. Metrics. In the experiments, we measured the performance of each approach according to the following metrics [23]:

- (1) Average task radius (δ): this metric measures the average spatial region size of the tasks, which is computed as the average radius of all tasks.

$$\beta = \frac{1}{|T|} \sum_j R^j. \quad (8)$$

- (2) Task assignment rate (η): this metric measures the algorithm's effectiveness in assigning several tasks successfully. The task assignment rate η is the percentage of assigned tasks among the total amounts of crowded spatial tasks.

$$\eta = \frac{|S|}{|T|}. \quad (9)$$

- (3) Average reputation of workers assigned tasks (ψ): the quality of crowdsourcing result is determined by the worker's quality, which is modeled by the worker's reputation. So, this metric evaluates the quality of tasks completed. It is computed as the total reputation of selected workers divided by the number of selected workers.

Input: a set W , the task j
Output: the normalized decision matrix DM^j for the task j

- (1) $DM^j =$
- (2) For each $w_i \in W$:
- (3) If $d_{ji} \leq R^j$ and $a^i < q^i$:
- (4) $s_i^j = \langle j, i, d_{ji}, r^i \rangle$, $DM^j \leftarrow DM^j \cup \{s_i^j\}$
- (5) Normalize DM^j by equations (2) and (3)
- (6) Return DM^j

ALGORITHM 1: Obtain the decision matrix for a task.

Input: a set of workers W , a set of tasks T , parameters P, β, γ
Output: S , which is the result of task assignment

- (1) $S =$
- (2) $a^i = 0$, $i \in \{1, 2, \dots, m\}$
- (3) For each $t_j \in T$:
- (4) Calculate the radius R^j
- (5) Calculate d_{ji} from the task j to each worker i
- (6) Obtain and normalize the decision matrix DM^j by calling Algorithm 1
- (7) If $|DM^j| > 0$:
- (8) For i' in range ($|DM^j|$):
- (9) Compute the score $_{ji}$ using equation (7) // $i = DM^j[i].i$, it is the number of the i' th worker for the task t_j .
- (10) Compute the reward p^j using equation (4).
- (11) $s_i^j = s_i^j.append(\langle p^j, score_{ji} \rangle)$ (i.e. $s_i^j = \langle j, i, d_{ji}, r^i, p^j, score_{ji} \rangle$)
- (12) Sorted DM^j descending by scores
- (13) For each item in DM^j :
- (14) If j in item:
- (15) $s^j = \text{item}$, $a^i + = 1$
- (16) $S \leftarrow S \cup \{s^j\}$
- (17) Break
- (18) Return S

ALGORITHM 2: TASC-MADM algorithm.

$$\psi = \frac{1}{|S|} \sum_{i \in S} r^i. \quad (10)$$

$$\omega = \frac{1}{|S|} \sum_{j \in S} p^j. \quad (13)$$

- (4) Average distance traveled (ζ): this metric measures the travel cost for workers when they complete the assigned tasks; it is computed as the average distance traveled by all the selected workers.

$$\zeta = \frac{1}{|S|} \sum_{\langle j, i \rangle \in S} d_{ji}. \quad (11)$$

- (5) Average budget utilization rate (ϕ): this metric is the average of the budget utilization for all assigned tasks. The budget utilization rate of each assigned task is the ratio of the actual reward paid for the task to the budget of that task.

$$\phi = \frac{1}{|S|} \sum_{j \in S} \frac{p^j}{B^j}. \quad (12)$$

- (6) Average reward (ω): this is the ratio of the total reward for all the assigned tasks to the number of assigned tasks.

The parameter δ is used to demonstrate the changing of other metrics along with the average radius. The workers' perspective would prefer to keep ζ as low as possible. The task requesters' perspective would prefer higher values of η and ψ , but lower values of ϕ and ω .

5.2. Experimental Setting

5.2.1. Datasets

- (i) *Real Dataset.* We used a real dataset [33] from a crowdsourcing event by taking photos with the location information for 1877 workers and 835 tasks. These tasks and workers were mainly obtained from four cities in China: Foshan, Guangzhou, Shenzhen, and Dongguan.
- (ii) *Synthetic Dataset.* The synthetic data were obtained from a single day dataset from Gowalla in 2010, which included 10956 tasks and 5087 workers

located in America. This dataset contains four attributes as follows: task ID, task's location, worker ID, and worker's location. Reputations are generated following the uniform distribution of $0 \sim 20000$. Budgets are generated following the uniform distribution of $65 \sim 85$. The task quotas of workers are generated following the uniform distribution of $5 \sim 10$.

5.2.2. Compared Algorithms. The RB-TPSC and Budget-TASC algorithms were selected as the baseline algorithms because they are most closely related to TASC-MADM.

- (i) RB-TPSC is a task package assignment method, which aims at maximizing the number of tasks assigned within budget constraints. Quality of results and travel costs are also being considered.
- (ii) Budget-TASC is a budget-aware spatial crowdsourcing task assignment method, which aims in maximizing the total quality of tasks completed within budget constraints.

We compared our TASC-MADM with RB-TPSC and Budget-TASC on both real and synthetic datasets.

5.3. Results on Real Dataset. The first three experiments compared TASC-MADM and RB-TPSC under different settings of parameters: β , γ , and B^j . The results are shown in Figures 1–3. For each parameter, the experiment evaluated the average metrics of the two algorithms under 21 different settings. We set the lowest budget of all tasks as $P = 65$. Moreover, for the fairness in considering the effect of distance and reputation, the weight $w_0 = 0.5$.

First, we compared TASC-MADM and RB-TPSC under different β values. The results are shown in Figure 1, where the extra allowance per kilometer β varies from $0 \sim 20$ monetary units in 1-unit increments, $\gamma = 0.5$, and B^j is obtained from the dataset. The horizontal axis represents the different settings of β , while the vertical axis represents the different values of the first six metrics. As can be seen, the average task radius δ is affected by the extra allowance per kilometer β (Figure 1(a)). δ is maximized when $\beta = 1$, increases with β if $\beta < 1$, and decreases with β if $\beta > 1$. The change in the trend of other metrics is consistent with the average task radius. This is because, in a region with a smaller radius, it is usually impossible to find enough workers with high reputations. Thus, other metrics decrease with the average radius. Compared with RB-TPSC, TASC-MADM ensures a high average task assignment rate (Figure 1(b)), but reduces the average travel cost of workers (Figure 1(d)) and saves the budget (Figures 1(e) and 1(f)). Besides, our result shows that the average reputation of workers decreases with the average radius (Figure 1(c)), which is more realistic because there are fewer workers to choose from.

Secondly, we compared TASC-MADM and RB-TPSC under different γ values. Figure 2 depicts the trend in which the above six metrics change when γ , the accepted distances without extra remote allowance, varies from $0 \sim 2$ km, where

$\beta = 2$, and B^j is obtained from the real dataset. If $\gamma > 0.5$, the average radius increases with γ and positively affects the abovementioned metrics, except for ϕ and ω . Our proposed method achieves a task assignment rate value of 98.56 (Figure 2(b)), but the maximum of the average distance traveled the average reward, and the average budget utilization rates are $\zeta = 1.03$ km, $\omega = 65.33$, and $\phi = 94.93$, respectively (Figures 2(d)–2(f)). Compared to the RB-TPSC method, our method greatly decreases the average distance traveled by the workers and the average reward offered by the tasks' requester, while maintaining an equally high or a greater average task assignment rate. Moreover, it significantly improves the quality of crowdsourcing results because of the average reputation value of the selected workers increasing by about 250% (Figure 2(c)).

The third experiment compared TASC-MADM and RB-TPSC under different budgets (B). Figure 3 depicts how the first six metrics change while the tasks' budget (B) varies from 90% to 110% in 1% increments, where $\beta = 2$, $\gamma = 0.5$. Since the task with a greater budget should have more workers to choose from, the average radius of the tasks is positively affected by the tasks' budget (Figure 3(a)). As the task radius increases further, the task assignment rate increases to 99.5% (Figure 3(b)). Compared to the RB-TPSC method, our method decreases the average distance traveled from a range of 1.4~2.5 km to a range of 0.8~1.5 km (Figure 3(d)), but increases the average reputation by about 147% (Figure 3(c)), and greatly decreases the average budget utilization rate (Figure 3(e)) and the average reward ((Figure 3(f))). More importantly, with our method, the greater the number of candidate workers, the lower the average reward offered by the requester, and the higher is the quality of the crowdsourcing result. In contrast, RB-TPSC increases the average reward continuously and decreases the average reputation to a stable value of around 500.

The fourth experiment compared TASC-MADM with RB-TPSC and Budget-TASC when radiuses were changed. For TASC-MADM and RB-TPSC, we selected the average metrics of different γ values. In the above experiment, the computed radius belongs to the interval $[0, 12]$, when γ varied from $0 \sim 2$ km and other parameters were fixed. We have experimented with Budget-TASC in different radiuses varying from $0 \sim 12$ km and then get the average of each metric. For the Budget-TASC, the other parameters were set as follows: D_C is set as the earth's radius [10], $P_L = 0$, P_H is set as the budget of the task, and P_M is set as the half of the budget of the task. Because we scaled the worker's reputation to the interval $[0, 1]$, $Th_{HM} = 0.75$, $Th_{ML} = 0.5$.

The results are shown in Table 2. TASC-MADM outperforms the baseline algorithms in all metrics, except the average reputation. For Budget-TASC, P_H is set as the task's total budget, which limits the task to high-quality workers, but also increases the average budget utilized.

5.4. Results on Synthetic Dataset. We repeated the first three experiments in Section 5.3 on the synthetic dataset. Figures 4–6 illustrate the results of TASC-MADM and RB-TPSC. Compared with RB-TPSC, our algorithm greatly

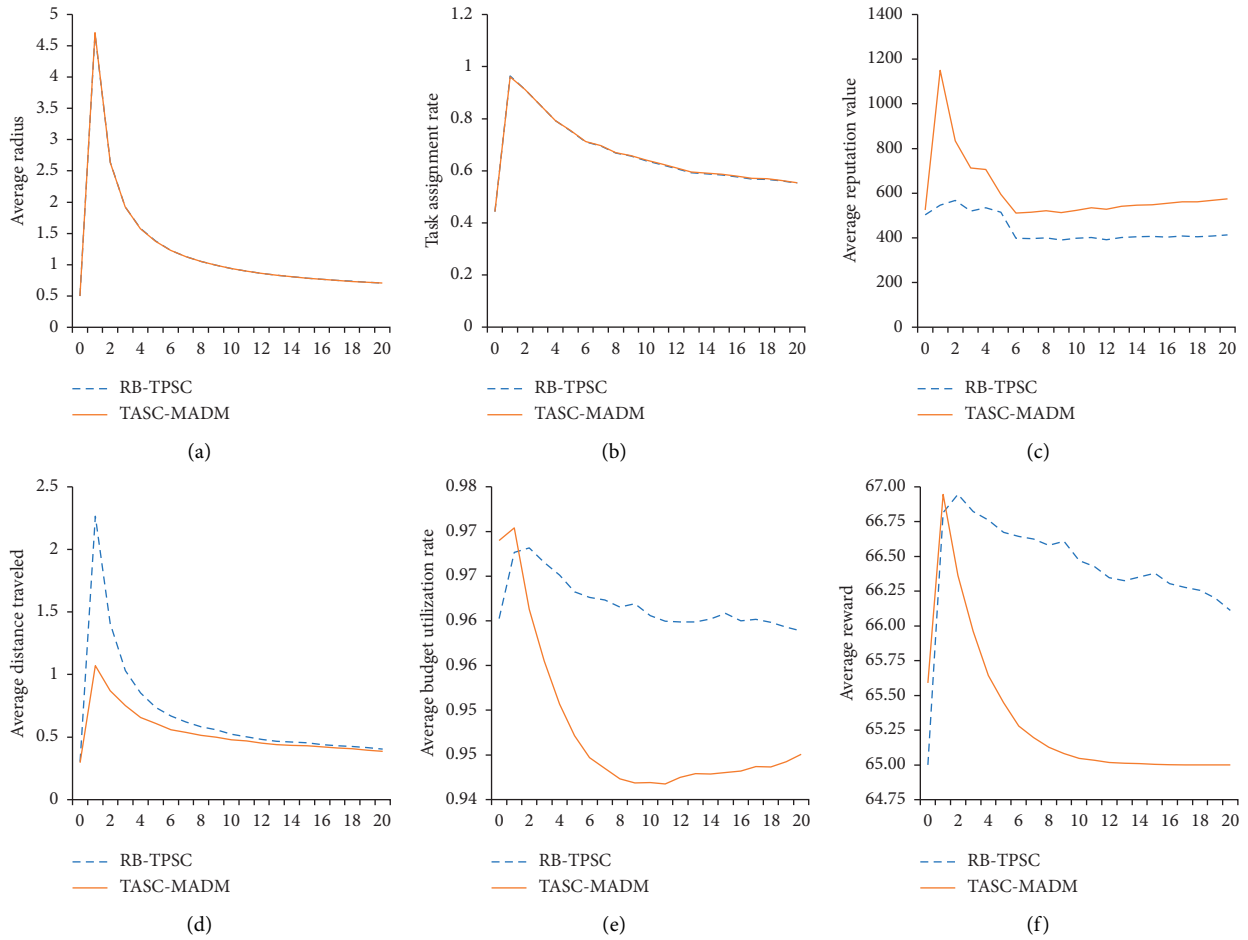


FIGURE 1: Performance with an extra allowance per kilometer (β) (monetary unit), real dataset.

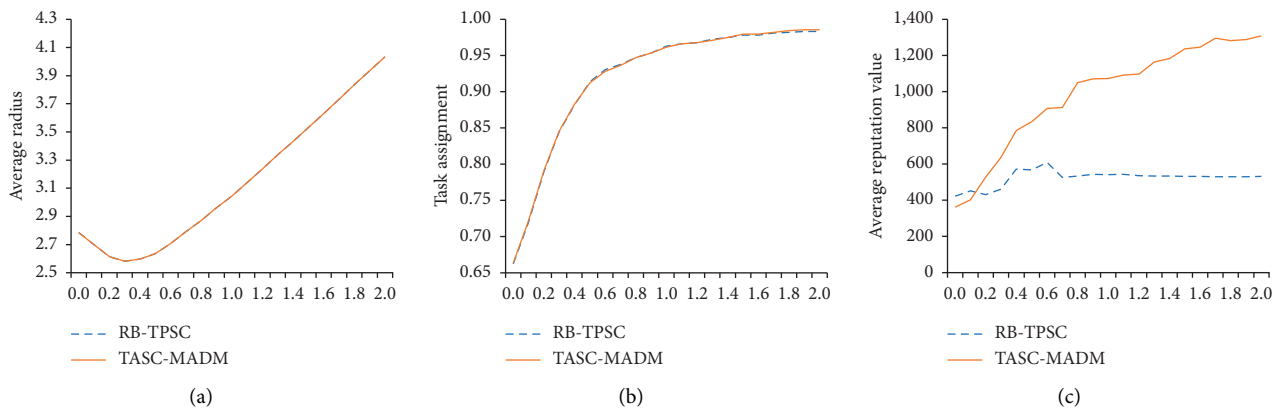


FIGURE 2: Continued.

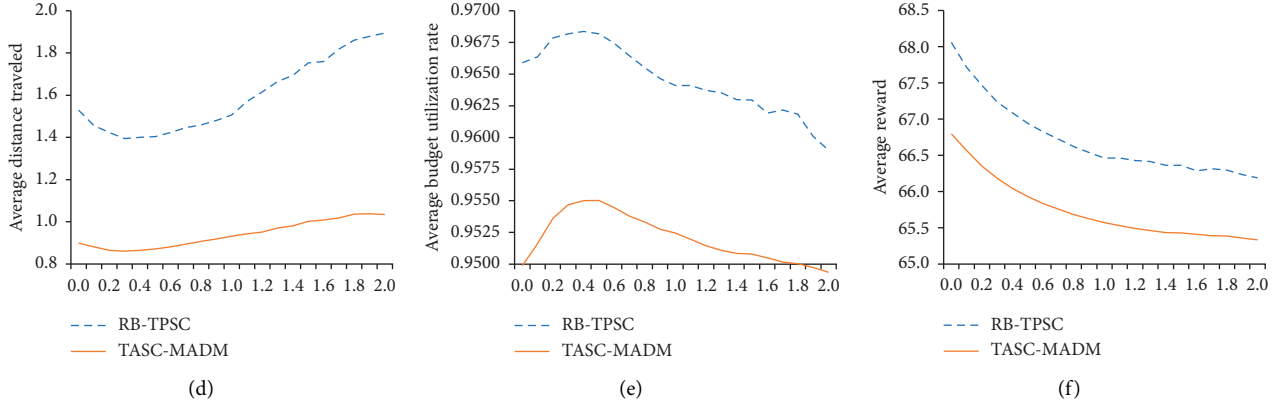
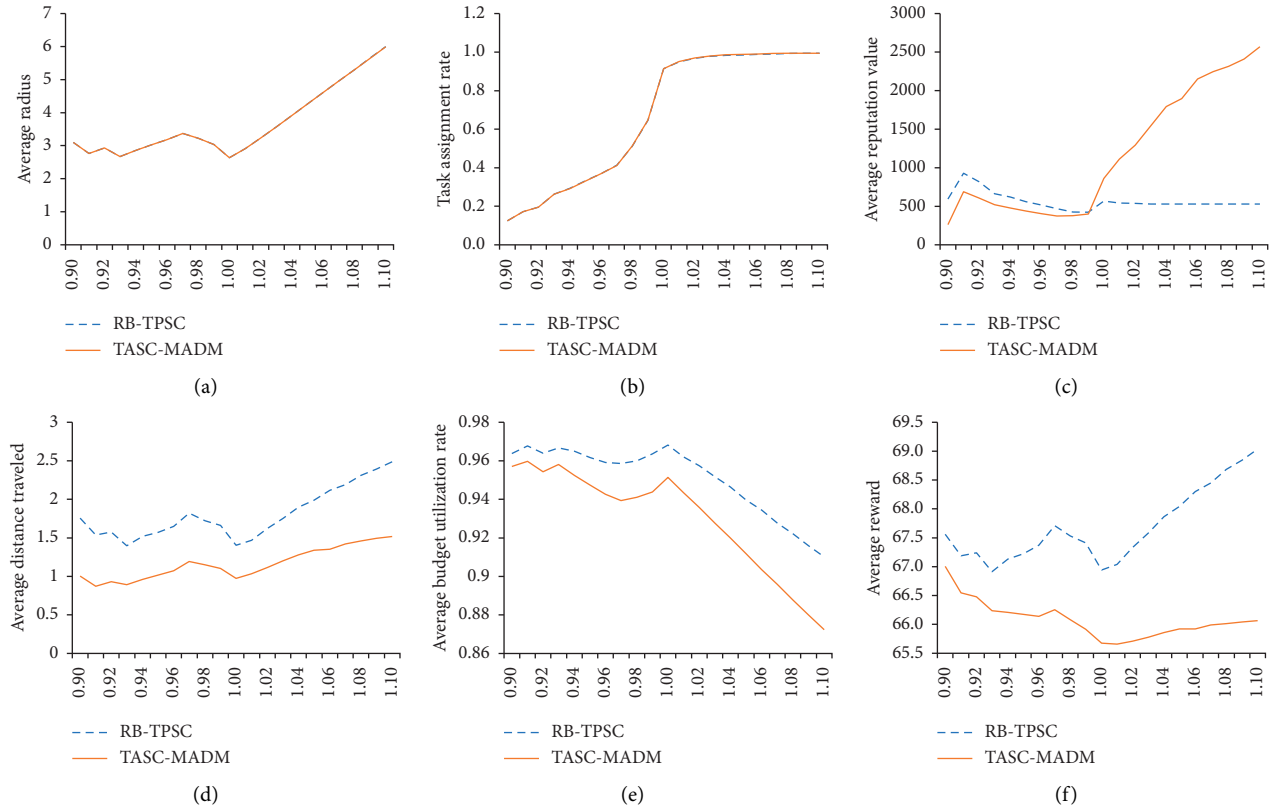

 FIGURE 2: Performance with different accepted distances without extra remote allowance (γ) (km), real dataset.

 FIGURE 3: Performance with an amplitude of variation in budget (B), real dataset.

TABLE 2: Results under different approaches, real dataset.

Algorithms	η (%)	ψ	ζ	ϕ	ω
Budget-TASC	85.67	1390.63	1.4	1	77.61
RB-TPSC	93.95	523.39	1.59	0.9645	66.72
TASC-MADM	94.18	987.46	0.78	0.9509	65.98

improves the task assignment rate (Figures 4(b), 5(b), and 6(b)) and the result's quality (Figures 4(c), 5(c), and 6(c)). However, for the synthetic dataset, the task locates in a

slightly sparse scenario with few workers. Then, the higher task assignment rate implies the higher travel cost (Figures 4(d), 5(d), and 6(d)). Correspondingly, the budget used is increased (Figures 4(e), 4(f), 5(e), 5(f), 6(e), and 6(f)).

Next, based on the synthetic dataset, we compared TASC-MADM with RB-TPSC and Budget-TASC when radiuses were changed. For TASC-MADM and RB-TPSC, we selected the average metrics of different γ values; the computed radius approximately belongs to the interval [6, 9]

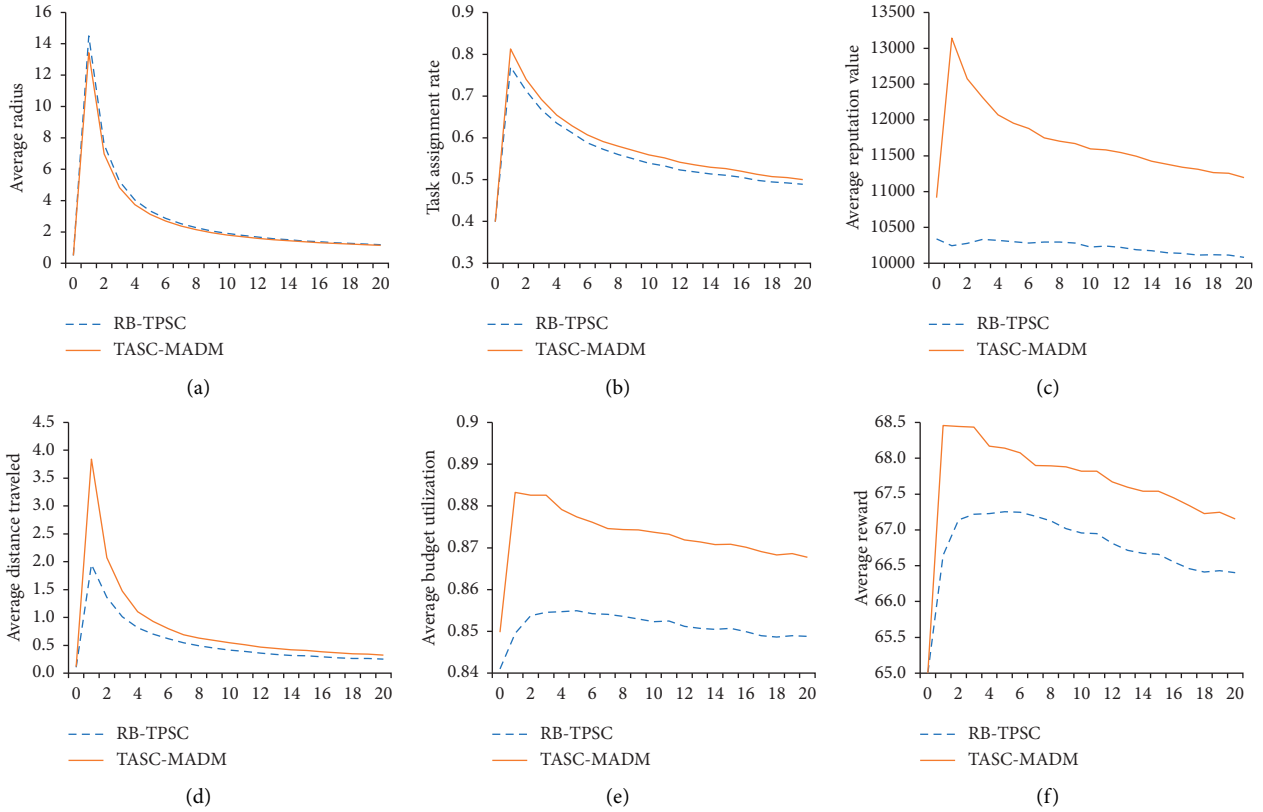


FIGURE 4: Performance with an extra allowance per kilometer (β) (monetary unit), synthetic dataset.

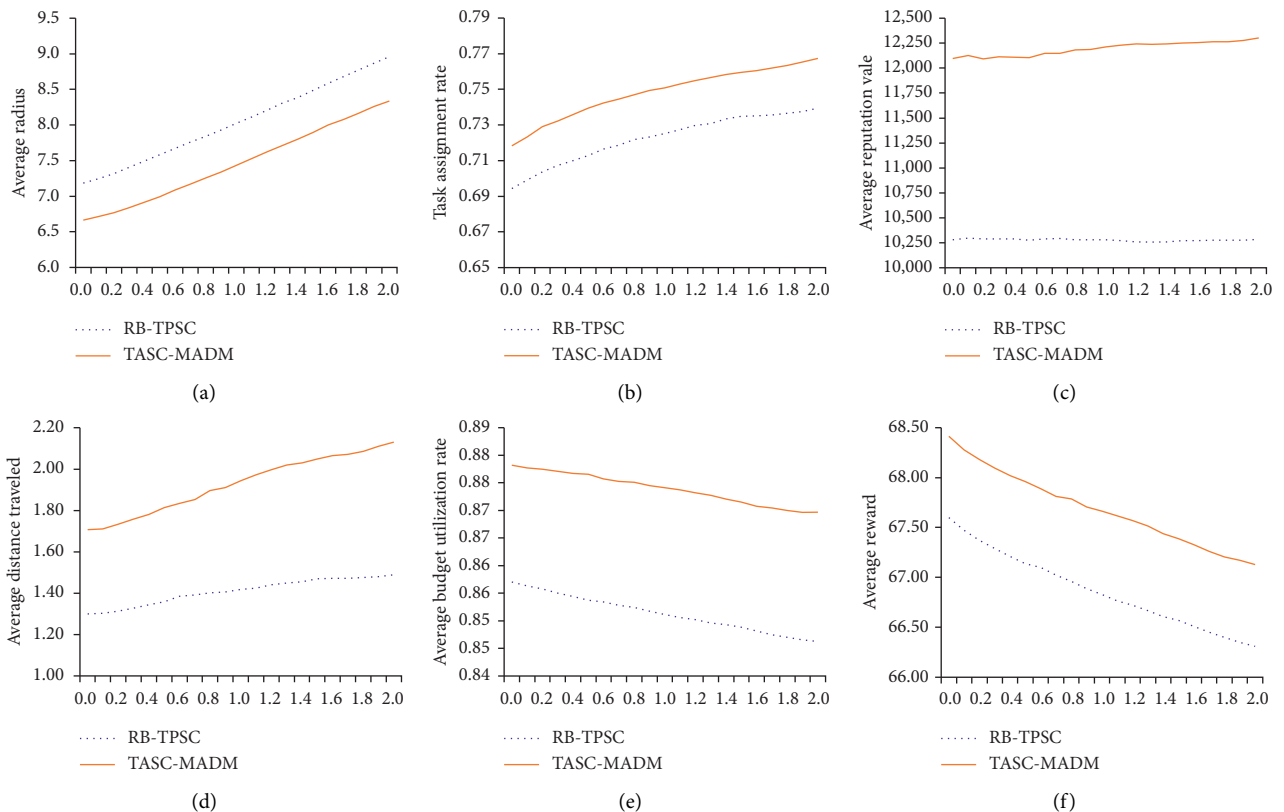


FIGURE 5: Performance with different accepted distances without extra remote allowance (γ) (km), synthetic dataset.

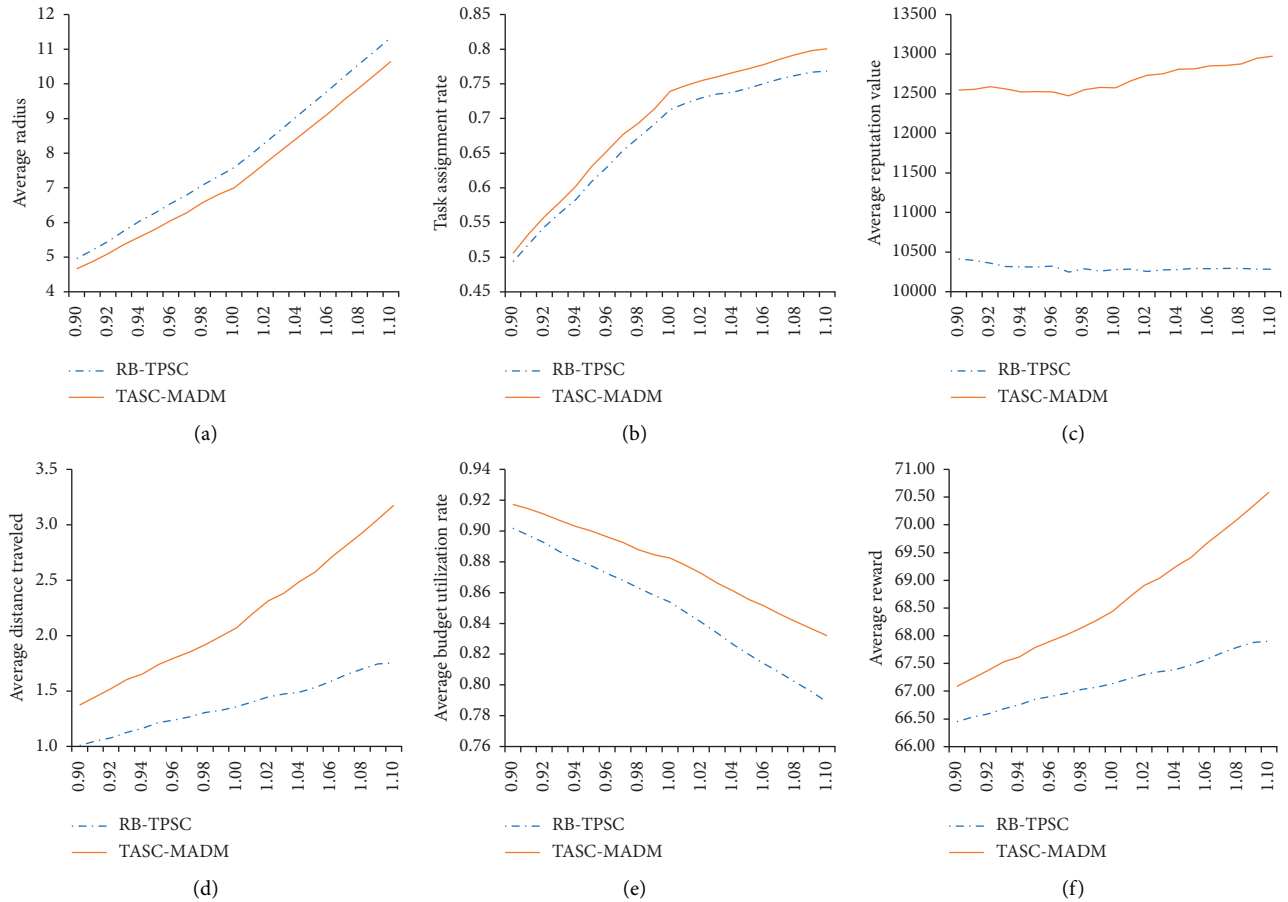


FIGURE 6: Performance with an amplitude of variation in budget (B), synthetic dataset.

(Figure 5(a)). So, for Budget-TASC, the radius varied from 6 ~ 9 km; other parameters were set as in the fourth experiment.

The experimental results (Table 3) show that TPSC-ADM obtains the highest task assignment rate, but Budget-ASC achieves the highest quality and RB-TPSC spends the lowest budget.

5.5. Effect of Task's Attribute Preference. Our approach, TASC-MADM, enables the task to be assigned to satisfy demands for different goals. If the result's quality is the primary goal, the task should select workers with a high reputation. However, if saving cost is the most concerned objective, the worker closer to the task should be chosen first. Figure 7 shows the influence of tasks' attributes preferences on the proposed approach's performance. $w_0 = 0.5$, all targets are considered fairly. $w_0 < 0.5$, the task prefers the worker's reputation to the distance; the result's quality is significantly improved. However, the travel cost and budget utilized are increased. When $w_0 > 0.5$, the travel cost and the budget utilized are saved, whereas the result's quality is decreased. Our approach can maximize task assignment rate by setting $w_0 = 1$, or maximize the quality of workers selected by setting $w_0 = 0$.

5.6. Efficiency of Algorithms. The efficiency of the algorithms is measured in CPU cost. The Budget-TASC algorithm's computational complexity is $O(n \times m^2)$, and that of RB-TPSC and TASC-MADM is $O(n \times m)$. we compare TASC-MADM with RB-TPSC and Budget-TASC in CPU cost. Each of the programs runs 3×21 rounds. The average time per round is used to measure the algorithm's efficiency. As shown in Figure 8, our approach significantly improves the efficiency of spatial task assignment.

5.7. Summary of Experiment Results. We summarized the major finding as follows:

- (i) If the task is located in a worker-density area, the proposed TASC-MADM approach exhibits better results than RB-TPSC. It also performs better than Budget-TASC on the metrics, except for the quality of workers.
- (ii) If the task is located in a worker-sparsity area, the proposed TASC-MADM approach performs better than RB-TPSC in terms of the average assignment rate and quality of workers. But it leads to more travel cost and budget utilized. Budget-TASC obtains the best quality because it considers the quality first.

TABLE 3: Results under different approaches, synthetic dataset.

Algorithms	η	ψ	ζ	ϕ	ω
Budget-TASC	0.72743975	13672.17	1.417697	1	77.58585
RB-TPSC	0.72253468	10278.82	1.408169	0.851329	66.86784
TPSC-ADM	0.74825306	12194.53	1.926809	0.873916	67.68671

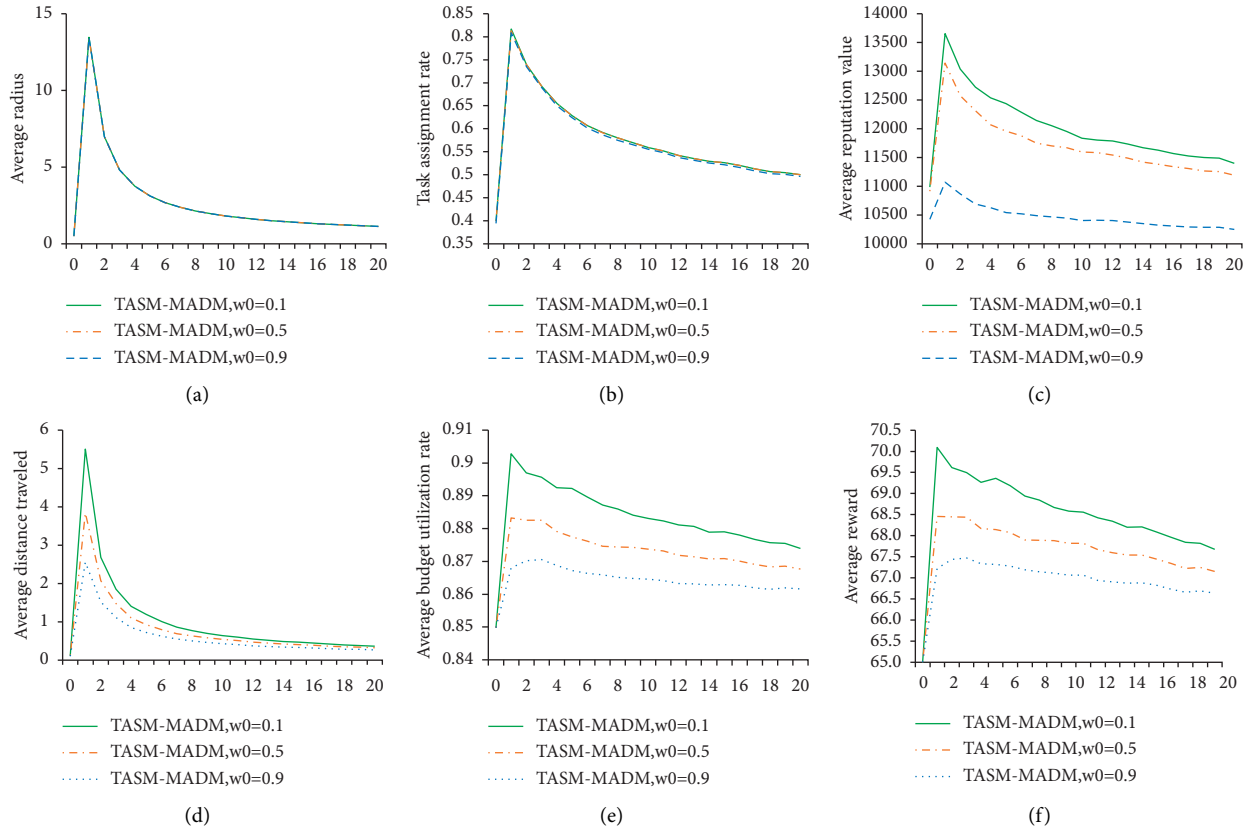


FIGURE 7: Effect of tasks' attributes preferences on the TASC-MADM approach. Performance with an extra allowance per kilometer (β) (monetary unit), synthetic dataset.

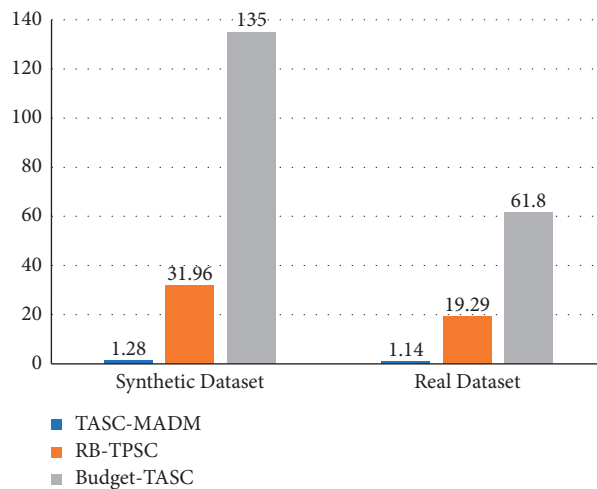


FIGURE 8: Average CPU cost.

- (iii) In the terms of CPU cost, the proposed TASC-MADM approach is superior to the baseline algorithms.

5.8. *Discussion.* This section discusses the advantages and limitations of the TASC-MADM approach. The advantages are listed as follows:

- (1) *Effective.* The TASC-MADM approach improves the task assignment rate. By setting different attribute weights, it can maximize the task assignment rate or the quality of crowdsourcing results.
- (2) *Efficient.* The TASC-MADM approach enhances efficiency because of computing simply.
- (3) *Extensible.* Theoretically, our method can be extended to solving decision problems involving any number of attributes.

There still exist the following limitations in the TASC-MADM approach.

- (1) *Quality Quantification of Workers.* In this paper, the worker quality is modeled as a reputation value, to reflect the quality of crowdsourced results. In practice, the same worker's quality may differ in specific tasks. How to quantify the quality of workers is not covered in this paper.
- (2) *Efficiency of Indexing Records.* The TASC-MADM approach exhaustively searches all the records to identify the candidate of a task, which makes it less efficient to get the decision matrix on large datasets.

6. Conclusion

This paper focuses on designing an efficient task assignment approach, which can deal with the situation where tasks have different require preferences for different task attributes, to achieve different goals. Our task assignment approach can be extended to scenarios containing any number of attributes. In addition to the distance and reputation, other criteria such as the workers' skills can be considered.

As for future work, more factors, such as workers' willingness to accept tasks and their quality differences in different professional fields, are included in task assignments. In addition, improving the efficiency of indexing records to make the allocation scheme suitable for large datasets is a valuable research topic.

Data Availability

The data of allocated sharing tasks are available from China Society for Industrial and Applied Mathematics, 2017, retrieved on September 2, 2020, from http://www.mcm.edu.cn/html_cn/node/460baf68ab0ed0e1e557a0c79b1c4648.html.

Conflicts of Interest

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

Acknowledgments

The authors thank researchers working at Guangxi Key Laboratory of Trusted Software, especially Xuguang Bao and Manli Zhu, for their suggestion during the research and preparation of the manuscript. This work was funded by Guangxi Key Laboratory of Trusted Software (No. kx201727), Project to Improve the Scientific Research Basic Ability of Middle-Aged and Young Teachers (No. 2019KY0226), Natural Science Foundation of China (Nos. 61966009, U1811264, and U1711263), and Natural Science Foundation of Guangxi Province (Nos. 2019GXNSFBA245049, 2019GXNSFBA245059, and 2018GXNSFDA281045).

References

- [1] L. Kazemi and C. Shahabi, "GeoCrowd: enabling query answering with spatial crowdsourcing. GIS," in *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, pp. 189–198, CA, USA, November 2012.
- [2] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.
- [3] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.
- [4] DiDi, *Didi Chuxing Corporate Citizenship Report*, <https://www.didiglobal.com/aboutdidi/responsibility>, 2017.
- [5] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2019.
- [6] C. Zhao, P. Cheng, L. Chen, X. Lin, and C. Shahabi, "Fair task assignment in spatial crowdsourcing," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 2479–2492, 2020.
- [7] M. Venzani, A. Rogers, and N. R. Jennings, "Crowdsourcing spatial phenomena using trust-based heteroskedastic Gaussian processes," in *Proceedings of the First Conference on Human Computation and Crowdsourcing (HCOMP)*, pp. 182–189, Palm Springs, CA, USA, November 2013.
- [8] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
- [9] B. Van, B. V. D. Haak, M. Parks, and M. Castells, "The future of journalism: networked journalism rethinking journalism in the networked digital age," *International Journal of Communication*, vol. 6, pp. 2923–2938, 2012.
- [10] M. Zook, M. Graham, S. Taylor, and S. Gorman, "Volunteered geographic information and crowdsourcing disaster relief: a case study of the Haitian earthquake," *World Medical Health Policy*, vol. 2, no. 2, pp. 7–33, 2010.
- [11] Y. Tong, L. Chen, and C. Shahabi, "Spatial crowdsourcing: challenges, techniques, and applications," *Proceedings of the VLDB Endowment*, vol. 10, pp. 1988–1991, 2017.
- [12] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, "A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2347–2360, 2020.

- [13] F. Alt, A. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, "Location-based crowdsourcing: extending crowdsourcing to the real world," in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction*, pp. 13–22, Reykjavik, Iceland, October 2010.
- [14] L. Kazemi, C. Shahabi, and L. Chen, "GeoTruCrowd: trustworthy query answering with spatial crowdsourcing," in *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, pp. 314–323, Orlando, Florida USA, November 2013.
- [15] U. ul Hassan and E. Curry, "Efficient task assignment for spatial crowdsourcing: a combinatorial fractional optimization approach with semi-bandit learning," *Expert Systems with Applications*, vol. 58, pp. 36–56, 2016.
- [16] C. Miao, H. Yu, Z. Shen, and C. Leung, "Balancing quality and budget considerations in mobile crowdsourcing," *Decision Support Systems*, vol. 90, pp. 56–64, 2016.
- [17] P. Wu, E. W. T. Ngai, Y. Wu, and Y. Wu, "Toward a real-time and budget-aware task package allocation in spatial crowdsourcing," *Decision Support Systems*, vol. 110, pp. 107–117, 2018.
- [18] H. To, C. Shahabi, and L. Kazemi, "A server-assigned spatial crowdsourcing framework," *ACM Transactions on Spatial Algorithms and Systems*, vol. 1, no. 1, pp. 1–28, 2015.
- [19] L. T. Thanh, T. D. Huynh, A. Rosenfeld, S. Ramchun, and N. R. Jennings, "BudgetFlx: budget limited crowdsourcing for interdependent task allocation with quality guarantees," 13th international conference on autonomous agents and multi-agent systems, *AAMAS 2014*, vol. 1, pp. 477–484, 2014.
- [20] U. Hassan and E. Curry, "A capability requirements approach for predicting worker performance in crowdsourcing," vol. 1, pp. 429–437, in *Proceedings of the 2013 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, vol. 1, IEEE Computer Society, Austin, TX, USA, October 2013.
- [21] H. To, G. Gabriel, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *Proceedings of the VLDB Endow*, vol. 7, no. 10, pp. 919–930, 2014.
- [22] U. U. Hassan, S. O'Riain, and E. Curry, *E_ects of Expertise Assessment on the Quality of Task Routing in Human Computation*, <https://doi.org/10.14236/ewic/sohuman2013.2>, 2013.
- [23] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *VLDB JOURNAL*, vol. 1, pp. 217–250, 2020.
- [24] U. U. Hassan and E. Curry, "A multi-armed bandit approach to online spatial task assignment," in *Proceedings of the 2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and its Associated Workshops*, pp. 212–219, IEEE, Bali, Indonesia, December 2014.
- [25] Y. Zeng, Y. Tong, L. Chen, and Z. Zhou, "Latency-Oriented task completion via spatial crowdsourcing," in *Proceedings of the 2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 317–328, {IEEE} Computer Society, Paris, France, April 2018.
- [26] L. Tran, H. To, L. Fan, and C. Shahabi, "A real-time framework for task assignment in h spatial crowdsourcing," *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 3, pp. 1–26, 2018.
- [27] Z. Shen, Y. Han, C. Miao, and J. Weng, "Trust-based web service selection in virtual communities," *Web Intelligence and Agent Systems*, vol. 9, pp. 227–238, 2011.
- [28] Y. Han, S. Liu, A. Kot, C. Miao, and C. Leung, "Dynamic witness selection for trustworthy distributed cooperative sensing in cognitive radio networks," in *Proceedings of the International Conference on Communication Technology Proceedings, ICCT*, pp. 1–6, Baku, Azerbaijan, October 2011.
- [29] P. Cheng, X. Lian, Z. Chen et al., "Reliable diversity based spatial crowdsourcing by moving workers," *Proceedings of the VLDB Endow*, vol. 8, no. 10, pp. 1022–1033, 2015.
- [30] Y. Zhao, J. Xia, G. Liu et al., "Preference-aware task assignment in spatial crowdsourcing," in *Proceedings of the The Thirty-Third AAAI Conference on Arti_cial Intelligence (AAAI-19)*, Honolulu, HI, USA., February 2019.
- [31] C. Veness, *Calculate Distance and Bearing between Two Latitude/Longitude Points Using Haversine Formula in JavaScript*, <http://www.movable-type.co.uk/scripts/latlong.html>, 2002-2020.
- [32] M. A. Mohammed, H. A. Karrar, S. Alaa et al., "Benchmarking methodology for selection of optimal covid-19 diagnostic model based on entropy and topsis methods," *IEEE Access*, vol. 8, 2020.
- [33] China Society for Industrial and Applied Mathematics, "Dataset: the data of allocated sharing tasks," 2017, http://www.mcm.edu.cn/html_cn/node/460baf68ab0ed0e1e557a0c79b1c4648.html.

Research Article

Privacy-Preserving Incentive Mechanism for Mobile Crowdsensing

Tao Wan ¹, Shixin Yue ¹ and Weichuan Liao ²

¹School of Information Engineer, East China Jiaotong University, Nanchang 330013, China

²School of Science, East China Jiaotong University, Nanchang 330013, China

Correspondence should be addressed to Tao Wan; wantao217@163.com

Received 1 May 2021; Revised 24 June 2021; Accepted 25 July 2021; Published 15 August 2021

Academic Editor: Jinbo Xiong

Copyright © 2021 Tao Wan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Incentive mechanisms are crucial for motivating adequate users to provide reliable data in mobile crowdsensing (MCS) systems. However, the privacy leakage of most existing incentive mechanisms leads to users unwilling to participate in sensing tasks. In this paper, we propose a privacy-preserving incentive mechanism based on truth discovery. Specifically, we use the secure truth discovery scheme to calculate ground truth and the weight of users' data while protecting their privacy. Besides, to ensure the accuracy of the MCS results, a data eligibility assessment protocol is proposed to remove the sensing data of unreliable users before performing the truth discovery scheme. Finally, we distribute rewards to users based on their data quality. The analysis shows that our model can protect users' privacy and prevent the malicious behavior of users and task publishers. In addition, the experimental results demonstrate that our model has high performance, reasonable reward distribution, and robustness to users dropping out.

1. Introduction

As more and more sensors are integrated into human-carried mobile devices, such as GPS locators, gyroscopes, environmental sensors, and accelerometers, they can collect various types of data [1]. Therefore, the MCS system [2–4] can utilize the sensors equipped in mobile devices to collect sensing data and complete various sensing tasks [5], such as navigation service [6], traffic monitoring [7], indoor positioning [8], and environmental monitoring [9]. In general, the MCS system consists of three entities: a task requester, a sensing server, and participating users, as shown in Figure 1. The task requester publishes sensing tasks and pays awards for sensing results. The server recruits users according to the sensing task, processes the data from users, and sends the results to the task publisher. Users collect sensing data based on the requirements of the sensing task and get rewards.

In the practical MCS system, the sensing data collected by users are not always reliable [10, 11] due to various factors (such as poor sensor quality, lack of effort, and background noise). Therefore, the final result may be inaccurate if we treat the data provided by each user equally (e.g., averaging).

To solve this problem, truth discovery [12–14] has been widely concerned by industry and academia. The main idea of most truth discovery schemes is that the user will be given a higher weight (i.e., reliability) if the user's data are closer to the ground truth. Also, the data provided by a user will be counted more in the aggregation procedure if this user has a higher weight. Recently, a number of truth discovery methods [15] have been proposed to calculate user's weight and aggregated results based on this basic idea. But one problem with these methods is that users have to be online to interact with the server. Otherwise, the MCS system may fail and have to restart. Therefore, if we design a truth discovery scheme that allows users to exit, the MCS system can get stronger robustness.

The proper functioning of the truth discovery requires enough users and high-quality sensing data. Generally, the MCS system utilizes an incentive mechanism [16–18] to motivate sufficient users to participate in sensing tasks. However, because of monetary incentives, malicious users attempt to earn rewards with little or no effort. Although the truth discovery can assign low weight to malicious users, their continuous input of erroneous data can result in the

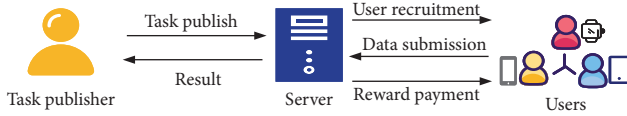


FIGURE 1: A general MCS system.

unavailability of the MCS system [19]. Consequently, the evaluation of data quality is critical to the MCS system. To improve data quality, users who provide incorrect data can be removed before sensing data get aggregated [20]. On the one hand, we can get more accurate aggregation results. On the other hand, users who provide eligible data can get more monetary rewards.

Although the incentive mechanism has been improved a lot, users' privacy protection remains inadequate. When users submit sensing data, their sensitive or private information [21–23] may be leaked, including identity privacy [24], location privacy, and data privacy. Also, privacy disclosure [25] will reduce users' willingness to participate in sensing tasks. Some incentive mechanism methods only consider the cost of users to collect sensing data but do not consider the potential cost of privacy disclosure. Recently, some researchers have designed privacy-preserving incentive mechanisms [26–28]. In [20], an incentive method is proposed to protect the user's identity and data privacy. Still, the user's sensing data will be submitted to the task publisher regardless of the privacy of the sensing data. In [29], the incentive mechanism is designed under the assumption of a trusted platform, which may not hold in practice since the platform itself might be attacked by hackers.

To address these issues, we propose a privacy-preserving incentive mechanism based on truth discovery, called PAID. In our PAID, the task publisher sets data constraints, such as time, location [30], budget [31], and sensing data. If the user does not collect the sensing data at the required time and location or sensing data are not in the qualified range, we believe that the user's sensing data are not credible (i.e., unqualified). After removing the unqualified user's data, the qualified user's sensing data will be submitted to the server to calculate the ground truth and weight. We also design a secure truth discovery scheme, which uses secret sharing technology and key agreement protocol and can still work when some users drop out. Moreover, our truth discovery can ensure that other parties cannot obtain users' sensing data except users themselves. Finally, we calculate every user's data quality according to the weight and distribute the reward.

In summary, the main contributions of this paper are as follows:

- (i) We introduce a privacy-preserving interval judgment scheme to remove users who provide unreliable data before performing the truth discovery scheme. Removing unqualified users in advance can greatly improve the quality of the sensing data used in the truth discovery scheme, improve the accuracy of results, and save the reward budget.

- (ii) We introduce a secure truth discovery scheme so that our incentive mechanism model can obtain the ground truth and the weight of each user's data while protecting the user's privacy. Then, we design a reasonable reward distribution scheme based on the data weight of users. Moreover, our incentive mechanism model can allow users to drop out at any time.

- (iii) Analysis shows that our model is secure. Also, experimental results demonstrate that our model has high performance and can achieve reasonable reward distribution.

The remainder of this paper is organized as follows. In Section 2, we describe the problem statement. In Sections 3 and 4, we introduce cryptography primitives and intuitive technology in our model. Then, we discuss PAID in detail in Section 5. Next, Sections 6 and 7 carry out the analysis and performance evaluation. Finally, we discuss the related work and conclude the paper in Sections 8 and 9.

2. Problem Statement

In this section, we introduce the background of truth discovery and our system model. Then, we describe the threat model and our design goals. Table 1 summarizes the main notations in this paper.

2.1. Truth Discovery. Truth discovery [32] is widely used in the MCS system to solve the conflicts between sensing data collected from multiple sources. Although the methods of estimating weights and calculating ground truth are different, their general processes are similar. Specifically, truth discovery initializes a random ground truth and then iteratively updates the weight and ground truth until convergence.

2.1.1. Weight Update. Suppose that the ground truth of the object is fixed. If the user's sensing data are close to the ground truth, a higher weight should be assigned to the user. The weight w_i of each user u_i can be iteratively updated as follows:

$$w_i = \log \left(\frac{\sum_{i'=1}^{|U|} d_{ist}(x_{i'}, x^*)}{d_{ist}(x_i, x^*)} \right), \quad (1)$$

where $d_{ist}(x_i, x^*)$ is a distance function and $d_{ist}(x_i, x^*) = (x_i - x^*)^2$. We use U to represent the set of users, and $|U|$ is the number of users in the set U . The sensing data collected by the user u_i are denoted as x_i , in which i is the number of u_i , and x^* is the estimated ground truth.

2.1.2. Truth Update. Similarly, we assume that the weight w_i of each user u_i is fixed. Then, we can calculate the ground truth x^* as follows:

TABLE 1: Summary of notations.

Notations	Description
\mathcal{T}	A sensing task
TP	A task publisher
\mathcal{S}	A server
u_i	A user who performs a sensing task
U	A set of users
$ U $	Number of users in the set U
w_i	The weight of the user u_i
x^*	Ground truth of the sensing object
x_i	Sensing data collected by the user u_i
τ_i	Time for u_i to collect sensing data
\tilde{l}_i	The longitude for u_i to collect sensing data
l_i	The latitude for u_i to collect sensing data
\mathcal{D}_i	The data submitted by the user u_i which is denoted by $(x_i, \tau_i, \tilde{l}_i, \tilde{l}_i)$
\mathcal{E}	Eligibility rank for \mathcal{D}_i is denoted by $(\mathcal{E}_x, \mathcal{E}_\tau, \mathcal{E}_{\tilde{l}}, \mathcal{E}_{\tilde{l}})$
B	Budget constraint of a sensing task
$(pk_{\mathcal{T}}, sk_{\mathcal{T}})$	Key pair of public-key encryption
$\text{Enc}(P, pk_{\mathcal{T}})$	(IND-CPA) Public-key encryption function $C = \text{Enc}(P, pk_{\mathcal{T}})$, where P is a plaintext
$\text{Dec}(C, sk_{\mathcal{T}})$	Public-key decryption function $P = \text{Dec}(C, sk_{\mathcal{T}})$, where C is a ciphertext
$\text{SEnc}(P, k_i)$	Symmetric encryption function $C = \text{SEnc}(P, k_i)$, where k_i is the key
$\text{SDec}(C, k_i)$	Symmetric encryption function $P = \text{SDec}(C, k_i)$
π	A reward control parameter
q_i	The data quality of the user u_i , and $q_i = (w_i / \sum_{u_i \in U} w_i)$
\bar{q}	Mean of data quality, and $\bar{q} = (\sum_{u_i \in U} q_i / U) = (1 / U)$
p_i	Monetary reward of the user u_i , and $p_i = (B / U) + \pi \cdot (q_i - \bar{q}) \geq 0$
c_i	The cost of the user u_i performing a sensing task
ut_i	Utility of the user u_i

$$x^* = \frac{\sum_{i=1}^{|U|} w_i \cdot x_i}{\sum_{i=1}^{|U|} w_i}. \quad (2)$$

The final ground truth x^* is obtained by iteratively running the weight update and the truth update until the convergence condition is satisfied.

2.2. System Model. Similar to the general MCS system, our PAID comprises three entities: a task publisher (TP), a server (\mathcal{S}), and users. In our PAID, the TP publishes tasks and requirements to \mathcal{S} and gets the ground truth of the object from \mathcal{S} . The server \mathcal{S} recruits adequate users and removes the users who provide unqualified data. After receiving the sensing data of all users, \mathcal{S} performs the truth discovery scheme and gets the ground truth and the weight of each user. To prevent the TP from refusing to pay the reward, we require the TP to prepay the reward to \mathcal{S} as a guarantee. After getting the weight of each user, the server \mathcal{S} calculates the data quality and distributes the rewards. Users collect sensing data and earn monetary rewards by providing qualified data. Moreover, our PAID can protect users' privacy of time, location, identity, and sensing data. Unlike general MCS models, in our PAID, the TP and \mathcal{S} can only get the aggregated result instead of users' sensing data. Figure 2 shows the flow of our PAID. The specific process of our model is as follows.

- (1) *Task Publish.* The TP publishes a sensing task to \mathcal{S} , including sensing objects, data eligibility requirements, and budget.

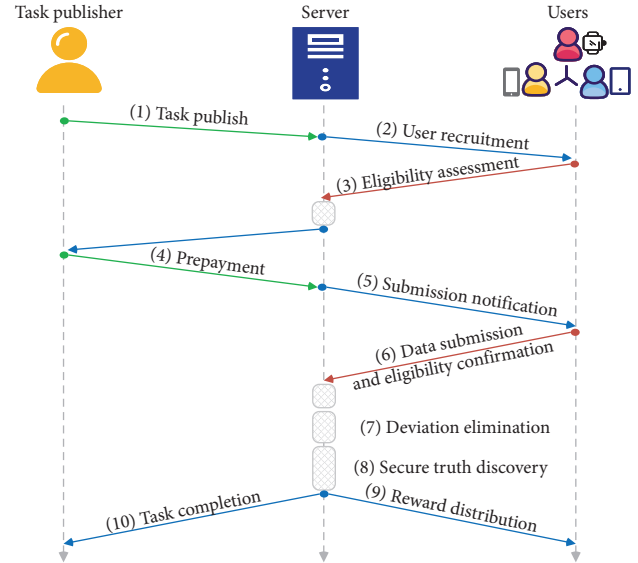


FIGURE 2: System model of PAID.

- (2) *User Recruitment.* The server \mathcal{S} broadcasts the sensing task and recruits participating users.
- (3) *Eligibility Assessment.* The server \mathcal{S} judges whether every user's sensing data meet qualification requirements.
- (4) *Prepayment.* The TP prepays \mathcal{S} monetary reward to avoid the denial of payment attack.
- (5) *Submission Notification.* The server \mathcal{S} notifies qualified users to submit sensing data.

- (6) *Data Submission and Eligibility Confirmation.* Users submit the masked sensing data to \mathcal{S} . And the server \mathcal{S} needs to confirm whether the submitted sensing data are qualified to prevent malicious users from tampering with the data.
- (7) *Deviation Elimination.* The server \mathcal{S} removes users who tamper with their sensing data and eliminates the deviation of data aggregation caused by these dropped users.
- (8) *Secure Truth Discovery.* The server \mathcal{S} calculates the ground truth and weight of each user by performing the security truth discovery scheme.
- (9) *Reward Distribution.* The server \mathcal{S} calculates the data quality of each user and distributes the rewards.
- (10) *Task Completion.* The server \mathcal{S} sends the ground truth of the sensing object to TP.

2.3. *Threat Model.* In this section, we mainly consider the potential threats from TP, the server \mathcal{S} , and users.

We suppose that TP is dishonest. After getting data from \mathcal{S} , TP may launch a *denial of payment attack* (DoP) and refuse to pay rewards.

The server \mathcal{S} is considered as honest-but-curious [33, 34]. Specifically, the server \mathcal{S} follows the agreement execution instructions, but it also attempts to spy on users' private data. In other words, the server \mathcal{S} may launch *inference attacks* (IAs) on the users' private data.

We assume that users are untrusted. Some malicious users may provide erroneous data and launch a *data pollution attack* (DPA). Besides, untrusted users may forge multiple identities and initiate a *Sybil attack* (SA), to earn more monetary rewards.

2.4. *Design Goals.* In this section, we introduce the design goals of our PAID, which are divided into privacy and security goals and property goals.

The privacy goals can protect the user's private data, and the security goals can avoid malicious attacks. The details are as follows.

- (i) *Privacy Goals.* PAID can protect user's location privacy, data privacy, and identity privacy. Specifically, the location and sensing data of a user cannot be obtained by any other parties except the user himself. And users' real identities would not be disclosed when performing a sensing task.
- (ii) *Security Goals.* In our PAID, users can avoid the *denial of payment attack* (DoP) of TP. The server \mathcal{S} cannot initiate an *inference attack* (IA) on users. The server \mathcal{S} can resist the *data pollution attack* (DPA) launched by malicious users. And our PAID guarantees fairness by resisting the *Sybil attack* (SA).

Our PAID also requires the following property goals.

- (i) *Eligibility.* If users' data do not meet the eligibility requirements, they cannot pass the eligibility

assessment. In other words, the sensing data adopted by our PAID must be eligible.

- (ii) *Zero Knowledge.* When the server \mathcal{S} assesses whether users' data meet the eligibility requirements, it cannot obtain the content of users' private data.
- (iii) *Payment Rationality.* Each user can get non-negative utility as long as the user provides qualified data.
- (iv) *Budget Rationality.* The total monetary reward paid by the TP does not exceed the budget constraint.

3. Preliminaries

In this section, we review the cryptographic primitives used in our PAID.

3.1. *Secret Sharing.* We use Shamir's t -out-of- N secret sharing protocol [35], which can split each user's secret s into N shares, where any t shares can be used to reconstruct s . Still, it is impossible to get any information about s if the shares obtained by attackers are less than t .

We assume that some integers can be identified with distinct elements in a finite field \mathcal{F} , where \mathcal{F} is parameterized with a size of $l > 2^k$ (in which k is the security parameter). These integers can represent all users' IDs, and we use a symbol U to denote the set of users' IDs. Then, Shamir's secret sharing protocol consists of two steps as below.

- (i) $\text{Shamir.share}(s, t, U) \rightarrow \{(u_i, s_i)\}_{u_i \in U}$: the inputs of the sharing algorithm are a secret s , a threshold $t \leq |U|$, and a set U of N field elements denoting the users' ID, where $|U| = N$. It outputs a set of shares s_i , each of which is associated with its corresponding the user u_i .
- (ii) $\text{Shamir.recon}(\{(u_i, s_i)\}_{u_i \in \mathcal{M}}, t) \rightarrow s$: the inputs of the reconstruction algorithm are the shares corresponding to a subset $\mathcal{M} \subseteq U$ and a threshold t , where $t \leq |\mathcal{M}|$, and it outputs the secret s .

Correctness requires that $\forall s \in \mathcal{F}, \forall t, N$ with $1 \leq t \leq N$. If $(u_i, s_i)_{u_i \in U} \leftarrow \text{Shamir.share}(s, t, U)$, where $\mathcal{M} \subseteq U$ and $t \leq |\mathcal{M}|$, then $\text{Shamir.recon}(\{(u_i, s_i)\}_{u_i \in \mathcal{M}}, t) \rightarrow s$.

Security requires $\forall s, s' \in \mathcal{F}$ and any $\mathcal{M} \subseteq U$ with $t > |\mathcal{M}|$. We have

$$\begin{aligned} & \{(u_i, s_i)\}_{u_i \in U} \leftarrow \text{Shamir.share}(s, t, U): \{(u_i, s_i)\}_{u_i \in \mathcal{M}} \\ \equiv & \{(u_i, s_i)\}_{u_i \in U} \leftarrow \text{Shamir.share}(s', t, U): \{(u_i, s_i)\}_{u_i \in \mathcal{M}}, \end{aligned} \quad (3)$$

where " \equiv " indicates that the two distributions are indistinguishable.

3.2. *Key Agreement.* We utilize the Diffie-Hellman key agreement called SIGMA [36] in our PAID to generate a session key between two users. Typically, SIGMA is described in three parts as follows.

- (i) $\text{KA.param}(k) \rightarrow (\mathbb{G}, g, q, H)$: the algorithm's input is a security parameter k . It samples a group \mathbb{G} of prime order q , along with a generator g and a hash function H , where H is set as SHA-256 for practicability in our model.
- (ii) $\text{KA.gen}(\mathbb{G}, g, q, H) \rightarrow (x, g^x)$: the algorithm's inputs are a group \mathbb{G} of prime order q , along with a generator g and a hash function H . It samples a random $x \leftarrow Z_q$ and g^x , where x and g^x will be marked as the secret key SK_i and the public key PK_i in the following sections.
- (iii) $\text{KA.agree}(\text{sign}_j(g^{x_i}, g^{x_j}), \text{MAC}_{k_j}(u_j), x_i, g^{x_i}, i, j) \rightarrow s_{i,j}$: the algorithm's inputs are the user u_i 's secret key x_i , the user u_j 's public key g^{x_j} , signed signature $\text{sign}_j(g^{x_i}, g^{x_j})$, and $\text{MAC}_{k_j}(u_j)$ from the user u_j , where k_j is used as the MAC key. It outputs a session key $s_{i,j}$ between user u_i and user u_j . For simplicity, we use $\text{KA.agree}(x_i, g^{x_j}) \rightarrow s_{i,j}$ to represent the above process in the following sections.

Correctness requires that $\text{KA.agree}(\text{SK}_i, \text{PK}_j) = \text{KA.agree}(\text{SK}_j, \text{PK}_i)$ for any private and public key generated by the users u_i and u_j if two users use the same parameters. Security requires that the shared key $s_{i,j}$ is indistinguishable from a uniformly random string for any adversary who is given public keys PK_i and PK_j (but do not have the corresponding secret keys SK_i and SK_j).

3.3. Paillier Cryptosystem. The Paillier cryptosystem [37] is a probabilistic public key cryptosystem. It consists of three parts as follows.

- (i) $\text{Paillier.gen}(N, g) \rightarrow (sk_p, pk_p)$: the key distribution algorithm inputs are a number N and $g \leftarrow Z_{N^2}^*$, where N is the product of two large primes p, q . It outputs a secret key sk_p and a public key pk_p , where pk_p is computed by (N, g) , and $sk_p = \text{lcm}(p-1, q-1)$.
- (ii) $\text{Paillier.enc}(m, pk_p) \rightarrow c$: the encryption algorithm inputs are a plaintext m (which $m < N$) and a public key pk_p . It outputs a ciphertext c .
- (iii) $\text{Paillier.dec}(c, sk_p) \rightarrow m$: the decryption algorithm inputs are a ciphertext c (which $c < N^2$) and a secret key sk_p . It outputs a plaintext m .

The Paillier cryptosystem has the property of homomorphic addition.

$$E_{pk}(a + b) = E_{pk}(a) \cdot E_{pk}(b) \pmod{N^2}. \quad (4)$$

We assume that E is an encryption function.

4. Technical Intuition

In this section, we first introduce how the interval judgment scheme can judge users' data eligibility while protecting users' privacy. Then, we notice that truth discovery mainly

involves the aggregation of multiple users' data in a secure manner. Therefore, we require that the server \mathcal{S} only get the sum of users' input, not content. And we propose a double-masking scheme to achieve this goal.

4.1. Interval Judgment Scheme for Privacy Protection. In our PAID, we use the interval judgment scheme [38] based on the Paillier cryptosystem to determine the sensing data eligibility. Every user u_i provides sensing data x_i , and the server \mathcal{S} provides a continuous integer interval $[y_1, y_2]$ ($y_1, y_2 \leftarrow Z^*$). The server \mathcal{S} can judge whether the user u_i 's sensing data x_i meet the interval range $[y_1, y_2]$ without knowing the data x_i . The user u_i also cannot obtain any information about the integer interval. The scheme is divided into four steps as follows.

- (i) The user u_i gets $(pk_p, sk_p) \leftarrow \text{Paillier.gen}(N, g)$ and then u_i computes $E(x_i)$ using pk_p and sends it to \mathcal{S} .
- (ii) The server \mathcal{S} picks two random numbers k, b ($k, b \leftarrow Z^*$) to construct a monotone increasing (or decreasing) function $f(x_i) = kx_i + b$. Then, the server \mathcal{S} computes $f(y_1), f(y_2), c = E(x_i)^k E(b) = E(kx + b)$ and sends them to u_i .
- (iii) After receiving the information from the server \mathcal{S} , the user u_i gets $f(x_i) \leftarrow \text{Paillier.dec}(c, sk)$ and then compares the size of $f(y_1), f(y_2)$, and $f(x_i)$. Next, the message is sent to the server \mathcal{S} .
- (iv) After receiving the message from u_i , the server \mathcal{S} judges whether $f(y_1) < f(x_i) < f(y_2)$. If so, we can know $x_i \in [y_1, y_2]$ because of the monotonicity of the function $f(x_i) = kx_i + b$, i.e., the user u_i passes the data eligibility assessment. Otherwise, the user u_i fails to pass the eligibility assessment of the server \mathcal{S} .

It should be noted that since the user u_i does not know the monotonicity of the function $f(x) = kx + b$, it is impossible to infer whether the data x_i are in the range of the interval $[y_1, y_2]$ from the size relationship. For simplicity, we formulate the above process as an interval judgment function denoted by $\text{ins}(x_i, y_1, y_2)$. If the user u_i passes the eligibility assessment of the server \mathcal{S} , $\text{ins}(x_i, y_1, y_2) = 1$; otherwise, $\text{ins}(x_i, y_1, y_2) = 0$.

4.2. One-Masking Scheme. Assume that all users are represented in sequence as integers $1, n$. And any pair of users $(u_i, u_j), i < j$, agrees on a random value $r_{i,j}$. Let us add $r_{i,j}$ to the user u_i 's data x_i and subtract $r_{i,j}$ from the user u_j 's data x_j to mask all users' raw data. In other words, each user u_i computes as follows.

$$y_i = x_i + \sum_{u_j \in \mathcal{U}: i < j} r_{i,j} - \sum_{u_j \in \mathcal{U}: i > j} r_{j,i} \pmod{R}, \quad (5)$$

where we assume x_i and $\sum_{u_j \in \mathcal{U}} r_{i,j}$ are in Z_R with order R for simplicity.

Then, each user u_i submits y_i to the server \mathcal{S} , and \mathcal{S} computes

$$\begin{aligned}
z &= \sum_{u_i \in U} y_i \\
&= \sum_{u_i \in U} \left(x_i + \sum_{u_j \in U: i < j} r_{i,j} - \sum_{u_j \in U: i > j} r_{j,i} \right) \quad (6) \\
&= \sum_{u_i \in U} x_i \pmod{R}.
\end{aligned}$$

However, this approach has two shortcomings. The first one is that every user u_i needs to exchange the value $r_{i,j}$ with all other users, which will result in quadratic communication overhead ($|U|^2$) if done naively. The second one is that the protocol will fail if any user u_i drops out since the server cannot eliminate the value $r_{i,j}$ associated with u_i in the final aggregated results z .

4.3. Double-Masking Scheme. To solve these security problems, we introduce a double-masking scheme [39, 40]. In the work [40], the double-masking scheme is used for privacy-preserving data aggregation. And the scheme in [40] can also protect location privacy and verify the aggregation results. In our model, location privacy protection is implemented by the interval judgment scheme, and our secure truth discovery will confirm the data consistency. The details of the double-masking scheme are as follows.

$$y_i = x_i + \text{PRG}(n_i) + \sum_{u_j \in U: i < j} \text{PRG}(r_{i,j}) - \sum_{u_j \in U: i > j} \text{PRG}(r_{i,j}) \pmod{R}. \quad (7)$$

Note that an honest user will never reveal both kinds of shares of the same user to the server \mathcal{S} . During the recovery round, the server \mathcal{S} can request either a share of $r_{i,j}$ or a share of n_i from each surviving user u_j . After gathering at least t shares of $r_{i,j}$ for all dropped users and t shares of n_i for all surviving users, the server \mathcal{S} can eliminate the remaining masks to reveal the sum.

5. Our Proposed Scheme

In this section, we first provide an overview of our PAID. Then, we show the details of the three critical designs in our PAID, including eligibility assessment, truth discovery, and reward distribution. In the eligibility assessment stage, the server \mathcal{S} judges whether users' sensing data meet the requirements of a sensing task. In the truth discovery stage, the server \mathcal{S} can calculate each user's weight and the ground truth required by the sensing task without knowing their sensing data. In the reward distribution stage, the server \mathcal{S} computes the quality of sensing data by each user's weight and then pays a reward to users.

5.1. Overview. For convenience, we introduce a simple case. We set up a sensing task \mathcal{T} to collect the temperature of urban roads in the evening. There are range requirements for time, location, and sensing data (i.e., temperature). To be more precise, the time range is required to be 5–8 pm on

February 3rd, the location range is required to be 12.45–12.55 E and 41.79–41.99 N, and the temperature requirement is 10–15°C. In our PAID, we consider the range requirement as the data eligibility requirement \mathcal{E} . The data \mathcal{D}_i ($\mathcal{D}_i = (x_i, \tau_i, \hat{l}_i, \bar{l}_i)$) collected by a user u_i meet the eligibility requirements \mathcal{E} , meaning that $10 \leq x_i \leq 15$, $5 \leq \tau_i \leq 8$, $12.45 \leq \hat{l}_i \leq 12.55$, $41.79 \leq \bar{l}_i \leq 41.99$. Since the data collected by mobile devices are usually rational numbers, in our PAID, we transform the eligible interval into an integer interval by moving the decimal point right. The sensing task \mathcal{T} consists of three entities: a task publisher (TP), a server (\mathcal{S}), and users. And the specific steps are as follows.

Every user u_i can get a session key $r_{i,j}$ with other user u_j by engaging the Diffie–Hellman key agreement after the server \mathcal{S} broadcasts all of the Diffie–Hellman public keys. Then, we can utilize a pseudorandom generator (PRG) to reduce the high communication overhead by having the parties agree on a common seed instead of the whole mask $r_{i,j}$.

We use the threshold secret sharing scheme to solve the issue that users are not allowed to drop out. Every user u_i can send his secret shares to other users. Once some users cannot submit data in time, other users can recover masks associated with these users by submitting shares of these users' secrets to \mathcal{S} , as long as the number of dropped users is less than t (i.e., threshold of Shamir's secret sharing).

However, there is a problem that may lead to users' data leaked to \mathcal{S} . There is a scenario where a user u_i is very slow to send data to \mathcal{S} . The server \mathcal{S} considers that the user u_i has dropped and asks for their shares of the user u_i 's secret from all other users. Then, the server receives the delayed data y_i after recovering u_i 's mask. At this time, the server \mathcal{S} can remove all the masks $r_{i,j}$ and get the plaintext x_i .

To improve the scheme, we introduce an additional random seed n_i to mask the data. Specifically, each user u_i selects a random seed n_i on the round of generating $r_{i,j}$ and then creates and distributes shares of n_i to all other users during the secret sharing round. Now, users calculate y_i as follows:

February 3rd, the location range is required to be 12.45–12.55 E and 41.79–41.99 N, and the temperature requirement is 10–15°C. In our PAID, we consider the range requirement as the data eligibility requirement \mathcal{E} . The data \mathcal{D}_i ($\mathcal{D}_i = (x_i, \tau_i, \hat{l}_i, \bar{l}_i)$) collected by a user u_i meet the eligibility requirements \mathcal{E} , meaning that $10 \leq x_i \leq 15$, $5 \leq \tau_i \leq 8$, $12.45 \leq \hat{l}_i \leq 12.55$, $41.79 \leq \bar{l}_i \leq 41.99$. Since the data collected by mobile devices are usually rational numbers, in our PAID, we transform the eligible interval into an integer interval by moving the decimal point right. The sensing task \mathcal{T} consists of three entities: a task publisher (TP), a server (\mathcal{S}), and users. And the specific steps are as follows.

Step 1 (Task Publish). The task publisher TP initializes a public key $pk_{\mathcal{T}}$ and a private key $sk_{\mathcal{T}}$, a reward control parameter π (π is a decimal number), a task budget B , the number of users N , and eligibility requirements \mathcal{E} for a sensing task \mathcal{T} . The public key $pk_{\mathcal{T}}$ is used to encrypt the information that the server \mathcal{S} needs to send to the TP, and the TP decrypts the ciphertext using the private key $sk_{\mathcal{T}}$. Then, the TP sends the information $\{\mathcal{T}, pk_{\mathcal{T}}, \pi, N, B, \mathcal{E}\}$ to \mathcal{S} as a task request.

Step 2 (User Recruitment). The server \mathcal{S} broadcasts the sensing task information $\{\mathcal{T}, \pi, N, B\}$ and recruits N users who request to participate in the sensing task. Then, \mathcal{S} generates a key pair $\{PK_{\mathcal{S}}^i, SK_{\mathcal{S}}^i\}$ using the key agreement scheme for every user u_i and sends $PK_{\mathcal{S}}^i$ to u_i .

Step 3 (Eligibility Assessment). Each user u_i confirms whether $c_i \leq (B - \pi/N)$, where c_i denotes the sensing cost of u_i , and the posted lowest reward is denoted as $(B - \pi/N)$. If $c_i \leq (B - \pi/N)$, and u_i starts the sensing task and collects the data \mathcal{D}_i . The user u_i then generates a key pair $\{PK_i, SK_i\}$ using the key agreement scheme and computes a session key $k_i \leftarrow \text{KA.agree}(SK_i, PK_{\mathcal{S}}^i)$ as u_i 's anonymous identity information. Then, the user u_i performs the interval judgment scheme $\text{ins}(\mathcal{D}_i, \mathcal{E})$ and sends the public key PK_i to \mathcal{S} . Specifically, $\text{ins}(\mathcal{D}_i, \mathcal{E})$ is divided into $\text{ins}(x_i, \mathcal{E})$, $\text{ins}(\tau_i, \mathcal{E})$, $\text{ins}(\hat{t}_i, \mathcal{E})$, $\text{ins}(\tilde{t}_i, \mathcal{E})$.

Step 4 (Prepayment). After recruiting N eligible users, the server \mathcal{S} requests TP to prepay a budget reward B for the sensing task \mathcal{T} to prevent the denial of payment attack. And the server \mathcal{S} calculates the session key $k_i \leftarrow \text{KA.agree}(SK_{\mathcal{S}}^i, PK_i)$ with the eligible user u_i .

Step 5 (Submission Notification). After getting the budget reward B , the server \mathcal{S} informs the eligible user u_i ($1 \leq i \leq N$) to submit data.

Step 6 (Data Submission and Eligibility Confirmation). After receiving the submission notification, each user u_i performs double-masking scheme to mask the sensing data x_i and get y_i and, at the same time, executes eligibility confirmation $\text{ins}(\mathcal{D}_i, \mathcal{E})$ to prevent malicious users from modifying data. Then, u_i encrypts the data y_i using the symmetric encryption algorithm and sends the ciphertext $\text{SEnc}(y_i, k_i)$ to \mathcal{S} . The session key k_i is the key of symmetric encryption.

Step 7 (Deviation Elimination). For users who tamper with data during data submission, the server \mathcal{S} regards them as dropped users and discards their data. Then, \mathcal{S} gets plaintext $\text{SDec}(\text{SEnc}(y_i, k_i), k_i)$ and requests seed n_i and the noise $r_{i,j}$ between the dropped user u_i and the surviving user u_j to eliminate the impact on the aggregate result.

Step 8 (Secure Truth Discovery). The server \mathcal{S} computes the surviving user u_i 's weight w_i and the ground truth x^* of the sensing object utilizing the truth discovery algorithm. The detailed algorithm process will be introduced later.

Step 9 (Reward Distribution). The server \mathcal{S} calculates the sensing data quality $q_i = (w_i / \sum_{i=1}^m w_i)$ of u_i , where $\sum_{i=1}^m q_i = 1$, m is the number of online users. Then, \mathcal{S} pays a monetary reward $p_i = (B/m) + \pi \cdot (q_i - \bar{q})$ for u_i , where $\pi \cdot (q_i - \bar{q})$ denotes the payment parameter, $m \leq N$, and $1 \leq i \leq m$.

Step 10 (Task Completion). The server \mathcal{S} encrypts the ground truth x^* using $pk_{\mathcal{S}}$ and sends $\text{Enc}(x^*, pk_{\mathcal{S}})$ to TP. And the TP can decrypt the data using $sk_{\mathcal{S}}$, i.e., $x^* = \text{Dec}(\text{Enc}(x^*, pk_{\mathcal{S}}), sk_{\mathcal{S}})$.

In our PAID, only users who passed the eligibility assessment and eligibility confirmation can obtain the monetary reward. Thus, users cannot cheat \mathcal{S} to get a reward with unreliable data. We can also ensure the quality of the sensing data used by the truth discovery algorithm and obtain more accurate ground truth x^* . Moreover, since the TP pays the

task reward to \mathcal{S} in advance and \mathcal{S} will pay a reward to u_i according to the quality of u_i 's sensing data after the task is accomplished, the TP cannot refuse to pay the reward. Besides, \mathcal{S} cannot get users' raw sensing data, time, and location information, which can protect the users' privacy. The anonymous identity of each user is determined by both the user and \mathcal{S} . \mathcal{S} only assigns one random identity token to each user, so malicious users cannot forge multiple identities.

5.2. Eligibility Assessment. In our PAID, there are three benefits to the design of the eligibility assessment. First, it can prevent users who provide unreliable or erroneous sensing data from receiving monetary rewards, which avoids wasting budgets. Secondly, filtering out unqualified sensing data can improve the accuracy of the sensing task result. Thirdly, the data quality q_i of each user u_i is related to the sensing object's ground truth x^* , and inaccurate ground truth will lead to unfair incentives.

The process of eligibility assessment and eligibility confirmation is similar. The purpose of the eligibility assessment is to filter out unqualified users preliminarily. Thus, the unqualified users do not need to communicate with other users to perform the double-masking scheme, by which the communication overhead can be reduced. The eligibility confirmation is designed to prevent malicious users from altering the original qualified data. The detailed process of eligibility assessment and eligibility confirmation is as follows.

Step 1. Each user u_i initializes a key pair $(pk_{p_i}, sk_{p_i}) \leftarrow \text{Paillier.gen}(N, g)$. Then, u_i encrypts the sensing data \mathcal{D}_i using pk_{p_i} and sends the ciphertext $E(\mathcal{D}_i)$ to \mathcal{S} . Generally, $E(\mathcal{D}_i)$ consists of four parts: $E(x_i)$, $E(\tau_i)$, $E(\hat{t}_i)$, and $E(\tilde{t}_i)$.

Step 2. After receiving $E(\mathcal{D}_i)$, the server \mathcal{S} picks different random k, b ($k, b \leftarrow \mathbb{Z}^*$) and constructs a monotone increasing (or decreasing) function $f(\mathcal{D}_i) = k\mathcal{D}_i + b$ for each value in the quadruples $(x_i, \tau_i, \hat{t}_i, \tilde{t}_i)$. The monotonicity of the four functions is inconsistent. For eligibility requirement interval \mathcal{E} ($\mathcal{E} = \{\mathcal{E}_x, \mathcal{E}_\tau, \mathcal{E}_{\hat{t}}, \mathcal{E}_{\tilde{t}}\}$, $\mathcal{E}_x = [x_l, x_r]$, $\mathcal{E}_\tau = [\tau_l, \tau_r]$, $\mathcal{E}_{\hat{t}} = [\hat{t}_l, \hat{t}_r]$, $\mathcal{E}_{\tilde{t}} = [\tilde{t}_l, \tilde{t}_r]$), the server \mathcal{S} calculates

$$\begin{cases} f(x_l), f(x_r), c_1 = E(x_l)^{k_1}, E(b_1) = E(k_1 x_l + b_1), \\ f(\tau_l), f(\tau_r), c_2 = E(\tau_l)^{k_2}, E(b_2) = E(k_2 \tau_l + b_2), \\ f(\hat{t}_l), f(\hat{t}_r), c_3 = E(\hat{t}_l)^{k_3}, E(b_3) = E(k_3 \hat{t}_l + b_3), \\ f(\tilde{t}_l), f(\tilde{t}_r), c_4 = E(\tilde{t}_l)^{k_4}, E(b_4) = E(k_4 \tilde{t}_l + b_4). \end{cases} \quad (8)$$

Then, \mathcal{S} sends $\{f(\mathcal{E}_l), f(\mathcal{E}_r), c\}$ ($c = \{c_1, c_2, c_3, c_4\}$) to u_i . For convenience, we will not describe \mathcal{D}_i and \mathcal{E} separately in the following text.

Step 3. After receiving $\{f(\mathcal{E}_l), f(\mathcal{E}_r), c\}$ from \mathcal{S} , each user u_i gets $f(\mathcal{D}_i) \leftarrow \text{Paillier.dec}(c, sk_{p_i})$ and then

compares the size of $f(\mathcal{E}_l), f(\mathcal{E}_r), f(\mathcal{D}_i)$. Next, the size relationship is sent to \mathcal{S} .

Step 4. After the server \mathcal{S} receives the information from u_i , if $f(\mathcal{E}_l) \leq f(\mathcal{D}_i) \leq f(\mathcal{E}_r)$, then $\mathcal{D}_i \in [\mathcal{E}_l, \mathcal{E}_r]$ because of the monotonicity of the functions. And \mathcal{S} determines whether u_i passes the eligibility assessment. Otherwise, it fails.

Because users do not know the function's monotonicity, they cannot infer the size relationship between the qualified data and eligibility requirement. Therefore, we can think that malicious users have a very low probability of passing the eligibility assessment. Moreover, during the eligibility assessment, u_i cannot know the specific qualified interval. \mathcal{S} also cannot get u_i 's sensing data, which can protect u_i 's privacy. The above process is represented by $\text{ins}(\mathcal{D}_i, \mathcal{E})$. If u_i passes the eligibility assessment, then $\text{ins}(\mathcal{D}_i, \mathcal{E}) = 1$. If not, $\text{ins}(\mathcal{D}_i, \mathcal{E}) = 0$.

5.3. Secure Truth Discovery. In the secure truth discovery scheme [15], data exchange is between users and the server \mathcal{S} . The user u_i needs to collect sensing data x_i , perform the double-masking scheme to mask the raw input data y_i ($y_i = d_{\text{ist}}(x_i - x^*)$), and then send the masked input data z_i to \mathcal{S} . The server \mathcal{S} receives masked input data z_i from each user u_i and aggregates the input data of online users. Each user u_i can drop out at any time. As long as the number of surviving users is not less than the threshold t , \mathcal{S} can eliminate the deviation caused by dropped users and restore the aggregation results. The detailed process is as follows.

Step 0 (Key Generation). Assume N users submit sensing data in the data submission phase. Given the security parameter k and threshold value t , a trusted third party creates three key pairs for each user u_i as follows.

$$\{(PK_i^s, SK_i^s), (PK_i^a, SK_i^a), (PK_i^r, SK_i^r)\} \leftarrow \text{KA.gen}(k), \quad (9)$$

where (PK_i^s, SK_i^s) are used for signature, (PK_i^a, SK_i^a) are used to generate a session key with other users for symmetric encryption, and (PK_i^r, SK_i^r) are used to generate a session key with other users u_j as the noise $r_{i,j}$. Then, each user u_i signs two public keys using SK_i^s

as $\rho_i \leftarrow \text{sign} \cdot (SK_i^s, PK_i^s \| PK_i^r)$ and sends $\{(\rho_i, \|PK_i^a\|PK_i^r)\}$ to \mathcal{S} .

When receiving messages from at least t users (which denotes the surviving users as a set $U_1 \subseteq U$), \mathcal{S} broadcasts $\{(u_j, \rho_j, PK_j^a, PK_j^r)\}_{u_j \in U_1}$ to all users. Otherwise, abort.

Step 1 (Key Sharing). After receiving the information from \mathcal{S} , each user u_i confirms whether $|U_1| \geq t$; then, u_i verifies whether the signature ρ_j is valid using the public key PK_j^s for other user u_j . If not, abort. Next, u_i selects a random parameter $n_i \leftarrow \mathcal{F}$ and generates shares of n_i and SK_i^r as follows.

$$\begin{aligned} \{(u_j, n_{j,i})\}_{u_j \in U_1} &\leftarrow \text{Shamir.share}(n_i, t, U_1), \\ \{(u_j, SK_{j,i}^r)\}_{u_j \in U_1} &\leftarrow \text{Shamir.share}(SK_i^r, t, U_1). \end{aligned} \quad (10)$$

Then, each user u_i generates a session key with other users $u_j \in U_1 \setminus \{u_i\}$ and uses the symmetric authenticated encryption to encrypt two types of shares as follows.

$$\mathcal{T}_{j,i} \leftarrow \text{AE.enc}(\text{KA.agree}(SK_i^a, PK_j^a), u_i \| u_j \| n_{j,i} \| SK_{j,i}^r), \quad (11)$$

where the symmetric authenticated encryption is indistinguishable under ciphertext integrity attack and chosen plaintext attack. It can ensure the confidentiality and integrity of messages, which are exchanged between two parties. We do not repeat the details here. If any of the above processes fails, abort. Otherwise, each user u_i sends $\mathcal{T}_{j,i}$ to \mathcal{S} .

When receiving messages from at least t users (which denotes the surviving users as a set U_2), \mathcal{S} randomly initializes the ground truth x^* and then broadcasts $\{\mathcal{T}_{j,i}\}_{u_j \in U_2}$ and x^* to all users. Otherwise, abort.

Step 2 (Masking Input Data). After receiving x^* and $\{\mathcal{T}_{j,i}\}_{u_j \in U_2}$ from \mathcal{S} , each user u_i confirms whether $|U_2| \geq t$, then computes $r_{i,j} \leftarrow \text{KA.agree}(SK_i^r, PK_j^r)$ for every user $u_j \in U_2 \setminus \{u_i\}$, and gets masked input data z_i^2 as follows.

$$\begin{aligned} z_i^2 &= y_i^2 + \sum_{j \in U_2: i < j} \text{PRG}(r_{i,j}) - \sum_{j \in U_2: i > j} \text{PRG}(r_{i,j}) + \text{PRG}(n_i) \pmod{R} \\ &= d_{\text{ist}}(x_i, x^*) + \sum_{j \in U_2: i < j} \text{PRG}(r_{i,j}) - \sum_{j \in U_2: i > j} \text{PRG}(r_{i,j}) + \text{PRG}(n_i) \pmod{R}, \end{aligned} \quad (12)$$

where $d_{\text{ist}}(x_i, x^*)$ is the input data in the second round, represented by y_i^2 for convenience, and z_i^2 indicates the masked input data. If any of the above processes fails, abort. Otherwise, each user u_i sends $\{z_i^2\}_{u_i \in U_2}$ to \mathcal{S} .

When receiving z_i^2 from at least t users (which denotes the surviving users as a set U_3), \mathcal{S} sends the list of U_3 to all users. Otherwise, abort.

Step 3 (Consistency Check). After receiving the list of U_3 from \mathcal{S} , each user u_i confirms whether $|U_3| \geq t$. Then, u_i calculates the signature $\rho_i^1 \leftarrow \text{sign} \cdot (SK_i^s, U_3)$ and sends it to \mathcal{S} .

When receiving ρ_i^1 from at least t users (which denotes the surviving users as a set U_4), \mathcal{S} sends $\{u_j, \rho_j^1\}_{u_j \in U_4}$ to all users. Otherwise, abort.

Step 4 (Unmasking). After receiving the list of U_4 from \mathcal{S} , each user u_i confirms whether $U_4 \subseteq U_3$, $|U_4| \geq t$, and the signature ρ_j^1 is valid using the public key PK_j^s . Then, u_i decrypts $\mathcal{F}_{j,i}$ for users $u_j \in U_2 \setminus \{u_i\}$ as follows.

$$u_i \| u_j \| n_{j,i} \| SK_{j,i}^r \leftarrow \text{AE.dec}(\text{KA.agree}(SK_i^a, PK_j^a), \mathcal{F}_{j,i}). \quad (13)$$

Then, $n_{j,i}$ ($u_j \in U_3$) and $SK_{j,i}^r$ ($u_j \in U_2 \setminus U_3$) will be sent to \mathcal{S} if $u_i = u'_i$ and $u_j = u'_j$. If any of the above processes fails, abort.

After receiving messages from users, \mathcal{S} performs the deviation elimination and regards users who modify the data as dropped users, and \mathcal{S} discards dropped users' data. The surviving users are then denoted as a set $U_5 \subseteq U_4$. If $|U_5| \geq t$, the secret key SK_i^r and masks $\text{PRG}(r_{i,j})$, $u_i \in U_2 \setminus U_3$ can be reconstructed as follows.

$$\begin{aligned} SK_i^r &\leftarrow \text{Shamir.recon}\left(\{(SK_{j,i}^r)\}_{u_j \in U_5}, t\right), \\ \text{PRG}(r_{i,j}) &\leftarrow \text{PRG}\left(\text{KA.agree}\left(\{SK_i^r, PK_j^r\}_{u_j \in U_5}\right)\right). \end{aligned} \quad (14)$$

Furthermore, the $\text{PRG}(n_i)$, $u_i \in U_3$ can be reconstructed as follows.

$$\text{PRG}(n_i) \leftarrow \text{PRG}\left(\text{Shamir.recon}\left(\{(n_{j,i}, t)\}_{u_j \in U_5}\right)\right). \quad (15)$$

Next, the aggregated results of y_i^2 can be calculated as follows.

$$\begin{aligned} \sum_{u_i \in U_3} y_i^2 &= \sum_{u_i \in U_3} y_i^2 - \sum_{u_j \in U_3} \text{PRG}(n_i) - \sum_{u_i \in U_3, u_j \in U_2 \setminus U_3: i < j} \text{PRG}(r_{i,j}) \\ &+ \sum_{u_i \in U_3, u_j \in U_2 \setminus U_3: i < j} \text{PRG}(r_{i,j}) \pmod R = \sum_{u_i \in U_3} d_{\text{ist}}(x_i, x^*). \end{aligned} \quad (16)$$

Then, \mathcal{S} selects a random positive noise value m to mask the raw aggregation results to prevent users from obtaining weight information.

$$\mathcal{W}_{\text{result}} = \text{Log}\left(\sum_{u_i \in U_3} d_{\text{ist}}(x_i, x^*)\right) + m. \quad (17)$$

Next, \mathcal{S} sends $\mathcal{W}_{\text{result}}^*$ to all users.

Step 5 (Masked Input Generation). After receiving $\mathcal{W}_{\text{result}}^*$ from \mathcal{S} , each user u_i computes $r_{i,j} \leftarrow \text{KA.agree}(SK_i^r, PK_j^r)$ for every surviving user $u_j \in U_5 \setminus \{u_i\}$. Then, each user u_i calculates the masked weight information as follows.

$$\begin{aligned} z_i^{5'} &= \mathcal{W}_{\text{result}}^* - \text{Log}(d_{\text{ist}}(x_i, x^*)) + \text{PRG}(n_i) + \sum_{u_j \in U_5: i > j} \text{PRG}(r_{i,j}) - \sum_{u_j \in U_5: i < j} \text{PRG}(r_{i,j}) \pmod R \\ &= w_i + m + \text{PRG}(n_i) + \sum_{u_j \in U_5: i > j} \text{PRG}(r_{i,j}) - \sum_{u_j \in U_5: i < j} \text{PRG}(r_{i,j}) \pmod R, \\ z_i^{5''} &= (\mathcal{W}_{\text{result}}^* - \text{Log}(d_{\text{ist}}(x_i, x^*))) \cdot x_i + \text{PRG}(n_i) + \sum_{u_j \in U_5: i > j} \text{PRG}(r_{i,j}) - \sum_{u_j \in U_5: i < j} \text{PRG}(r_{i,j}) \pmod R \\ &= (w_i + m) \cdot x_i + \text{PRG}(n_i) + \sum_{u_j \in U_5: i > j} \text{PRG}(r_{i,j}) - \sum_{u_j \in U_5: i < j} \text{PRG}(r_{i,j}) \pmod R. \end{aligned} \quad (18)$$

So, the masked input data are denoted as $(z_i^{\#}, z_i^{\#})$, where the raw input data are $y_i^{\#} = w_i + m$, $y_i^{\#} = (w_i + m) \cdot x_i$. If any of the above processes fails, abort. Otherwise, each user u_i sends $(z_i^{\#}, z_i^{\#})$ to \mathcal{S} .

Step 6 (Unmasking). After receiving $(z_i^{\#}, z_i^{\#})$ from at least t users (which denotes the surviving users as a set U_6), \mathcal{S} sends the list of U_6 to all users. Otherwise, abort. Then, each user $u_j \in U_6 \setminus \{u_i\}$ decrypts $\mathcal{F}_{j,i}$ as follows.

$$u_i \| u_j \| n_{j,i} \| SK_{j,i}^r \leftarrow \text{AE.dec}(\text{KA.agree}(SK_i^a, PK_j^a), \mathcal{T}_{j,i}). \quad (19)$$

Then, $n_{j,i}$ ($u_j \in U_6$) and $SK_{j,i}^r$ ($u_j \in U_5 \setminus U_6$) will be sent to \mathcal{S} if $u_i = u'_i$ and $u_j = u'_j$. If any of the above processes fails, abort.

After receiving the information from at least t users (which denotes the surviving users as a set U_7), \mathcal{S} restores the secret key SK_i^r , $u_j = u'_j$ for each user ($u_j \in U_5 \setminus U_6$) and $\{\text{PRG}(n_i)\}_{u_i \in U_6}$ as follows.

$$\begin{aligned} SK_i^r &\leftarrow \text{Shamir.recon}\left(\left\{\{SK_{j,i}^r\}_{u_j \in U_7}, t\right\}\right), \\ \text{PRG}(r_{i,j}) &\leftarrow \text{PRG}\left(\text{KA.agree}\left(\left\{\{SK_i^r, PK_j^r\}_{u_j \in U_6}\right\}\right)\right), \\ \text{PRG}(n_i) &\leftarrow \text{PRG}\left(\text{Shamir.recon}\left(\left\{\{n_{j,i}, t\}_{u_j \in U_7}\right\}\right)\right). \end{aligned} \quad (20)$$

Then, \mathcal{S} can calculate the aggregation results as follows.

$$\begin{aligned} \sum_{u_i \in U_6} y_i^{5'} &= \sum_{u_i \in U_6} z_i^{5'} - \sum_{u_i \in U_6} \text{PRG}(n_i) \\ &\quad - \sum_{u_i \in U_6, u_j \in U_5 \setminus U_6: i < j} \text{PRG}(r_{i,j}) \\ &\quad + \sum_{u_i \in U_6, u_j \in U_5 \setminus U_6: i > j} \text{PRG}(r_{j,i}) \pmod{R} \\ &= \sum_{u_i \in U_6} (w_i + m), \\ \sum_{u_i \in U_6} y_i^{5''} &= \sum_{u_i \in U_6} z_i^{5''} - \sum_{u_i \in U_6} \text{PRG}(n_i) \\ &\quad - \sum_{u_i \in U_6, u_j \in U_5 \setminus U_6: i < j} \text{PRG}(r_{i,j}) \\ &\quad + \sum_{u_i \in U_6, u_j \in U_5 \setminus U_6: i > j} \text{PRG}(r_{j,i}) \pmod{R} \\ &= \sum_{u_i \in U_6} (w_i + m) \cdot x_i. \end{aligned} \quad (21)$$

Next, \mathcal{S} eliminates the random noise value m as follows.

$$\begin{aligned} \mathcal{W}_{\text{result}'} &= \sum_{u_i \in U_6} y_i^{5'} \text{mod} \left(\sum_{u_i \in U_6} m \right) \\ &= \sum_{u_i \in U_6} w_i + \sum_{u_i \in U_6} m \text{mod} \left(\sum_{u_i \in U_6} m \right) \\ &= \sum_{u_i \in U_6} w_i, \\ \mathcal{W}_{\text{result}''} &= \sum_{u_i \in U_6} y_i^{5''} \text{mod} \left(\sum_{u_i \in U_6} m \right) \\ &= \sum_{u_i \in U_6} (w_i + m) \cdot x_i \text{mod} \left(\sum_{u_i \in U_6} m \right) \\ &= \sum_{u_i \in U_6} w_i \cdot x_i. \end{aligned} \quad (22)$$

Therefore, the current ground truth x^* and the weight w_i of every user $u_i \in U_6$ can be calculated using formulas (1) and (2) as follows.

$$x^* = \frac{\sum_{u_i \in U_6} w_i \cdot x_i}{\sum_{u_i \in U_6} w_i} = \frac{\mathcal{W}_{\text{result}''}}{\mathcal{W}_{\text{result}'}} \quad (23)$$

$$w_i = \text{Log} \left(\frac{\sum_{u_i \in U_6} d_{ist}(x_i, x^*)}{d_{ist}(x_i, x^*)} \right).$$

Thus, \mathcal{S} can get the final ground truth x^* and the weight w_i of every user u_i by repeating steps 0 to 6 until the convergence conditions are met. And the weight w_i will be used to calculate the data quality q_i of each user u_i .

5.4. Reward Distribution. The weight w_i calculated by truth discovery can represent the effective contribution of users. Still, to facilitate reward distribution, we need to quantify the data quality q_i of every user u_i further. Then, \mathcal{S} can compute the monetary reward p_i according to the data quality q_i of u_i .

To achieve the rationality of reward distribution, we set $\sum_{u_i \in U_6} q_i = 1$, so the data quality q_i of each user u_i can be calculated as follows.

$$q_i = \frac{w_i}{\sum_{u_i \in U_6} w_i}. \quad (24)$$

Next, we calculate the monetary reward p_i of each user u_i as follows. And the higher the quality q_i of u_i 's data, the more reward u_i can get.

$$p_i = \frac{B}{|U_6|} + \pi \cdot (q_i - \bar{q}), \quad (25)$$

where \bar{q} ($\bar{q} = (\sum_{u_i \in U_6} q_i / |U_6|) = (1/|U_6|)$) is the average quality of all surviving users. π ($0 < \pi < (B/N)$) represents the reward control parameter, which is a small rational number. The function of π is to ensure that the reward p_i is non-negative. And $|U_6|$ is the number of surviving users.

Since $|U_6| \leq N$, we can know that the lowest reward p_i which a user can get is $(B - \pi/N)$. When the number of final online users $|U_6| = N$, each user u_i 's reward is $p_i = (B/N) + \pi \cdot (q_i - \bar{q})$. If some users dropped out and $|U_6| < N$, \mathcal{S} will distribute the task budget to each surviving user $u_i \in U_6$, and each user 's reward is $p_i = (B/|U_6|) + \pi \cdot (q_i - \bar{q})$. Therefore, our reward distribution formula is applicable regardless of whether there are users offline.

6. Analysis

In this section, we introduce property analysis, privacy analysis, and security analysis to illustrate the feasibility of our PAID.

6.1. Property Analysis. In this section, we introduce eligibility, zero knowledge, payment rationality, and budget rationality of our PAID.

Theorem 1. (eligibility) If the data \mathcal{D}_i ($\mathcal{D}_i = (x_i, \tau_i, \widehat{t}_i, \widetilde{t}_i)$) collected by users do not meet the eligibility requirement \mathcal{E} , these users cannot pass the eligibility assessment.

Proof. We assume that the user's data are denoted as s , and the eligibility requirement interval is $[a, b]$. The user gets ciphertext $E(s)$ using homomorphic encryption. Then, \mathcal{S} picks different random k, b and constructs a monotone increasing (or decreasing) function $f(x) = kx + b$. Then, \mathcal{S} computes $f(a)$, $f(b)$, and $c = E(s)^k E(b) = E(ks + b)$. When receiving $f(a)$, $f(b)$, c from \mathcal{S} , the user decrypts c to get $f(s)$ and compares the sizes of $f(a)$, $f(b)$, $f(s)$. Because the user does not know the monotonicity of the function, it is impossible to determine the size relationship among the three numbers. Therefore, if the user's data are not qualified, then it cannot pass the qualification judgment. \square

Theorem 2. (zero knowledge) The server \mathcal{S} can determine whether the user's data meet the eligibility requirements, but it cannot know the user's specific data content.

Proof. Similar to the description in Theorem 1, we assume that the user's data are s , and the server \mathcal{S} can receive the user's homomorphic encrypted ciphertext $E(s)$. Since the Paillier cryptosystem is indistinguishable under the chosen plaintext attack, a malicious user has no way to recover the plaintext s . The server \mathcal{S} may be curious about each user's data, but it cannot obtain each user's data s without knowing the secret key. \square

Theorem 3. (payment rationality) If an honest user u_i provides qualified data, u_i can obtain a non-negative utility.

Proof. The utility ut_i of each user u_i is determined by the cost of u_i and the real reward from task publisher TP, i.e., $ut_i = p_i - c_i$.

If the data provided by an untrusted user u_i are not qualified, u_i cannot pass the eligibility assessment, so the untrusted user's utility $ut_i = 0$. However, when $c_i > (B - \pi/N)$ ($(B - \pi/N)$ is the posted lowest pricing), an honest user u_i will refuse to participate in the sensing task, so the trusted user's utility $ut_i = 0$. When $c_i \leq (B - \pi/N)$, an honest user u_i will participate in the sensing task and earn a reward $p_i = (B/m) + \pi \cdot (q_i - \bar{q})$. Since $\bar{q} = \sum_{i=1}^m q_i/m = 1/m$, $0 < q_i < 1$, and $m \leq N$, we have

$$p_i = \frac{B}{m} + \pi \cdot (q_i - \bar{q}) = \frac{B - \pi}{m} + \pi \cdot q_i > \frac{B - \pi}{N}. \quad (26)$$

Therefore, we can know that $p_i - c_i > 0$. To summarize, a user's real reward is always non-negative. \square

Theorem 4. (budget rationality) The total payment of the task publisher TP is no larger than budget B in our PAID.

Proof. The total rewards for all users are calculated as follows.

$$\begin{aligned} \sum_{i=0}^m p_i &= \sum_{i=0}^m \left(\frac{B}{m} + \pi \cdot (q_i - \bar{q}) \right) \\ &= B + \pi \cdot \sum_{i=0}^m \left(q_i - \frac{\sum_{i=1}^m q_i}{m} \right) \\ &= B + \pi \cdot \left(\sum_{i=0}^m q_i - 1 \right) \\ &= B. \end{aligned} \quad (27)$$

Hence, $\sum_{i=0}^m p_i \leq B$, i.e., our PAID is budget rational. \square

6.2. Privacy Analysis. In this section, we demonstrate the protection of user's sensing data, location, and identity privacy in our PAID.

Theorem 5. (data and location privacy protection) Except for the user himself, other parties cannot obtain the user's sensing data and location data.

Proof. In PAID, the objects that steal users' data and location privacy are mainly the server \mathcal{S} and external attackers. Specifically, the server \mathcal{S} may obtain users' sensing data and location privacy in eligibility assessment and truth discovery. External attackers steal data and location privacy by eavesdropping on the communication between the server \mathcal{S} and users.

According to Theorem 2, we can know that our PAID has the property of zero knowledge, so the server \mathcal{S} cannot learn users' sensing data and location data in the eligibility assessment. In truth discovery, users' sensing data are sent to \mathcal{S} after performing the double-masking scheme. However, the server \mathcal{S} cannot recover users' raw sensing data by double-masking sensing data. Furthermore, before the communication between the user u_i and \mathcal{S} , the data are encrypted by AES symmetric encryption function $\text{SEnc}(y_i, k_i)$. Therefore, as long as $\text{SEnc}(y_i, k_i)$ is secure, external attackers cannot steal the data y_i by eavesdropping communication. \square

Theorem 6. (identity privacy protection) When users participate in a sensing task, they use an anonymous identity rather than their real identity. Therefore, any PPT adversary cannot distinguish the users' identities.

Proof. In PAID, the anonymous identity of a user u_i is represented by $k_i \leftarrow \text{KA.agree}(SK_i, PK_{\mathcal{S}}^i)$, and the real identity of u_i is SK_i where $SK_i = x_i \leftarrow Z_q$, and $PK_{\mathcal{S}}^i = g^{x_i s}$ ($PK_{\mathcal{S}}^i$ is a token assigned by \mathcal{S}). The user u_i uses an anonymous identity k_i rather than a real identity SK_i to participate in a sensing task. Because of the DDH problem, the PPT adversary cannot get the real identity SK_i of the user u_i by the anonymous identity k_i . We omit the detailed proof, and interested readers can learn more details in the literature [36]. \square

6.3. *Security Analysis.* In this section, we describe the attacks our PAID can resist, including denial of payment attack (DoP), inference attack (IA), data pollution attack (DPA), and Sybil attack (SA).

- (1) *Resistance to Denial of Payment Attack (DoP).* We use the prepayment mechanism in our PAID. At the beginning of a sensing task, the task publisher TP pays the monetary rewards of users to \mathcal{S} in advance. If a malicious TP refuses to pay the monetary reward after receiving the data, \mathcal{S} can pay the reward to users according to the reward distribution formula. Therefore, the TP cannot refuse to pay users the reward.
- (2) *Resistance to Inference Attack (IA).* The server \mathcal{S} cannot initiate an inference attack on users' data due to the zero-knowledge property of our PAID.
- (3) *Resistance to Data Pollution Attack (DPA).* Our PAID introduces eligibility assessment, and the unqualified data submitted by users are not used in the truth discovery algorithm. Therefore, our PAID can resist the data pollution attack (DPA).
- (4) *Resistance to Sybil Attack (SA).* The anonymous identity k_i of a user u_i needs the information PK_i provided by the user and the token $PK_{\mathcal{S}}^i$ assigned by \mathcal{S} . Each user can only obtain one token from \mathcal{S} and then get the anonymous identity k_i using the key agreement algorithm. Hence, untrusted users cannot forge vast fake identities to launch the Sybil attack (SA).

7. Performance Evaluation

In this section, we use a temperature dataset from Roma for performance evaluation. First, we describe the computational and communication overhead of the eligibility assessment. Then, we show the performance of the truth discovery algorithm. Finally, the comparison with the related work shows that the quality quantification and incentive mechanism are effective.

In our experiment, the server has Intel(R) Xeon(R) E3-1231v3 3.4 GHz CPU, 16 GB RAM, 256 GB SSD, and 1 TB mechanical hard disk and runs on Ubuntu 18.04 operating system. These mobile devices are equipped with Android system with 2.2 GHz CPU and 4 GB RAM. The Roma temperature dataset includes users' ID, date, time, longitude, latitude, and temperature. In particular, the range accuracy of location, time, and sensing data (temperature) is 1 meter, 1 second, and 0.01°C, respectively. Before performing the eligibility assessment, we convert the decimal interval to the corresponding integer interval by moving the decimal point to the right. Figure 3 shows the statistical results of 232 qualified users. And we select 100 data from all qualified data for performance evaluation.

7.1. *Evaluation of Eligibility Assessment.* In this section, we analyze the computational and communication overhead in the eligibility assessment. Table 2 shows the performance comparison between our PAID and related work.

7.1.1. *Computational Overhead.* The Paillier homomorphic encryption requires two exponents (exp), one multiplication (mul), and one modular operation (mod). One decryption operation needs to perform two exponents (exp), three divisions (div), and two modular operations (mod). And in our interval judgment scheme, the user u_i needs to perform one encryption and one decryption, so the computational cost of the user is $4n \cdot \text{exp} + n \cdot \text{mul} + 3n \cdot \text{div} + 3n \cdot \text{mod}$, where n is the number of users. The server needs to perform one encryption $E(b)$ and calculates $c = E(x_i)^k E(b)$, so the computational overhead of the server is $3n \cdot \text{exp} + 2n \cdot \text{mul} + 2n \cdot \text{mod}$. Consequently, the total computational overhead is $n \cdot (7 \cdot \text{exp} + 3 \cdot \text{mul} + 3 \cdot \text{div} + 5 \cdot \text{mod})$ and the computation complexity of the interval judgment scheme is $O(n)$.

7.1.2. *Communication Overhead.* According to our interval judgment scheme, users need to send encrypted data $E(x_i)$ to the server \mathcal{S} , and the communication overhead is $\|N^2\|$ bits, where N is the product of two large primes p, q . After receiving the encrypted data $E(x_i)$, the server \mathcal{S} calculates $c = E(x_i)^k E(b)$ and sends it to the user. The communication overhead is $\|c\|$ bits, where $\|c\|$ denotes the bit length of ciphertext. So, we can conclude that the total communication overhead is $\|N^2\| + \|c\|$ bits.

7.2. *Evaluation of Truth Discovery.* In this section, we select 100 users to participate in the performance comparison of truth discovery. We compare the truth discovery of our PAID with the related work from five aspects, including accuracy, convergence, robustness to users dropping out, computational overhead, and communication overhead. The evaluation results show that our truth discovery algorithm has good accuracy, quick convergence, and high robustness to users dropping out. Besides, the computational overhead and communication overhead of our algorithm are better than those of the related work. Therefore, our truth discovery algorithm is reasonable.

7.2.1. *Accuracy.* We utilize the root of mean squared error (RMSE) to measure the resulting accuracy between PAID and CRH [32]. Figure 4 shows that the accuracy rates of PAID and CRH are similar when different numbers of users participate in a sensing task.

7.2.2. *Convergence.* To prove the convergence ability of our truth discovery algorithm in PAID, we choose four different initial values to calculate the error rate of ground truth. As shown in Figure 5, our PAID can converge quickly in a few iterations when choosing different initial values.

7.2.3. *Robustness to Users Dropping Out.* To analyze the robustness of our PAID to dropped users, we count the number of PAID failures and compare with related work PPTD [41]. Failure means that the model cannot continue to

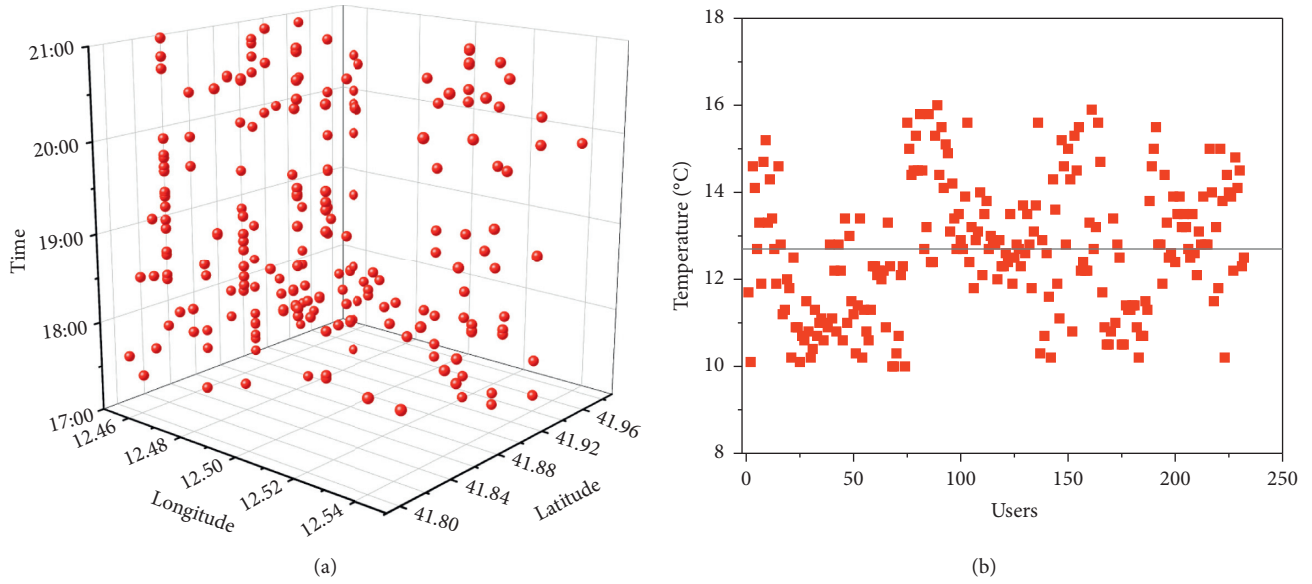


FIGURE 3: Statistics of dataset. (a) Space-time statistics of users. (b) Temperature statistics of users.

TABLE 2: Performance comparison between PAID and related work.

Protocol	Computational overhead	Communication overhead (bits)
PAID	$n \cdot (7 \cdot \text{exp} + 3 \cdot \text{mul} + 3 \cdot \text{div} + 5 \cdot \text{mod})$	$\ N^2\ + \ c\ $
[20]	$n \cdot (10 \cdot \text{exp} + 5 \cdot \text{mul} + 5 \cdot \text{comp} + 5 \cdot \text{mod})$	$3\ p\ + 2\ c\ $

Note. p is a large prime. And exp, mul, div, comp, and mod represent one exponent arithmetic, one multiplication, one division, one comparison operation, and one modular arithmetic.

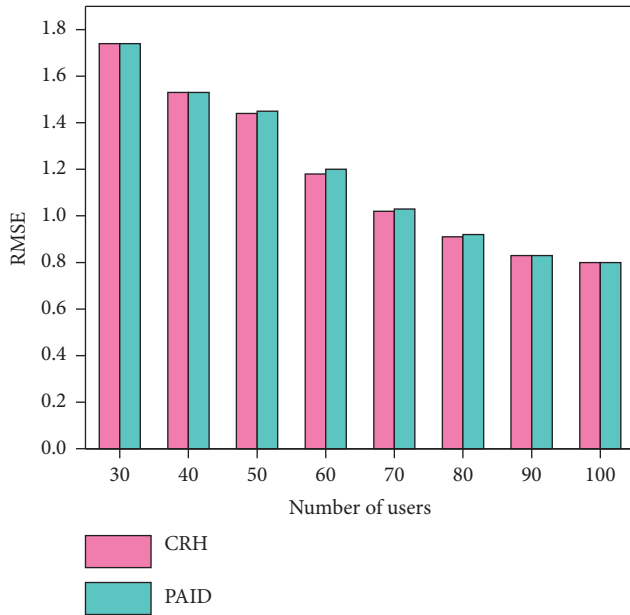


FIGURE 4: Accuracy evaluation.

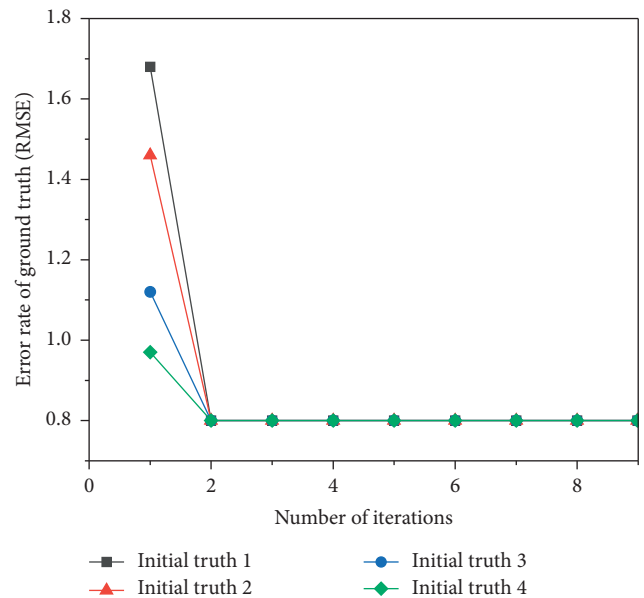


FIGURE 5: Convergence evaluation.

run and have to restart because of users' exit. In the PPTD, it is considered as a failure once a user quits in the whole truth discovery process. In our PAID, it is deemed to be a failure only when the number of online users is less than the threshold t ($t = 25$ in our experiment). And we repeat the experiment 50 times to count the failure times of the two

models. Figure 6 shows the failure times of the two models when different users participate in a sensing task. We can know that the number of PPTD failures increases as the number of users increases. However, as long as online users' number is greater than the threshold, our PAID is robust to dropped users.

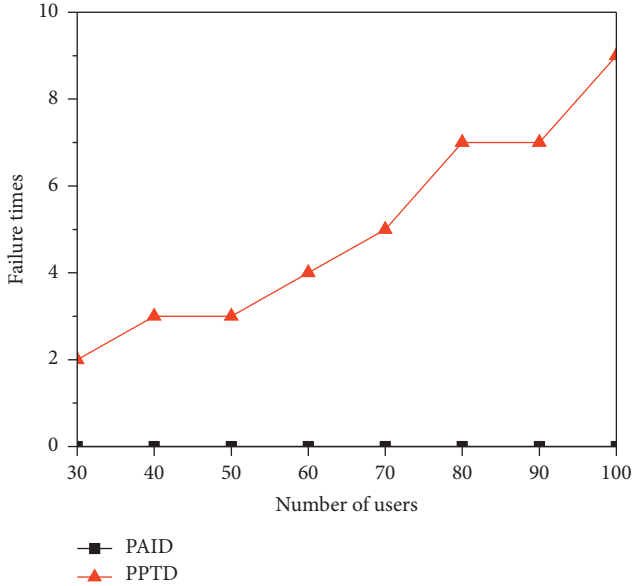


FIGURE 6: Robustness evaluation.

7.2.4. Computational Overhead. We compare the computational overhead of PAID and PPTD [41]. Figure 7 shows the running time of the two schemes for different users. It is evident that the running time of our PAID is far less than that of PPTD.

7.2.5. Communication Overhead. We count the communication overhead of users in a complete iterative process and compare our scheme with PPTD [41]. And we do not count the server's communication overhead because we can regard the total communication overhead of all users as the communication cost of the server. Table 3 shows that the communication overhead of our PAID is far less than that of PPTD, although the number of users is different.

7.3. Evaluation of Incentive Mechanism. In this section, we compare the monetary rewards of our PAID and related work. In the experiment, we select 100 users, including 80 qualified users and 20 unqualified users. And the budget $B = 100$, $\pi = 0.3$. DQTE [42] is a scheme that includes unqualified users in reward distribution, while DQTE+ removes unqualified users before reward distribution. As Figure 8 shows, users in DQTE get almost the same rewards. Although DQTE+ removes unqualified users, there is no obvious difference for users' rewards except for the increase in each user's monetary rewards. However, our scheme can provide higher monetary rewards for users who submit higher quality data. Therefore, our scheme can effectively motivate users to provide high-quality sensing data.

8. Related Work

Truth discovery is an effective technology that can calculate the ground truth and users' data quality from conflicting sensing data. Li et al. [32] proposed a general truth discovery scheme, but privacy protection is not in their work

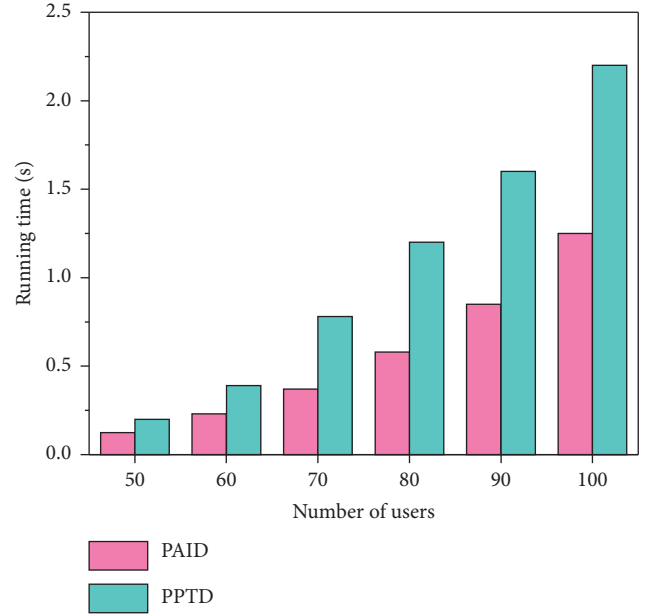


FIGURE 7: Running time for different number of users.

TABLE 3: Communication overhead of users (kB).

Number of users	PAID	PPTD
50	0.16	1.38
60	0.24	1.78
70	0.32	1.86
80	0.45	2.31
90	0.58	2.63
100	0.76	2.97

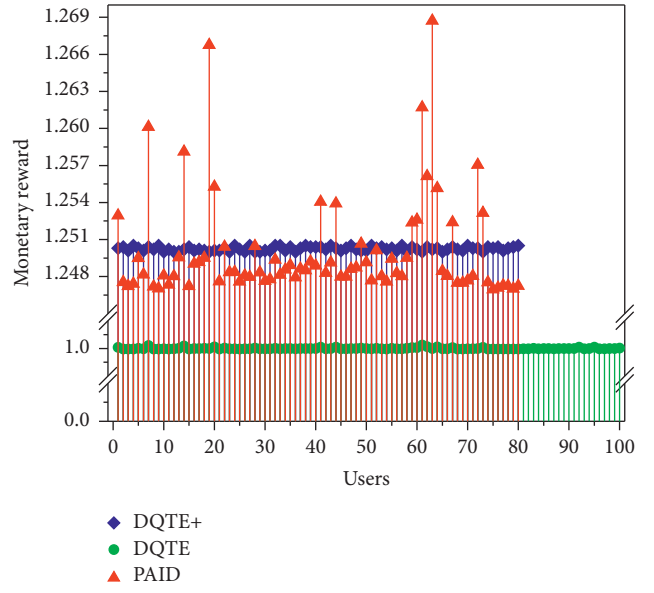


FIGURE 8: The reward evaluation of users.

scope. To protect users' privacy data, Miao et al. [41] proposed the first privacy-preserving truth discovery scheme using the Paillier cryptosystem, but the computational and communication costs are huge. Zheng et al.

[43] designed a privacy-aware truth discovery, which greatly reduced the computational and communication overhead through a secure sum protocol. Zhang et al. [44] designed a truth discovery scheme using a one-way hash chain to ensure privacy security, and all truth discovery operations are completed by fog and cloud platforms. Tang et al. [45] used two servers to complete the calculation process of truth discovery, which can effectively protect users' sensing data privacy. However, these works do not take into account the failure of the MCS system caused by users' exit. Bonawitz et al. [39] proposed a double-masking scheme for secure data aggregation, and this scheme allows users to exit. After that, Xu et al. [15] designed a privacy-preserving truth discovery scheme based on the double-masking scheme. However, these truth discovery schemes do not incorporate incentive mechanisms. If malicious users constantly input erroneous data, it will affect the reliability of the results in the MCS system.

Another previous work [42, 46] related to this paper is the incentive mechanism in the MCS system. Zhang et al. [47] presented a reverse auction model which can motivate online users to participate in sensing tasks. Jin et al. [16] designed an incentive mechanism model based on reverse combinatorial auctions, which can maximize social welfare and effectively motivate users. Yang et al. [42] introduced a quality-aware incentive mechanism, which can distribute rewards to users after calculating the data quality. However, these works do not consider the privacy of users. In [27], the authors designed a privacy-preserving incentive mechanism model. Nevertheless, these solutions can not eliminate users who provide error data. Zhao et al. [20] presented an incentive mechanism model to evaluate the reliability of users' data while protecting data privacy. Still, the user's sensing data need to be submitted to the task publisher, so the privacy protection of sensing data is still insufficient. Later, Zhao et al. [48] proposed a privacy-preserving incentive mechanism based on truth discovery. This model uses two servers to achieve real-time reward distribution while protecting users' privacy. However, most existing works do not take users' exit into account.

9. Conclusion

In this paper, we propose a privacy-preserving incentive mechanism based on truth discovery in the MCS system. Specifically, we introduce an eligibility assessment scheme to estimate whether the data submitted by users are qualified. Next, the truth discovery scheme calculates the ground truth and the weight of each user. Then, we quantify the data quality of users by the weight and distribute the rewards. Besides, we also demonstrate that PAID meets eligibility, zero knowledge, payment rationality, and budget rationality. And the analysis shows that our PAID can resist the denial of payment attack, inference attack, data pollution attack, and Sybil attack. Finally, experiments illustrate that PAID is effective, efficient, and robust to dropped users. In future work, we will design an incentive mechanism model for the application of multidimensional sensing data collection.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (nos. 61962022 and 62062034) and Key Research and Development Plan of Jiangxi Province (no. 20192BBE50077).

References

- [1] R. Gao, M. Zhao, T. Ye et al., "Jigsaw: indoor floor plan reconstruction via mobile crowdsensing," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, pp. 249–260, Association for Computing Machinery, Maui Hawaii, US, September 2014.
- [2] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An ai-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of iot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2019.
- [3] X. Wei, B. Sun, and J. Cui, "Task replica assignment in mobile self-organized crowdsensing," *International Journal of Performance Engineering*, vol. 16, no. 1, pp. 152–162, 2020.
- [4] Z. Wang, J. Zhao, J. Hu et al., "Towards personalized task-oriented worker recruitment in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 2080–2093, 2021.
- [5] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, "A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2347–2360, 2020.
- [6] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "Greengps: a participatory sensing fuel-efficient maps application," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, pp. 151–164, San Francisco, California, USA, June 2010.
- [7] K. Abualsaud, T. M. Elfouly, T. Khatib et al., "A survey on mobile crowd-sensing and its applications in the iot era," *IEEE Access*, vol. 7, pp. 3855–3881, 2018.
- [8] Z. Yin, C. Wu, Z. Yang, and Y. Liu, "Peer-to-peer indoor navigation using smartphones," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1141–1153, 2017.
- [9] F. Montori, L. Bedogni, and L. Bononi, "A collaborative internet of things architecture for smart cities and environmental monitoring," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 592–605, 2017.
- [10] S. Zhang, H. Li, Y. Dai, J. Li, M. He, and R. Lu, "Verifiable outsourcing computation for matrix multiplication with improved efficiency and applicability," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5076–5088, 2018.
- [11] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in internet of things with privacy preserving: challenges, solutions and opportunities," *IEEE Network*, vol. 32, no. 6, pp. 144–151, 2018.

- [12] R. W. Ouyang, M. Srivastava, A. Toniolo, and T. J. Norman, "Truth discovery in crowdsourced detection of spatial events," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 1047–1060, 2015.
- [13] Y. Li, J. Gao, C. Meng et al., "A survey on truth discovery," *ACM Sigkdd Explorations Newsletter*, vol. 17, no. 2, pp. 1–16, 2016.
- [14] F. Ma, Y. Li, Q. Li et al., "Faitcrowd: fine grained truth discovery for crowdsourced data aggregation," in *Proceedings of the 21th Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pp. 745–754, Sydney, NSW, Australia, August 2015.
- [15] G. Xu, H. Li, S. Liu, M. Wen, and R. Lu, "Efficient and privacy-preserving truth discovery in mobile crowd sensing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3854–3865, 2019.
- [16] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 167–176, Hangzhou, China, June 2015.
- [17] H. Jin, L. Su, and K. Nahrstedt, "Theseus: incentivizing truth discovery in mobile crowd sensing systems," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 1–10, Chennai India, July 2017.
- [18] X. Zhang, Z. Yang, W. Sun et al., "Incentives for mobile crowd sensing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2015.
- [19] D. Peng, F. Wu, and G. Chen, "Data quality guided incentive mechanism design for crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 307–319, 2017.
- [20] B. Zhao, S. Tang, X. Liu, and X. Zhang, "Pace: privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1924–1939, 2020.
- [21] D. Dharminder and D. Mishra, "Lcpga: lattice-based conditional privacy preserving authentication in vehicular communication," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 2, p. e3810, 2020.
- [22] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
- [23] K. Huang, X. Liu, S. Fu, D. Guo, and M. Xu, "A lightweight privacy-preserving cnn feature extraction framework for mobile sensing," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1441–1455, 2021.
- [24] T. Wan, X. Liu, W. Liao, and N. Jiang, "Cryptanalysis and improvement of a smart card based authentication scheme for multi-server architecture using ecc," *IJ Network Security*, vol. 21, no. 6, pp. 993–1002, 2019.
- [25] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in iiot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2019.
- [26] B. Zhao, S. Tang, X. Liu, X. Zhang, and W.-N. Chen, "Ironm: privacy-preserving reliability estimation of heterogeneous data for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5159–5170, 2020.
- [27] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *Proceedings of IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2053–2061, IEEE, Paris, France, April 2019.
- [28] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in dwsns," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
- [29] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Incentive mechanism for privacy-aware data aggregation in mobile crowd sensing systems," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2019–2032, 2018.
- [30] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4w1h in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 9, pp. 42–48, 2014.
- [31] L. Zhang, Y. Li, X. Xiao et al., "Crowdbuy: privacy-friendly image dataset purchasing via crowdsourcing," in *Proceedings of IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 2735–2743, IEEE, Honolulu, HI, USA, April 2018.
- [32] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pp. 1187–1198, Snowbird, Utah USA, June 2014.
- [33] H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: when qoe meets qop," *IEEE Wireless Communications*, vol. 22, no. 4, pp. 74–80, 2015.
- [34] H. Li, D. Liu, Y. Dai, T. H. Luan, and S. Yu, "Personalized search over encrypted data with efficient and secure updates in mobile clouds," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 97–109, 2015.
- [35] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [36] H. Krawczyk, "SIGMA: the 'SIGn-and-MAC' approach to authenticated diffie-hellman and its use in the IKE protocols," in *Proceedings of Annual International Cryptology Conference*, pp. 400–425, Springer, Santa Barbara, California, USA, August 2003.
- [37] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, Springer, Prague, Czech Republic, May 1999.
- [38] L. Chen, W. Zhang, S. Li, Q. Huang, and Z. Chen, "Fully privacy-preserving determination of point-range relationship," *SCIENTIA SINICA Informationis*, vol. 48, no. 2, pp. 187–204, 2018.
- [39] K. Bonawitz, V. Ivanov, B. Kreuter et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, Dallas, Texas, USA, October 2017.
- [40] Y. Jiang, B. Zhao, S. Tang, and H. T. Wu, "A verifiable and privacy-preserving multidimensional data aggregation scheme in mobile crowdsensing," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 5, p. e4008, 2021.
- [41] C. Miao, W. Jiang, L. Su et al., "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pp. 183–196, Seoul, South Korea, November 2015.
- [42] S. Yang, F. Wu, S. Tang, X. Gao, B. Yang, and G. Chen, "On designing data quality-aware truth estimation and surplus

- sharing method for mobile crowdsensing,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 832–847, 2017.
- [43] Y. Zheng, H. Duan, X. Yuan, and C. Wang, “Privacy-aware and efficient mobile crowdsensing with truth discovery,” *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 1, pp. 121–133, 2017.
- [44] C. Zhang, L. Zhu, C. Xu, X. Liu, and K. Sharif, “Reliable and privacy-preserving truth discovery for mobile crowdsensing systems,” *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1245–1260, 2019.
- [45] J. Tang, S. Fu, X. Liu, Y. Luo, and M. Xu, “Achieving privacy-preserving and lightweight truth discovery in mobile crowdsensing,” *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2021.
- [46] D. Peng, F. Wu, and G. Chen, “Pay as how well you do: a quality based incentive mechanism for crowdsensing,” in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 177–186, Hangzhou China, June 2015.
- [47] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, “Free market of crowdsourcing: incentive mechanism design for mobile sensing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3190–3200, 2014.
- [48] B. Zhao, X. Liu, W. N. Chen et al., “Privacy and reliability-aware real-time incentive system for crowdsensing,” *IEEE Internet of Things Journal*, p. 1, 2021.

Research Article

An Efficient and Provable Multifactor Mutual Authentication Protocol for Multigateway Wireless Sensor Networks

Shuailiang Zhang ^{1,2}, Xiujuan Du ^{1,2} and Xin Liu^{1,2}

¹Computer Department, Qinghai Normal University, Xining 810008, China

²Academy of Plateau Science and Sustainability, Xining 810008, China

Correspondence should be addressed to Xiujuan Du; dxj@qhnu.edu.cn

Received 19 April 2021; Revised 25 June 2021; Accepted 19 July 2021; Published 4 August 2021

Academic Editor: James Ying

Copyright © 2021 Shuailiang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As the most popular way of communication technology at the moment, wireless sensor networks have been widely concerned by academia and industry and plays an important role in military, agriculture, medicine, and other fields. Identity authentication offers the first line of defence to ensure the security communication of wireless sensor networks. Since the sensor nodes are resource-limited in the wireless networks, how to design an efficient and secure protocol is extremely significant. The current authentication protocols have the problem that the sensor nodes need to execute heavy calculation and communication consumption during the authentication process and cannot resist node capture attack, and the protocols also cannot provide perfect forward and backward security and cannot resist replay attack. Multifactor identity authentication protocols can provide a higher rank of security than single-factor and two-factor identity authentication protocols. The multigateway wireless sensor networks' structure can provide a larger communication coverage area than the single-gateway network structure, so it has become the focus of recent studies. Therefore, we design a novel multifactor authentication protocol for multigateway wireless sensor networks, which only apply the lightweight hash function and are given biometric information to achieve a higher level of security and efficiency and a larger communication coverage area. We separately apply BAN logic, random oracle model, and AVISPA tool to validate the security of our authentication protocol in Case 1 and Case 2. We put forward sixteen evaluation criteria to comprehensively evaluate our authentication protocol. Compared with the related authentication protocols, our authentication protocol is able to achieve higher security and efficiency.

1. Introduction

As the prevalent way of communication and the significant section of the Internet of Things, wireless sensor networks are composed of massive sensor nodes, which have collection and computing abilities, and communicate with the corresponding communication parties via wireless technology [1]. Wireless sensor networks' communications are widely applied in military, industrial, agricultural monitoring, wearable health monitoring systems, smart home environment, intelligent transportation systems, and other fields. These sensor nodes are small and resource-constrained, and they are often randomly deployed in unattended or hostile region under the regulation of one or more gateway nodes to gather and transmit the information on

public network channel [2]. Due to the characteristics of the communication channel in wireless sensor network, the communication information is prone to various types of attacks. Mutual authentication plays a significant role in guaranteeing the security among the existing security mechanisms [3] and is considered as the basic access control that the user must first pass through the verification of the sensor node before accessing the gathered information [4].

The current identity authentication technology can be divided into three types: password based single-factor authentication technology, password and smart card based two-factor authentication technology, and password, smart card, and biometric based three-factor authentication technology [5]. The aforementioned third type is the most commonly used authentication technology, and it enhances

the security of the wireless network works to a higher level [6, 7]. At present, most of the researches are keen on the identity authentication technology of single gateway, while only a few people are engaged in identity authentication technology of multigateway structure [8]. We can apply multiple gateway nodes to extend the communication coverage area and increase scalability [9]. However, the current multigateway authentication technology has some disadvantages such as high computational complexity and heavy storage consumption and is vulnerable to various attacks. Therefore, for the sake of eliminating the security flaws and increasing the computation efficiency, we design a novel lightweight mutual authentication protocol for the multiple gateway nodes networks.

1.1. Network Model. As shown in Figure 1, it involves three communication entities, that is, sensor nodes, home/foreign gateway node, and user in case 1. The sensor node and user should complete registration at the gateway node. After registration, the user delivers the login request to the gateway node. The gateway authenticates and is in charge of transmitting authentication information between the user and the sensor node. After completing authentication process, the registered user has ability to obtain information gathered by the sensors under the negotiated session key.

As shown in Figure 2, it involves four communication entities, that is, sensor nodes, home gateway node, foreign gateway node, and user in case 2. In addition to completing the authentication of case 1, it is also necessary to achieve the authentication between the home gateway node and the foreign gateway node.

1.2. Related Works. Gope and Hwang [10] proposed an efficient and secure authentication scheme and claimed that their scheme is able to preserve the user anonymity for roaming services in global mobility networks by way of using the one-way hash function operation. Xu et al. [11] discovered that the scheme of Gope and Hwang is vulnerable to replay attacks and has a heavy storage cost. Similarly, Lu et al. [3] also pointed out that scheme of Gope and Hwang is susceptible to specific known information attack, and the password alteration section is inaccurate. Fan et al. [12] found that the scheme of Gope and Hwang is vulnerable to offline guessing attack and the desynchronization attack and does not retain robust forward security. Then, they proposed a novel efficient mutual and key agreement scheme with desynchronization for anonymous roaming service in global mobility networks. However, Mohit and Narendra [13] reviewed the scheme of Wu and showed that the scheme has the problem of the traceability of the mobile user and inefficient wrong password detection.

In order to preserve security and privacy and reduce communication and computation costs, Das et al. [14] proposed a biometric-based authentication protocol for the Industrial Internet of Things. Unfortunately, Hussain and Chaudhry [15] discovered that the protocol of Das et al. is unable to prevent the assailant from obtaining the public parameters kept in the smart device and fails to resist session

key attack and achieve perfect forward secrecy. So, against offline password guessing attack and user impersonation attack, Amin et al. [16] demonstrated a secure three-factor mutual authentication protocol, and this protocol lengthens the lifetime of network by means of decreasing the cost of sensor nodes. Later, Sharif et al. [17] claimed that the protocol of Amin et al. cannot boycott strong replay attacks and cannot realize the perfect forward secrecy. However, Wu et al. [18] pointed out that both of the two protocols [14, 17] suffer from under offline surmising attack.

To overcome user and sensor node impersonation attacks, He et al. [19] introduced a novel mutual authentication design based on the temporal credential for wireless sensor networks. Afterwards, Kumari et al. [20] demonstrated that there are seven security problems in the design of He et al. Jiang et al. [21] revealed that the design of He et al. is prone to malicious user impersonation attack, stolen smart card attack, and tracking attack in the authentication process and proposed an untraceable and secure two-factor authentication design based on elliptic curve cryptography for wireless sensor networks. After analyzing the design of Jiang et al., Xiong et al. [22] received the result that the design has no detection mechanism for unauthorized login and clock synchronization problem and introduced a three-factor anonymous authentication design for wireless sensor networks by applying the fuzzy commitment to deal with biometric information.

For the purpose of withstanding the node capture attack, impersonation attack, and man-in-the-middle attack, Das [23] then put forward an original biometric-based mutual authentication design for wireless sensor networks. In the same year, Lu et al. [24] found that the design of Das does not really implement the three-factor security and user anonymity and has no ability to boycott user impersonation attack. Li et al. [25] pointed out that the design of Ruhul et al. [26] is vulnerable to DoS attack and lacks forward secrecy. In view of previous studies, Li introduced a three-factor mutual authentication design with forward secrecy for wireless medical sensor networks, which settles the contradiction of local password verification and mobile device lost attack via fuzzy verifier and honey_list technology. Nevertheless, Mo and Chen [27] discovered that the protocol of Xiong et al. [22] is vulnerable to resist stolen smart card attack and divulge the biometric information. Mo and Chen [27] pointed out that the protocol of Lu et al. [24] is prone to known session-specific temporary information attack and cannot realize three-factor security and backward secrecy. Mo and Chen [27] found that the protocol of Li et al. [25] is susceptible to withstanding replay attack.

Mutual authentication is used to supply the fundamental security requirement by confirming the legality of the communication parities for various network applications, such as smart city [28, 29], Internet of Drones [30, 31], vehicular networks [32, 33], multiserver environment [34, 35], and mobile device [36, 37].

1.3. Organization. The remainder of the paper is organized as follows. In part 2, we discuss the preliminaries. In part 3, we present our proposed mutual authentication protocol. In

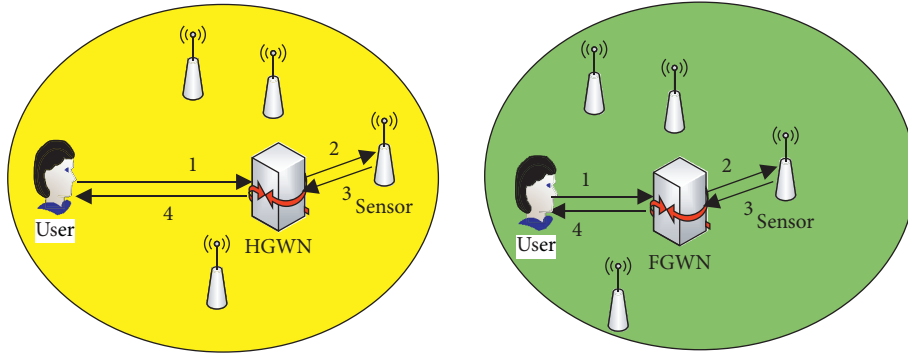


FIGURE 1: Network architecture in case 1.

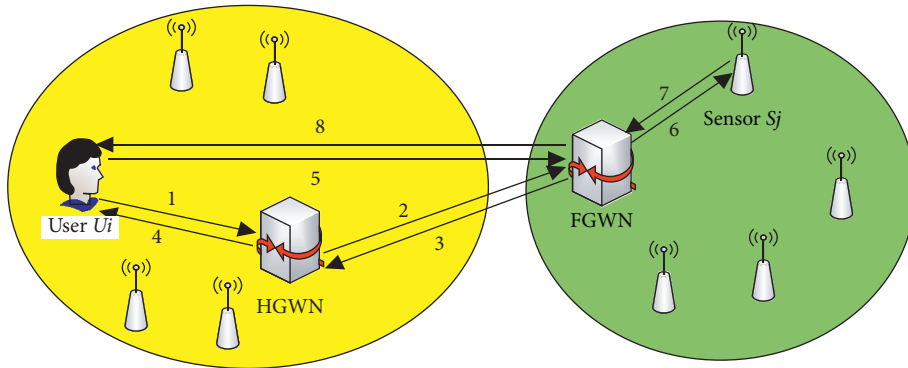


FIGURE 2: Network architecture in case 2.

part 4, we show formal analysis of our proposed mutual authentication protocol through three methods, that is, BAN logic, random oracle model, and AVISPA. In part 5, we demonstrate informal analysis of our proposed mutual authentication protocol through sixteen security authentication protocol evaluation criteria. In part 6, we compare our proposed mutual authentication protocol with other related authentication protocols in terms of security, computation time, and communication cost. Finally, we come to a conclusion in part 7.

2. Preliminaries

This part presents the preliminaries in our designed mutual authentication protocol involving biometric fuzzy extractor, threat model, and protocol evaluation criteria.

2.1. Biometric Fuzzy Extractor. So as to prevent the given biometric information BIO_i from various noises in the process of information acquisition, this paper introduces the biometric fuzzy extractor. There are two functions in biometric fuzzy extractor [28, 36]: GEN function and REP function. The concrete representations of the two functions are as follows:

- (1) $GEN(BIO_i) = (\sigma_i, \tau_i)$. GEN is a probabilistic generation function that separates out the secret string σ_i and an auxiliary string τ_i from the given biometric information BIO_i

- (2) $REP(BIO_i, \tau_i) = \sigma_i$. REP is a deterministic function that recovers the secret string σ_i from the given biometric information BIO_i with the assistance of the auxiliary string τ_i

2.2. Threat Model. The threat model presents the possibilities of an assailant obtaining the information about the authentication protocol without authorizing and the competence of potential destruction. Before evaluating the security authentication protocol, we assume that the assailant has the following abilities in the authentication process:

- (1) The assailant is able to revise, intercept, delete, and transmit the communication information on the public network channel [38, 39]
- (2) The assailant is able to obtain the parameters kept in the smart card via power analysis attack [40], in case the smart card is stolen or lost
- (3) The assailant is able to carry out the online and offline password guessing attack [35]
- (4) The assailant is able to implement the impersonation attack [4]
- (5) The assailant is aware of the authentication protocol system [41]
- (6) The assailant may be a legitimate user or an external entity [42, 43]

2.3. Protocol Evaluation Criteria. Since the information is interacted on the public network channel, the assailant is able to intercept and manipulate the interactive information [41, 44]. To guarantee the security of the interactive information on the public network channel, we design a mutual authentication and session key agreement protocol among the communication parties for the multiple gateway nodes networks. From four aspects of users, gateway nodes, sensor nodes, and communication protocol itself, we define the following sixteen security authentication protocol evaluation criteria:

- (1) Session key security
- (2) Three-factor security
- (3) Perfect forward and backward security
- (4) Resist sensor node capture attack
- (5) Resist stolen smart card attack
- (6) Resist user impersonation attack
- (7) Resist gateway impersonation attack
- (8) Resist sensor node impersonation attack
- (9) Resist reply attack
- (10) Resist privileged insider attack
- (11) Resist online password-guessing attack
- (12) Resist offline password-guessing attack
- (13) Resist user tracking attack
- (14) Biometric template protection
- (15) Mutual authentication
- (16) User anonymity

3. The Proposed Protocol

In this part, we will demonstrate our three-factor remote user authentication and key agreement protocol in the wireless sensor network environment with multiple gateways. Our protocol is related to five sections, which are initialization section, registration section, login section, authentication and key agreement section, and password change section.

3.1. Initialization Section. SA picks the distinctive identity ID_{SN_j} and private key SX_{SN_j} , for the SN, calculates the value $SNX_j = h(ID_{SN_j} \| SX_{SN_j})$ and dispatches the information $\{ID_{SN_j}, SNX_j\}$ to the SN. SA chooses the distinctive identity ID_{GWNh} and private key SX_{GWNh} for the HGWN. SA selects the distinctive identity ID_{GWNf} and private key SX_{GWNf} for the FGWN in the same way. Each pair of HGWN and FGWN keeps a private session key K_{hf} .

3.2. Registration Section. The registration section is divided into two parts, namely, sensor node registration and user registration.

3.2.1. Sensor Node Registration. A1: in the light of the received information $\{ID_{SN_j}, SNX_j\}$ in the initialization section, SN_j calculates $MSN_j = SNX_j \oplus h(ID_{SN_j})$ and

dispatches the information $\{ID_{SN_j}, MSN_j\}$ to GWN_H . A2: after obtaining the information sent by the SN, $HGWN$ computes $SNX_j = MSN_j \oplus h(ID_{SN_j})$, preserves the information $\{ID_{SN_j}, MSN_j\}$, and replies to the sensor node with a confirmation message.

3.2.2. User Registration

A1: U_i picks the essential parameters, identity ID_i , password PW_i , and two stochastic digits r_i and r_p and counts $UID_i = h(ID_i \| r_i)$ and $UPW_i = h(PW_i \| r_i \| r_p)$. After the calculation, U_i delivers UID_i and UPW_i to $HGWN$ as the registration request.

A2: after getting the registration request, $HGWN$ generates a stochastic digit r_{GWNh} and computes $GUID_i = h(r_{GWNh} \| SX_{GWNh} \| ID_{GWNh}) \oplus UID_i$, $GE_i = h(UID_i \| UPW_i)$, and $GF_i = GE_i \oplus GUID_i \oplus UID_i$ in combination with its own privacy parameters. $HGWN$ loads GE_i and GF_i into the smart card and transmits the smart card to U_i .

A3: after reception of the smart card, U_i imprints his or her unique biometric BIO_{U_i} on the sensor device specific terminal and further counts $GEN(BIO_{U_i}) = (\sigma_{U_i}, \tau_{U_i})$, $USC_1 = r_i \oplus h(ID_i \| PW_i \| \sigma_{U_i})$,

$USC_2 = r_p \oplus h(\sigma_{U_i} \| r_i)$, and $USC_3 = h(UID_i \| UPW_i \| \sigma_{U_i} \| r_i \| r_p)$. Then, U_i loads (USC_1, USC_2, USC_3) into the smart card.

3.3. Login Section. A1: U_i inserts smart card and inputs his or her identity ID_i , password PW_i , and biometric BIO_{U_i} . A2: smart card counts $REP(BIO_{U_i}, \tau_{U_i}) = \sigma_{U_i}$, $r_i^* = USC_1 \oplus h(ID_i \| PW_i \| \sigma_{U_i})$, $r_p^* = USC_2 \oplus h(\sigma_{U_i} \| r_i)$, $UID_i^* = h(ID_i \| r_i^*)$, $UPW_i^* = h(PW_i \| r_i^* \| r_p^*)$, and $USC_3^* = h(UID_i^* \| UPW_i^* \| \sigma_{U_i} \| r_i^* \| r_p^*)$ and confirms the correctness of the formula $USC_3^* = USC_3$. A3: if it is not right, smart card suspends the session promptly. Otherwise, smart card picks stochastic identity SCN_i , stochastic digit r_{SCN} , and time stamp T_{sc} and counts $SCG_1 = GUID_i \oplus SCN_i$, $SCG_2 = r_{SCN} \oplus h(SCN_i \| T_{sc})$, $SCG_3 = GF_i \oplus h(UID_i \| UPW_i)$, and $SCG_4 = h(SCN_i \| r_{SCN} \| T_{sc} \| GUID_i \| SCG_3 \| ID_{SN_j})$. Finally, U_i delivers the login request $(SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SN_j})$ to GWN_H .

3.4. Authentication and Key Agreement Section. On the basis of UID_i in the login request, GWN_H computes $GUID_i = h(r_{GWNh} \| SX_{GWNh} \| ID_{GWNh}) \oplus UID_i$, $SCN_i^* = GUID_i \oplus SCG_1$, $r_{SCN}^* = SCG_2 \oplus h(SCN_i^* \| T_{sc})$, $SCG_3^* = GUID_i \oplus UID_i$, and $SCG_4^* = h(SCN_i^* \| r_{SCN}^* \| T_{sc} \| GUID_i \| SCG_3^* \| ID_{SN_j})$ and confirms the correctness of the formula $SCG_4^* = SCG_4$. If it is not right, GWN_H terminates the session promptly. Otherwise, GWN_H finds whether ID_{SN_j} is in the information about the sensor node it preserves. If it is in the information, execute case 1 as shown in Figure 3; if it is not, execute case 2 as shown in Figure 4.

Case 1:

A1: GWN_H generates time stamp T_{gwnh} and computes the freshness of the login request by the formula

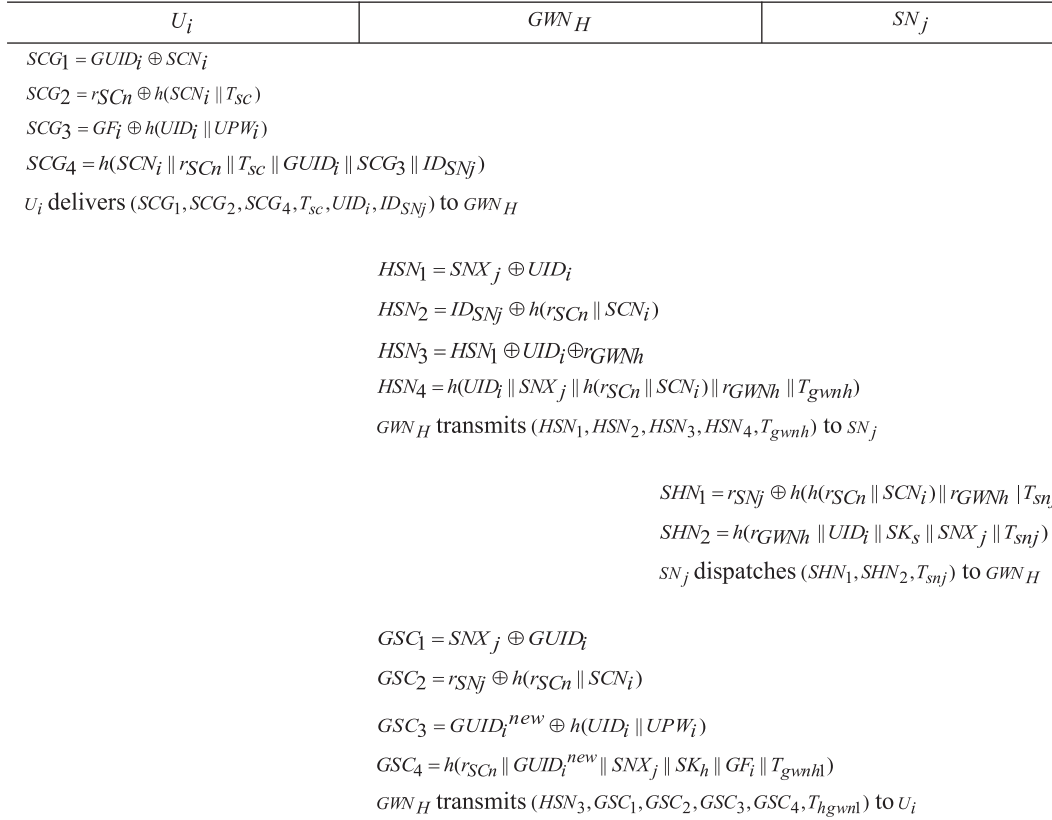


FIGURE 3: The main authentication steps in case 1.

$|T_{gwnh} - T_{sc}| \leq \Delta T$. If it is not right, GWN_H terminates the session promptly. Otherwise, GWN_H computes $HSN_1 = SNX_j \oplus UID_i$, $HSN_2 = ID_{SNj} \oplus h(r_{SCn} \parallel SCN_i)$, $HSN_3 = HSN_1 \oplus UID_i \oplus r_{GWNh}$, and $HSN_4 = h(UID_i \parallel SNX_j \parallel h(r_{SCn} \parallel SCN_i) \parallel r_{GWNh} \parallel T_{gwnh})$ and transmits the information $(HSN_1, HSN_2, HSN_3, HSN_4, T_{gwnh})$ to SN_j .

A2: upon receiving the information $(HSN_1, HSN_2, HSN_3, HSN_4, T_{gwnh})$ at time T_{snj} , SN_j calculates the freshness of the information by the formula $|T_{snj} - T_{gwnh}| \leq \Delta T$. If it is not right, SN_j ends the session promptly. Otherwise, SN_j calculates $UID_i^* = SNX_j \oplus HSN_1$, $h(r_{SCn} \parallel SCN_i)^* = ID_{SNj} \oplus HSN_2$, $r_{GWNh}^* = HSN_3 \oplus UID_i^* \oplus HSN_3$, and $HSN_4^* = h(UID_i^* \parallel SNX_j \parallel h(r_{SCn} \parallel SCN_i)^* \parallel r_{GWNh}^* \parallel T_{gwnh})$ and confirms the correctness of the formula $HSN_4^* = HSN_4$.

A3: if it is not right, SN_j ends the session promptly. Otherwise, SN_j selects stochastic digit r_{SNj} and calculates $SK_s = h(r_{SNj} \parallel r_{GWNh} \parallel UID_i \parallel h(r_{SCn} \parallel SCN_i) \parallel SNX_j)$, $SHN_1 = r_{SNj} \oplus h(h(r_{SCn} \parallel SCN_i) \parallel r_{GWNh} \parallel T_{snj})$, and $SHN_2 = h(r_{GWNh} \parallel UID_i \parallel SK_s \parallel SNX_j \parallel T_{snj})$. Then, SN_j dispatches the information (SHN_1, SHN_2, T_{snj}) to GWN_H .

A4: upon receiving the information (SHN_1, SHN_2, T_{snj}) at time T_{gwnh1} , GWN_H computes the freshness of the information by the formula

$|T_{gwnh1} - T_{snj}| \leq \Delta T$. If it is not right, GWN_H terminates the session promptly. Otherwise, GWN_H computes $r_{SNj}^* = SHN_1 \oplus h(h(r_{SCn} \parallel SCN_i) \parallel r_{GWNh} \parallel T_{snj})$, $SK_h = h(r_{SNj}^* \parallel r_{GWNh} \parallel UID_i \parallel h(r_{SCn} \parallel SCN_i) \parallel SNX_j)$, and $SHN_2^* = h(r_{GWNh} \parallel UID_i \parallel SK_h \parallel SNX_j \parallel T_{snj})$ and confirms the correctness of the formula $SHN_2^* = SHN_2$.

A5: if it is not right, GWN_H terminates the session promptly. Otherwise, GWN_H generates new stochastic digit r_{GWNh}^{new} and computes $GUID_i^{new} = h(r_{GWNh}^{new} \parallel SX_{GWNh} \parallel ID_{GWNh}) \oplus UID_i$,

$GSC_1 = SNX_j \oplus GUID_i$, $GSC_2 = r_{SNj} \oplus h(r_{SCn} \parallel SCN_i)$, $GSC_3 = GUID_i^{new} \oplus h(UID_i \parallel UPW_i)$, and $GSC_4 = h(r_{SCn} \parallel GUID_i^{new} \parallel SNX_j \parallel SK_h \parallel GF_i \parallel T_{gwnh1})$. Then, GWN_H transmits the information $(HSN_3, GSC_1, GSC_2, GSC_3, GSC_4, T_{hgwn1})$ to U_i .

A6: upon receiving the information $(HSN_3, GSC_1, GSC_2, GSC_3, GSC_4, T_{hgwn1})$ at time T_{ui} , U_i computes the freshness of the information by the formula $|T_{ui} - T_{gwnh1}| \leq \Delta T$. If it is not right, U_i suspends the session promptly. Otherwise, U_i counts $SNX_j^* = GSC_1 \oplus GUID_i$, $r_{SNj}^* = GSC_2 \oplus h(r_{SCn} \parallel SCN_i)$,

$r_{GWNh}^* = SNX_j^* \oplus HSN_3$, $SK_u = h(r_{SNj}^* \parallel r_{GWNh}^* \parallel UID_i \parallel h(r_{SCn} \parallel SCN_i) \parallel SNX_j^*)$, $GUID_i^{new} = GSC_3 \oplus h(UID_i \parallel UPW_i)$, and $GSC_4^* = h(r_{SCn} \parallel GUID_i^{new} \parallel SNX_j^* \parallel SK_u \parallel GF_i \parallel T_{gwnh1})$ and confirms the correctness of the formula $GSC_4^* = GSC_4$.

U_i	GWN_H	GWN_F	SN_j
$SCG_1 = GUID_i \oplus SCN_i$ $SCG_2 = r_{SCn} \oplus h(SCN_i \ T_{sc})$ $SCG_3 = GF_i \oplus h(UID_i \ UPW_i)$ $SCG_4 = h(SCN_i \ r_{SCn} \ T_{sc} \ GUID_i \ SCG_3 \ ID_{SNj})$ U_i delivers $(SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SNj})$ to GWN_H			
GWN_H transmits (ID_{SNj}, ID_{GWNh}) to GWN_F			
$FHN_1 = h(SNX_j \ K_{FH}) \oplus SX_{GWNf}$ $FHN_2 = K_{FH} \oplus SNX_j$ $FHN_3 = ID_{SNj} \oplus ID_{GWNf}$ $FHN_4 = h(K_{FH} \ ID_{GWNh} \ ID_{GWNf} \ SNX_j \ ID_{SNj} \ SX_{GWNf})$ GWN_F transmits $(FHN_1, FHN_2, FHN_3, FHN_4)$ to the GWN_H			
$FHN_2 = K_{FH} \oplus SNX_j$ $GSC_5 = h(SNX_j \ ID_{SNj})$ $GSC_6 = UID_i \oplus SNX_j$ $GSC_7 = FHN_1 \oplus GSC_5$ GWN_H transmits (FHN_2, GSC_6, GSC_7) to U_i			
$SCF_5 = h(SNX_j \ K_{FH}) \oplus r_{ui}$ $SCF_6 = UID_i \oplus FHN_1$ $SCF_7 = h(r_{ui} \ UID_i \ T_{ui} \ SX_{GWNf} \ K_{FH} \ SNX_j)$ U_i delivers $(SCF_5, SCF_6, SCF_7, T_{ui})$ to GWN_F			
$FSN_1 = SNX_j \oplus UID_i$ $FSN_2 = ID_{SNj} \oplus h(UID_i \ r_{ui})$ $FSN_3 = r_{GWNf} \oplus UID_i \oplus h(r_{ui} \ UID_i)$ $FSN_4 = h(UID_i \ SNX_j \ h(r_{ui} \ UID_i) \ r_{GWNf} \ T_{gwnf})$ GWN_F transmits $(FSN_1, FSN_2, FSN_3, FSN_4, T_{gwnf})$ to SN_j			
$SFN_2 = r_{SNj} \oplus h(r_{GWNf} \ UID_i \ SNX_j)$ $SFN_3 = h(r_{SNj} \ r_{GWNf} \ SK_s \ UID_i \ SNX_j \ T_{snj})$ SN_j dispatches (SFN_2, SFN_3, T_{snj}) to GWN_F			
$FSC_1 = K_{FH} \oplus r_{GWNf}$ $FSC_2 = r_{SNj} \oplus h(SNX_j \ K_{FH} \ SX_{GWNf})$ $FSC_3 = GUID_i^{new} \oplus h(UID_i \ r_{ui})$ $FSC_4 = h(r_{SNj} \ GUID_i^{new} \ SNX_j \ K_{FH} \ SX_{GWNf} \ SK_f \ T_{gwnf1})$ GWN_F transmits $(FSC_1, FSC_2, FSC_3, FSC_4, T_{gwnf1})$ to U_i			

FIGURE 4: The main authentication steps in case 2.

A7: if it is not right, U_i suspends the session promptly. Otherwise, U_i counts $GF_i^{new} = GE_i \oplus GUID_i^{new} \oplus UID_i$ and substitutes $(GF_i^{new}, GUID_i^{new})$ for $(GF_i, GUID_i)$ in smart card.

Case 2:

A1: first, GWN_H broadcasts the information (ID_{SNj}, ID_{GWNh}) among all gateway nodes. GWN_F finds whether ID_{SNj} is in the information about the

sensor node it preserves. If it is in the information, GWN_F finds SNX_j based on ID_{SN_j} . Next, GWN_F finds and computes $FHN_1 = h(SNX_j \| K_{FH}) \oplus SX_{GWN_f}$, $FHN_2 = K_{FH} \oplus SNX_j$, $FHN_3 = ID_{SN_j} \oplus ID_{GWN_f}$, and $FHN_4 = h(K_{FH} \| ID_{GWN_f} \| ID_{GWN_f} \| SNX_j \| ID_{SN_j} \| SX_{GWN_f})$. Then, GWN_F transmits the information ($FHN_1, FHN_2, FHN_3, FHN_4$) to GWN_H .

A2: after reception of the information ($FHN_1, FHN_2, FHN_3, FHN_4$), GWN_H computes $ID_{GWN_f} = ID_{SN_j} \oplus FHN_3$, GWN_H finds the private key K_{FH} between them according to identity ID_{GWN_f} of GWN_F and computes $SNX_j^* = K_{FH} \oplus FHN_2$, $SX_{GWN_f}^* = h(SNX_j^* \| K_{FH}) \oplus FHN_1$, and $FHN_4^* = h(K_{FH} \| ID_{GWN_f} \| ID_{GWN_f} \| SNX_j^* \| ID_{SN_j} \| SX_{GWN_f}^*)$. Then, GWN_H confirms the correctness of the formula $FHN_4^* = FHN_4$.

A3: if it is not right, GWN_H terminates the session promptly. Otherwise, GWN_H computes $GSC_5 = h(SNX_j \| ID_{SN_j})$, $GSC_6 = UID_i \oplus SNX_j$, and $GSC_7 = FHN_1 \oplus GSC_5$ and transmits the information (FHN_2, GSC_6, GSC_7) to U_i .

A4: after reception of the information (FHN_2, GSC_6, GSC_7), U_i counts $SNX_j = UID_i \oplus GSC_6$, $K_{FH} = FHN_2 \oplus SNX_j$, $FHN_1 = GSC_7 \oplus h(ID_{SN_j} \| SNX_j)$, and $SX_{GWN_f} = h(SNX_j \| K_{FH}) \oplus FHN_1$. Then, U_i picks the stochastic digit r_{ui} and time stamp T_{ui} and counts $SCF_5 = h(SNX_j \| K_{FH}) \oplus r_{ui}$, $SCF_6 = UID_i \oplus FHN_1$, and $SCF_7 = h(r_{ui} \| UID_i \| T_{ui} \| SX_{GWN_f} \| K_{FH} \| SNX_j)$. Finally, U_i delivers the information ($SCF_5, SCF_6, SCF_7, T_{ui}$) to GWN_F .

A5: upon receiving the information ($SCF_5, SCF_6, SCF_7, T_{ui}$) at time T_{gwn_f} , GWN_F computes the freshness of the information by the formula $|T_{gwn_f} - T_{ui}| \leq \Delta T$. If it is not right, GWN_F terminates the session promptly. Otherwise, GWN_F computes $r_{ui}^* = h(SNX_j \| K_{FH}) \oplus SCF_5$, $UID_i^* = SCF_6 \oplus FHN_1$, and $SCF_7^* = h(r_{ui}^* \| UID_i^* \| T_{ui} \| SX_{GWN_f} \| K_{FH} \| SNX_j)$ and confirms the correctness of the formula $SCF_7^* = SCF_7$.

A6: if it is not right, GWN_F terminates the session promptly. Otherwise, GWN_F generates stochastic digit r_{GWN_f} and computes $FSN_1 = SNX_j \oplus UID_i$, $FSN_2 = ID_{SN_j} \oplus h(UID_i \| r_{ui})$, $FSN_3 = r_{GWN_f} \oplus UID_i \oplus h(r_{ui} \| UID_i)$, and $FSN_4 = h(UID_i \| SNX_j \| h(r_{ui} \| UID_i) \| r_{GWN_f} \| T_{gwn_f})$. Then, GWN_F transmits the information ($FSN_1, FSN_2, FSN_3, FSN_4, T_{gwn_f}$) to SN_j .

A7: upon receiving the information ($FSN_1, FSN_2, FSN_3, FSN_4, T_{gwn_f}$) at time T_{sn_j} , SN_j calculates the freshness of the information by the formula $|T_{sn_j} - T_{gwn_f}| \leq \Delta T$. If it is not right, SN_j ends the session promptly. Otherwise, SN_j calculates $UID_i^* = SNX_j \oplus FSN_1$, $h(r_{ui} \| UID_i)^* = ID_{SN_j} \oplus FSN_2$, $r_{GWN_f}^* = FSN_3 \oplus UID_i^* \oplus h(r_{ui} \| UID_i)^*$, and $FSN_4^* = h(UID_i^* \| SNX_j \| h(r_{ui} \| UID_i)^* \| r_{GWN_f}^* \| T_{gwn_f})$ and confirms the correctness of the formula $FSN_4^* = FSN_4$.

A8: if it is not right, SN_j ends the session promptly. Otherwise, SN_j selects stochastic digit r_{SN_j} and calculates $SK_s = h(r_{SN_j} \| r_{GWN_f} \| UID_i \| h(r_{ui} \| UID_i))$, $SFN_1 = h(UID_i \| r_{GWN_f} \| SNX_j)$, $SFN_2 = r_{SN_j} \oplus h(r_{GWN_f} \| UID_i \| SNX_j)$, and $SFN_3 = h(r_{SN_j} \| r_{GWN_f} \| SK_s \| UID_i \| SNX_j \| T_{sn_j})$. Then, SN_j dispatches the information (SFN_2, SFN_3, T_{sn_j}) to GWN_F .

A9: upon receiving the information (SFN_2, SFN_3, T_{sn_j}) at time T_{gwn_f1} , GWN_F computes the freshness of the information by the formula $|T_{gwn_f1} - T_{sn_j}| \leq \Delta T$. If it is not right, GWN_F terminates the session promptly. Otherwise, GWN_F computes $r_{SN_j}^* = SFN_2 \oplus h(r_{GWN_f} \| UID_i \| SNX_j)$, $SK_f = h(r_{SN_j}^* \| r_{GWN_f} \| UID_i \| h(r_{ui} \| UID_i))$, and $SFN_3^* = h(r_{SN_j}^* \| r_{GWN_f} \| SK_f \| UID_i \| SNX_j \| T_{sn_j})$ and confirms the correctness of the formula $SFN_3^* = SFN_3$.

A10: if it is not right, GWN_F terminates the session promptly. Otherwise, GWN_F generates new stochastic digit $r_{GWN_f}^{new}$ and computes $GUID_i^{new} = h(r_{GWN_f}^{new} \| SX_{GWN_f} \| ID_{GWN_f}) \oplus UID_i$, $FSC_1 = K_{FH} \oplus r_{GWN_f}$, $FSC_2 = r_{SN_j} \oplus h(SNX_j \| K_{FH} \| SX_{GWN_f})$, $FSC_3 = GUID_i^{new} \oplus h(UID_i \| r_{ui})$, and $FSC_4 = h(r_{SN_j} \| GUID_i^{new} \| SNX_j \| K_{FH} \| SX_{GWN_f} \| SK_f \| T_{gwn_f1})$. Then, GWN_F transmits the information ($FSC_1, FSC_2, FSC_3, FSC_4, T_{gwn_f1}$) to U_i .

A11: upon receiving the information ($FSC_1, FSC_2, FSC_3, FSC_4, T_{gwn_f1}$) at time T_{ui} , U_i computes the freshness of the information by the formula $|T_{ui} - T_{gwn_f1}| \leq \Delta T$. If it is not right, U_i suspends the session promptly. Otherwise, U_i counts $r_{GWN_f}^* = K_{FH} \oplus FSC_1$, $r_{SN_j}^* = FSC_2 \oplus h(SNX_j \| K_{FH} \| SX_{GWN_f})$, $GUID_i^{new} = FSC_3 \oplus h(UID_i \| r_{ui})$, $SK_u = h(r_{SN_j}^* \| r_{GWN_f}^* \| UID_i \| h(r_{ui} \| UID_i))$, $r_{GWN_h}^* = SNX_j^* \oplus HSN_3$, and $FSC_4 = h(r_{SN_j}^* \| GUID_i^{new} \| SNX_j \| K_{FH} \| SX_{GWN_f} \| SK_u \| T_{gwn_f1})$ and confirms the correctness of the formula $FSC_4^* = FSC_4$.

A12: if it is not right, U_i suspends the session promptly. Otherwise, U_i counts $GF_i^{new} = GE_i \oplus GUID_i^{new} \oplus UID_i$ and substitutes ($GF_i^{new}, GUID_i^{new}$) for ($GF_i, GUID_i$) in smart card.

3.5. Password and Biometric Change Section

A1: U_i inserts smart card and inputs his or her identity ID_i , password PW_i , and biometric BIO_{U_i} .

A2: smart card counts $REP(BIO_{U_i}, \tau_{U_i}) = \sigma_{U_i}$, r_i^* , r_p^* , UID_i^* , UPW_i^* , and USC_3^* and confirms the correctness of the formula $USC_3^* = USC_3$.

A3: if it is not right, smart card suspends the session promptly. Otherwise, U_i picks the new parameters,

identity ID_i , password PW_i^{new} , and two stochastic digits r_i and r_p , and counts $UID_i = h(ID_i || r_i)$ and $UPW_i^{new} = h(PW_i || r_i || r_p)$. After the calculation, U_i delivers UID_i and UPW_i^{new} to HGWN as the change request.

A4: after getting the change request, HGWN generates a stochastic digit r_{GWNh} and computes $GUID_i^{new}$, GE_i^{new} , and GF_i^{new} in combination with its own privacy parameters. HGWN loads GE_i^{new} and

GF_i^{new} into the smart card and transmits the smart card to U_i .

A5: after reception of the smart card, U_i imprints his or her unique biometric $BIO_{U_i}^{new}$ on the sensor device specific terminal and further counts $GEN(BIO_{U_i}^{new}) = (\sigma_{U_i}^{new}, \tau_{U_i}^{new})$, USC_1^{new} , USC_2^{new} , and USC_3^{new} . Then, U_i loads $(USC_1^{new}, USC_2^{new}, USC_3^{new})$ into the smart card to replace the old parameters.

4. Formal Security Analysis of Protocol

In this section, we separately apply BAN logic and AVISPA tool to validate the security of our proposed authentication and key agreement protocol in case 1 and case 2.

4.1. BAN Logic (Case 1). In this section, we will validate our proposed designed authentication protocol by applying the BAN logic in case 1.

BAN logic notations are as follows:

- (1) $\partial | \equiv \beta$: ∂ trusts the realness in β
- (2) $\partial \triangleleft \beta$: ∂ obtains or sees information β
- (3) $\partial \sim \beta$: ∂ sent or said information β
- (4) $\partial \Rightarrow \beta$: ∂ has jurisdiction over β
- (5) $\#(\beta)$: β is fresh
- (6) $\partial \stackrel{SK}{\longleftrightarrow} \beta$: SK is the private session key between ∂ and β
- (7) $(\beta)_{SK}$: β is encrypted with the private session key SK

BAN logic postulate rules:

PR1: Message-meaning rule: $(\partial | \equiv \beta \stackrel{SK}{\longleftrightarrow} \partial, \partial \triangleleft \{M\}_k) / \partial | \equiv \beta | \sim M$

PR2: Nonce-verification rule: $(\partial | \equiv \#(M), \partial | \equiv \beta | \sim M) / \partial | \equiv \beta | \equiv M$

PR3: Jurisdiction rule: $(\partial | \equiv \beta | \equiv M, \partial | \equiv \beta | \Rightarrow M) / \partial | \equiv M$

PR4: Fresh rule: $(\partial | \equiv \#(M)) / \partial | \equiv \#(M, P)$

PR5: Belief rule: $(\partial | \equiv \beta | \equiv (M, P)) / \partial | \equiv \beta | \equiv M$

Security goals are as follows:

- Goal 1: $U_i | \equiv (U_i \stackrel{SK}{\longleftrightarrow} GWN_h)$
 Goal 2: $U_i | \equiv GWN_h | \equiv (U_i \stackrel{SK}{\longleftrightarrow} GWN_h)$
 Goal 3: $GWN_h | \equiv (U_i \stackrel{SK}{\longleftrightarrow} GWN_h)$
 Goal 4: $GWN_h | \equiv U_i | \equiv (U_i \stackrel{SK}{\longleftrightarrow} GWN_h)$
 Goal 5: $SN_j | \equiv (SN_j \stackrel{SK}{\longleftrightarrow} GWN_h)$
 Goal 6: $SN_j | \equiv GWN_h | \equiv (SN_j \stackrel{SK}{\longleftrightarrow} GWN_h)$

Goal 7: $GWN_h | \equiv (SN_j \stackrel{SK}{\longleftrightarrow} GWN_h)$

Goal 8: $GWN_h | \equiv SN_j | \equiv (SN_j \stackrel{SK}{\longleftrightarrow} GWN_h)$

Rational assumptions are as follows:

RA1: $GWN_h | \equiv (U_i \stackrel{GUID_i}{\longleftrightarrow} GWN_h)$

RA2: $GWN_h | \equiv (\#T_{sc})_{SNX_j}$

RA3: $SN_j | \equiv (GWN_h \longleftrightarrow SN_j)$

RA4: $SN_j | \equiv (\#SNX_j)$

RA5: $GWN_h | \equiv (SN_j \stackrel{r_{SNj}}{\longleftrightarrow} GWN_h)$

RA6: $GWN_h | \equiv (\#T_{snj})$

RA7: $GWN_h | \equiv SN_j | \Rightarrow (SN_j \stackrel{SK}{\longleftrightarrow} GWN_h)$

RA8: $U_i | \equiv (GWN_h \longleftrightarrow U_i)$

RA9: $U_i | \equiv (\#T_{hgwm1})$

RA10: $U_i | \equiv GWN_h | \Rightarrow (U_i \stackrel{SK}{\longleftrightarrow} GWN_h)$

RA11: $SN_j | \equiv GWN_h | \Rightarrow (SN_j \stackrel{SK}{\longleftrightarrow} GWN_h)$

RA12: $GWN_h | \equiv U_i | \Rightarrow (U_i \stackrel{SK}{\longleftrightarrow} GWN_h)$

The idealized form of the information is as follows:

Inf 1: $U_i \longrightarrow GWN_h$ ($SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SNj}$)

Inf 2: $GWN_h \longrightarrow SN_j$ ($HSN_1, HSN_2, HSN_3, HSN_4, T_{gwnh}$)

Inf3: $SN_j \longrightarrow GWN_h$ (SHN_1, SHN_2, T_{snj})

Inf4: $GWN_h \longrightarrow U_i$ ($HSN_3, GSC_1, GSC_2, GSC_3, GSC_4, T_{hgwm1}$)

In view of Inf1, we are ready to receive the following:

F1: $GWN_h \triangleleft (SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SNj})_{GUID}$

In view of F1, RA1, and PR1, we are ready to receive the following:

F2: $GWN_h | \equiv U_i | \sim (SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SNj})$

The equivalent form of F2 is the following:

F3: $GWN_h | \equiv U_i | \sim (UID_i, SCN_i, r_{SCn}, T_{sc})$

In view of F3, RA2, PR4, and PR2, we are ready to receive the following:

F4: $GWN_h | \equiv U_i | \equiv (UID_i, SCN_i, r_{SCn}, T_{sc})$

In view of F4 and PR5, we are ready to receive the following:

F5: $GWN_h | \equiv U_i | \equiv (UID_i, SCN_i, r_{SCn})$

In view of Inf2, we are ready to receive the following:

F6: $SN_j \triangleleft (HSN_1, HSN_2, HSN_3, HSN_4, T_{gwnh})_{SNX_j}$

In view of F6, RA3, and PR1, we are ready to receive the following:

F7: $SN_j | \equiv GWN_h | \sim (HSN_1, HSN_2, HSN_3, HSN_4, T_{gwnh})$

The equivalent form of F7 is the following:

F8: $SN_j | \equiv GWN_h | \sim (SNX_j, UID_i, r_{SCn}, SCN_i, r_{GWNh}, T_{gwnh})$

In view of F8, RA4, PR4, and PR2, we are ready to receive the following:

F9: $SN_j | \equiv GWN_h | \equiv (SNX_j, UID_i, r_{SCn}, SCN_i, r_{GWNh}, T_{gwnh})$

In view of F9 and PR5, we are ready to receive the following:

F10: $SN_j | \equiv GWN_h | \equiv (SNX_j, UID_i, r_{SCn}, SCN_i, r_{GWNh})$

In view of Inf3, we are ready to receive the following:

F11: $GWN_h \triangleleft (SHN_1, SHN_2, T_{snj})_{r_{snj}}$

In view of F11, RA5, and PR1, we are ready to receive the following:

F12: $GWN_h | \equiv SN_j | \sim (SHN_1, SHN_2, T_{snj})$

The equivalent form of F12 is the following:

F13: $GWN_h | \equiv SN_j | \sim (r_{SNj}, r_{GWNh}, r_{SCn}, SCN_i, T_{snj}, UID_i, SNX_j)$

In view of F13, RA6, PR4, and PR2, we are ready to receive the following:

F14: $GWN_h | \equiv SN_j | \equiv (r_{SNj}, r_{GWNh}, r_{SCn}, SCN_i, T_{snj}, UID_i, SNX_j)$

In view of F14 and PR5, we are ready to receive the following:

F15: $GWN_h | \equiv SN_j | \equiv (r_{SNj}, r_{GWNh}, r_{SCn}, SCN_i, UID_i, SNX_j)$

The private session key is $SK = h(r_{SNj} || r_{GWNh} || UID_i || h(r_{SCn} || SCN_i) || SNX_j)$

In view of F15, we are ready to receive the following:

F16: $GWN_h | \equiv SN_j | \equiv (SN_j \xleftrightarrow{SK} GWN_h)$ Goal 8

In view of F16, RA7, and PR3, we are ready to receive the following:

F17: $GWN_h | \equiv (SN_j \xleftrightarrow{SK} GWN_h)$ Goal 7

In view of Inf4, we are ready to receive the following:

F18:

$U_i \triangleleft (HSN_3, GSC_1, GSC_2, GSC_3, GSC_4, T_{hgwm1})_{GF_i}$

In view of F18, RA8, and PR1, we are ready to receive the following:

F19: $U_i | \equiv GWN_h | \sim (HSN_3, GSC_1, GSC_2, GSC_3, GSC_4, T_{hgwm1})$

The equivalent form of F19 is the following:

F20: $U_i | \equiv GWN_h | \sim (SNX_j, UID_i, r_{GWNh}, r_{SNj}, r_{SCn}, SCN_i, T_{hgwm1})$

In view of F20, RA9, PR4, and PR2, we are ready to receive the following:

F20: $U_i | \equiv GWN_h | \equiv (SNX_j, UID_i, r_{GWNh}, r_{SNj}, r_{SCn}, SCN_i, T_{hgwm1})$

In view of F20 and PR5, we are ready to receive the following:

F21: $U_i | \equiv GWN_h | \equiv (SNX_j, UID_i, r_{GWNh}, r_{SNj}, r_{SCn}, SCN_i)$

The private session key is $SK = h(r_{SNj} || r_{GWNh} || UID_i || h(r_{SCn} || SCN_i) || SNX_j)$

In view of F21, we are ready to receive the following:

F22: $U_i | \equiv GWN_h | \equiv (U_i \xleftrightarrow{SK} GWN_h)$ Goal 2

In view of F22, RA10, and PR3, we are ready to receive the following:

F23: $U_i | \equiv (U_i \xleftrightarrow{SK} GWN_h)$ Goal 1

The private session key is $SK = h(r_{SNj} || r_{GWNh} || UID_i || h(r_{SCn} || SCN_i) || SNX_j)$

In view of F10 and F15, we are ready to receive the following:

F24: $SN_j | \equiv GWN_h | \equiv (SN_j \xleftrightarrow{SK} GWN_h)$ Goal 6

In view of F24, RA11, and PR3, we are ready to receive the following:

F25: $SN_j | \equiv (SN_j \xleftrightarrow{SK} GWN_h)$ Goal 5

The private session key is $SK = h(r_{SNj} || r_{GWNh} || UID_i || h(r_{SCn} || SCN_i) || SNX_j)$

In view of F5 and F21, we are ready to receive the following:

F26: $GWN_h | \equiv U_i | \equiv (U_i \xleftrightarrow{SK} GWN_h)$ Goal 4

In view of F26, RA12, and PR3, we are ready to receive the following:

F27: $GWN_h | \equiv (U_i \xleftrightarrow{SK} GWN_h)$ Goal 3

4.2. *AVISPA Tool (Case 1)*. In this section, we will validate our proposed designed authentication protocol by applying the AVISPA tool in case 1. In AVISPA tool, four validation models are supported: OFMC, ATSE, SATMC, and TA4SP. The security of our designed authentication protocol is simulated by applying the HLPSSL (High Level Protocol Specifications Language). Figures 5 and 6 present the result of the simulation by applying the OFMC and ATSE.

4.3. *BAN Logic (Case 2)*. In this section, we will validate our proposed designed authentication protocol by applying the BAN logic in case 2.

Goal 1: $U_i | \equiv (U_i \xleftrightarrow{SK} GWN_f)$

Goal 2: $U_i | \equiv GWN_f | \equiv (U_i \xleftrightarrow{SK} GWN_f)$

Goal 3: $GWN_f | \equiv (U_i \xleftrightarrow{SK} GWN_f)$

Goal 4: $GWN_f | \equiv U_i | \equiv (U_i \xleftrightarrow{SK} GWN_f)$

Goal 5: $SN_j | \equiv (SN_j \xleftrightarrow{SK} GWN_f)$

Goal 6: $SN_j | \equiv GWN_f | \equiv (SN_j \xleftrightarrow{SK} GWN_f)$

Goal 7: $GWN_f | \equiv (SN_j \xleftrightarrow{SK} GWN_f)$

Goal 8: $GWN_f | \equiv SN_j | \equiv (SN_j \xleftrightarrow{SK} GWN_f)$

Rational assumptions are as follows:

RA1: $GWN_f | \equiv (U_i \xleftrightarrow{UID_i} GWN_f)$

RA2: $GWN_f | \equiv (\#T_{ui})$

RA3: $SN_j | \equiv (GWN_f \xleftrightarrow{r_{GWNf}} SN_j)$

RA4: $SN_j | \equiv (\#T_{gwnf})$

RA5: $GWN_f | \equiv (SN_j \xleftrightarrow{FSN_1} GWN_f)$

RA6: $GWN_f | \equiv (\#T_{snj})$

RA7: $GWN_f | \equiv SN_j | \Rightarrow (SN_j \xleftrightarrow{SK} GWN_f)$

RA8: $U_i | \equiv (GWN_f \xleftrightarrow{FSC_1} U_i)$

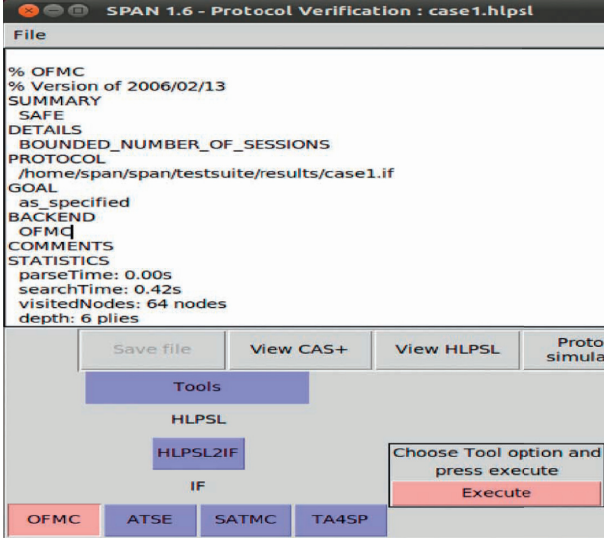


FIGURE 5: The simulation result of OFMC.

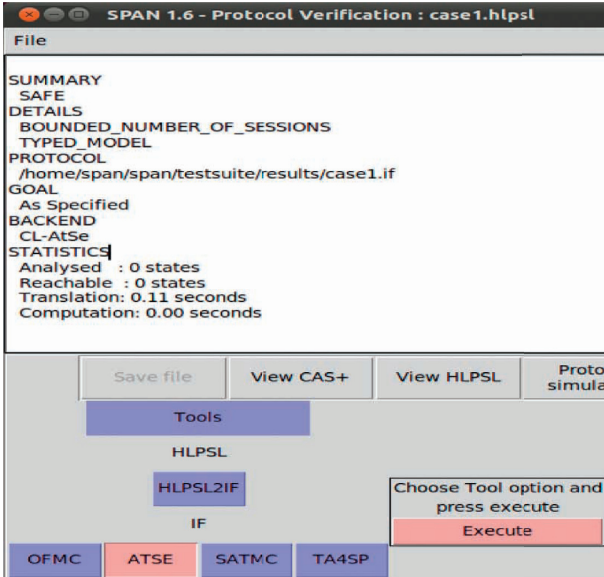


FIGURE 6: The simulation result of ATSE.

$$\text{RA9: } U_i | \equiv (\#T_{\text{gwnf}1})$$

$$\text{RA10: } U_i | \equiv \text{GWN}_f | \Rightarrow (U_i \xleftrightarrow{\text{SK}} \text{GWN}_f |)$$

$$\text{RA11: } \text{SN}_j | \equiv \text{GWN}_f | \Rightarrow (\text{SN}_j | \xleftrightarrow{\text{SK}} \text{GWN}_f |)$$

$$\text{RA12: } \text{GWN}_f | \equiv U_i | \Rightarrow (U_i \xleftrightarrow{\text{SK}} \text{GWN}_f |)$$

The idealized form of the information is as follows:

$$\text{Inf1: } U_i \longrightarrow \text{GWN}_f | (\text{SCF}_5, \text{SCF}_6, \text{SCF}_7, T_{ui})$$

$$\text{Inf2: } \text{GWN}_f \longrightarrow \text{SN}_j | (\text{FSN}_1, \text{FSN}_2, \text{FSN}_3, \text{FSN}_4, T_{\text{gwnf}})$$

$$\text{Inf3: } \text{SN}_j \longrightarrow \text{GWN}_f | (\text{SFN}_2, \text{SFN}_3, T_{\text{snj}})$$

$$\text{Inf4: } \text{GWN}_f \longrightarrow U_i | (\text{FSC}_1, \text{FSC}_2, \text{FSC}_3, \text{FSC}_4, T_{\text{gwnf}1})$$

In view of Inf1, we are ready to receive the following:

$$\text{F1: } \text{GWN}_f | \triangleleft (\text{SCF}_5, \text{SCF}_6, \text{SCF}_7, T_{ui})_{\text{UID}_i}$$

In view of F1, RA1, and PR1, we are ready to receive the following:

$$\text{F2: } \text{GWN}_f | \equiv U_i | \sim (\text{SCF}_5, \text{SCF}_6, \text{SCF}_7, T_{ui})$$

The equivalent form of F2 is the following:

$$\text{F3: } \text{GWN}_f | \equiv U_i | \sim (\text{SNX}_j, K_{\text{FH}}, r_{ui}, \text{UID}_i, \text{ID}_{\text{SN}_j}, \text{SX}_{\text{GWN}_f}, T_{ui})$$

In view of F3, RA2, PR4, and PR2, we are ready to receive the following:

$$\text{F4: } \text{GWN}_f | \equiv U_i | \equiv (\text{SNX}_j, K_{\text{FH}}, r_{ui}, \text{UID}_i, \text{ID}_{\text{SN}_j}, \text{SX}_{\text{GWN}_f}, T_{ui})$$

In view of F4 and PR5, we are ready to receive the following:

$$\text{F5: } \text{GWN}_f | \equiv U_i | \equiv (r_{ui}, \text{UID}_i)$$

In view of Inf2, we are ready to receive the following:

$$\text{F6: } \text{SN}_j | \triangleleft (\text{FSN}_1, \text{FSN}_2, \text{FSN}_3, \text{FSN}_4, T_{\text{gwnf}})_{r_{\text{GWN}_f}}$$

In view of F6, RA3, and PR1, we are ready to receive the following:

$$\text{F7: } \text{SN}_j | \equiv \text{GWN}_f | \sim (\text{FSN}_1, \text{FSN}_2, \text{FSN}_3, \text{FSN}_4, T_{\text{gwnf}})$$

The equivalent form of F7 is the following:

$$\text{F8: } \text{SN}_j | \equiv \text{GWN}_f | \sim (\text{SNX}_j, \text{UID}_i, \text{ID}_{\text{SN}_j}, r_{ui}, r_{\text{GWN}_f}, T_{\text{gwnf}})$$

In view of F8, RA4, PR4, and PR2, we are ready to receive the following:

$$\text{F9: } \text{SN}_j | \equiv \text{GWN}_f | \equiv (\text{SNX}_j, \text{UID}_i, \text{ID}_{\text{SN}_j}, r_{ui}, r_{\text{GWN}_f}, T_{\text{gwnf}})$$

In view of F9 and PR5 we are ready to receive the following:

$$\text{F10: } \text{SN}_j | \equiv \text{GWN}_f | \equiv (\text{UID}_i, r_{ui}, r_{\text{GWN}_f})$$

In view of Inf3, we are ready to receive the following:

$$\text{F11: } \text{GWN}_f | \triangleleft (\text{SFN}_2, \text{SFN}_3, T_{\text{snj}})_{\text{SFN}_1}$$

In view of F11, RA5, and PR1, we are ready to receive the following:

$$\text{F12: } \text{GWN}_f | \equiv \text{SN}_j | \sim (\text{SFN}_2, \text{SFN}_3, T_{\text{snj}})$$

The equivalent form of F12 is the following:

$$\text{F13: } \text{GWN}_f | \equiv \text{SN}_j | \sim (r_{\text{SN}_j}, r_{\text{GWN}_f}, r_{ui}, \text{UID}_i, \text{SNX}_j, T_{\text{snj}})$$

In view of F13, RA6, PR4, and PR2, we are ready to receive the following:

$$\text{F14: } \text{GWN}_f | \equiv \text{SN}_j | \equiv (r_{\text{SN}_j}, r_{\text{GWN}_f}, r_{ui}, \text{UID}_i, \text{SNX}_j, T_{\text{snj}})$$

In view of F14 and PR5, we are ready to receive the following:

$$\text{F15: } \text{GWN}_f | \equiv \text{SN}_j | \equiv (r_{\text{SN}_j}, r_{\text{GWN}_f}, r_{ui}, \text{UID}_i)$$

The private session key is $\text{SK} = h(r_{\text{SN}_j} \| r_{\text{GWN}_f} \| \text{UID}_i \| h(r_{ui} \| \text{UID}_i))$

In view of F15, we are ready to receive the following:

$$\text{F16: } \text{GWN}_f | \equiv \text{SN}_j | \equiv (\text{SN}_j \xleftrightarrow{\text{SK}} \text{GWN}_f) \text{ Goal 8}$$

In view of F16, RA7, and PR3, we are ready to receive the following:

$$\text{F17: } \text{GWN}_f | \equiv (\text{SN}_j \xleftrightarrow{\text{SK}} \text{GWN}_f) \text{ Goal 7}$$

In view of Inf4, we are ready to receive the following:

$$\text{F18: } U_i | \triangleleft (\text{FSC}_1, \text{FSC}_2, \text{FSC}_3, \text{FSC}_4, T_{\text{gwnf}1})_{\text{FSC}_1}$$

In view of F18, RA8, and PR1, we are ready to receive the following:

$$F19: U_i | \equiv \text{GWN}_f | \sim (FSC_1, FSC_2, FSC_3, FSC_4, T_{\text{gwn}f1})$$

The equivalent form of F19 is the following:

$$F20: U_i | \equiv \text{GWN}_f | \sim (K_{\text{FH}}, r_{\text{GWN}_f}, r_{\text{SN}_j}, \text{SNX}_j, \text{UID}_i, r_{ui}, \text{SX}_{\text{GWN}_f}, T_{\text{gwn}f1})$$

In view of F20, RA9, PR4, and PR2, we are ready to receive the following:

$$F20: U_i | \equiv \text{GWN}_f | \equiv (K_{\text{FH}}, r_{\text{GWN}_f}, r_{\text{SN}_j}, \text{SNX}_j, \text{UID}_i, r_{ui}, \text{SX}_{\text{GWN}_f}, T_{\text{gwn}f1})$$

In view of F20 and PR5, we are ready to receive the following:

$$F21: U_i | \equiv \text{GWN}_f | \equiv (r_{\text{GWN}_f}, r_{\text{SN}_j}, \text{UID}_i, r_{ui})$$

The private session key is $\text{SK} = h(r_{\text{SN}_j} \| r_{\text{GWN}_f} \| \text{UID}_i \| h(r_{ui} \| \text{UID}_i))$

In view of F21, we are ready to receive the following:

$$F22: U_i | \equiv \text{GWN}_f | \equiv (U_i \xleftrightarrow{\text{SK}} \text{GWN}_f) \text{ Goal 2}$$

In view of F22, RA10, and PR3, we are ready to receive the following:

$$F23: U_i | \equiv (U_i \xleftrightarrow{\text{SK}} \text{GWN}_f) \text{ Goal 1}$$

The private session key is $\text{SK} = h(r_{\text{SN}_j} \| r_{\text{GWN}_f} \| \text{UID}_i \| h(r_{ui} \| \text{UID}_i))$

In view of F10 and F15, we are ready to receive the following:

$$F24: \text{SN}_j | \equiv \text{GWN}_f | \equiv (\text{SN}_j \xleftrightarrow{\text{SK}} \text{GWN}_f) \text{ Goal 6}$$

In view of F24, RA11, and PR3, we are ready to receive the following:

$$F25: \text{SN}_j | \equiv (\text{SN}_j \xleftrightarrow{\text{SK}} \text{GWN}_f) \text{ Goal 5}$$

The private session key is $\text{SK} = h(r_{\text{SN}_j} \| r_{\text{GWN}_f} \| \text{UID}_i \| h(r_{ui} \| \text{UID}_i))$

In view of F5 and F21, we are ready to receive the following:

$$F26: \text{GWN}_f | \equiv U_i | \equiv (U_i \xleftrightarrow{\text{SK}} \text{GWN}_f) \text{ Goal 4}$$

In view of F26, RA12, and PR3, we are ready to receive the following:

$$F27: \text{GWN}_f | \equiv (U_i \xleftrightarrow{\text{SK}} \text{GWN}_f) \text{ Goal 3}$$

4.4. AVISPA Tool (Case 2). In this section, we will validate our proposed designed authentication protocol by applying the AVISPA tool in case 2. Figures 7 and 8 present the result of the simulation by applying the ATSE and OFMC.

5. Informal Security Analysis of Protocol

In this section, we demonstrate informal security analysis of our proposed mutual authentication protocol through sixteen evaluation criteria as defined in Section 2.3.

5.1. Session Key Security. In our designed protocol, the private session key is derived from the relevant privacy parameters of the three parties involved in the communication process through hash function operation. In case 1,

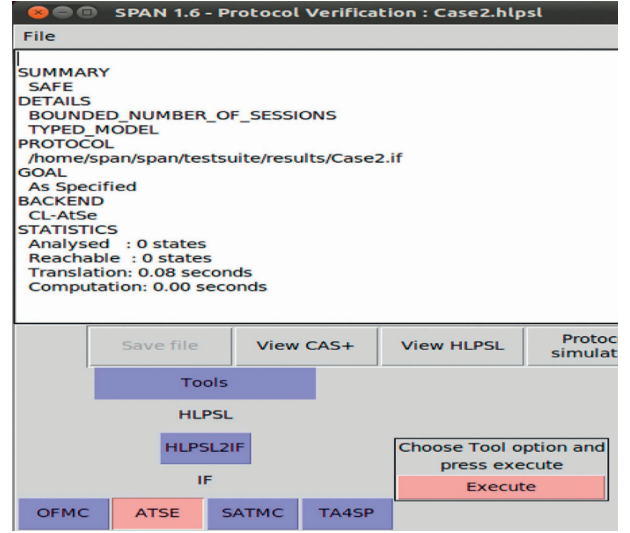


FIGURE 7: The simulation result of ATSE.

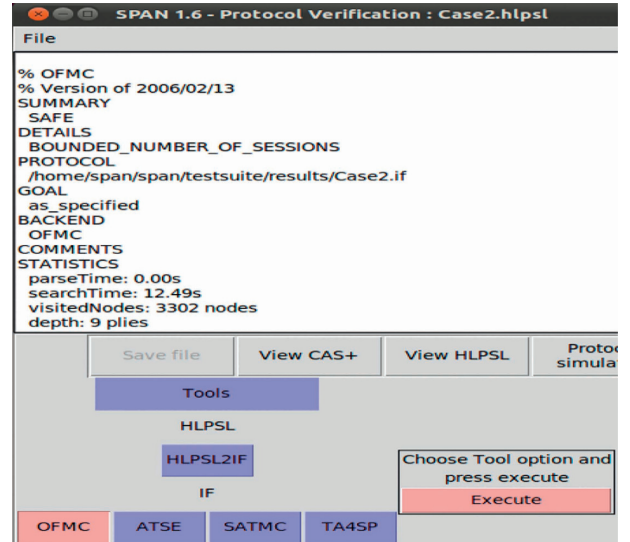


FIGURE 8: The simulation result of OFMC.

the private session key is $\text{SK} = h(r_{\text{SN}_j} \| r_{\text{GWN}_f} \| \text{UID}_i \| h(r_{\text{SC}_n} \| \text{SCN}_i) \| \text{SNX}_j)$. The information $(\text{HSN}_3, \text{GSC}_1, \text{GSC}_2, \text{GSC}_3, \text{GSC}_4, T_{\text{hgwn}1})$ transmitted from GWN_H to U_i comprises the session key; that is, $\text{GSC}_4 = h(r_{\text{SC}_n} \| \text{GUID}_i^{\text{new}} \| \text{SNX}_j \| \text{SK}_h \| \text{GF}_i \| T_{\text{gwn}h1})$. Let us assume that the assailant captures the information; then the assailant intends to figure out $\text{SK}^* = h(r_{\text{SN}_j} \| r_{\text{GWN}_f} \| \text{UID}_i \| h(r_{\text{SC}_n} \| \text{SCN}_i) \| \text{SNX}_j)$ by creating r_{SN_j} , r_{GWN_f} , r_{SC_n} , $\text{SNX}_j = h(\text{ID}_{\text{SN}_j} \| \text{SX}_{\text{SN}_j})$, $h(r_{\text{SC}_n} \| \text{SCN}_i)$, and $\text{UID}_i = h(\text{ID}_i \| r_i)$. In case 2, the private session key is $\text{SK}_s = h(r_{\text{SN}_j} \| r_{\text{GWN}_f} \| \text{UID}_i \| h(r_{ui} \| \text{UID}_i))$. The information $(\text{SFN}_2, \text{SFN}_3, T_{\text{snj}})$ dispatched from SN_j to GWN_F includes the session key; that is, $\text{SFN}_3 = h(r_{\text{SN}_j} \| r_{\text{GWN}_f} \| \text{SK}_s \| \text{UID}_i \| \text{SNX}_j \| T_{\text{snj}})$. By creating

r_{SNj} , r_{GWNh} , $UID_i = h(ID_i \| r_i)$, and $h(r_{Ui} \| UID_i)$, the assailant is able to figure out the session key $SK^* = h(r_{SNj} \| r_{GWNf} \| UID_i \| h(r_{ui} \| UID_i))$. Nevertheless, it is impracticable for the assailant to figure out the session key without knowing these privacy parameters and finishing inversion of hash function in polynomial time. Thus, our designed protocol is capable of achieving session key security.

5.2. Three-Factor Security. In our designed protocol, if the assailant only knows two of three factors, he is unable to launch an attack in our designed protocol. The first possibility is that the assailant only knows smart card and biometric. In this condition, assume that the assailant captures $(GE_i, GF_i, USC_1, USC_2, USC_3)$ kept in smart card and regains σ_{Ui} by the formula $GEN(BIO_{Ui}) = (\sigma_{Ui}, \tau_{Ui})$. Later, the assailant will speculate ID_i , PW_i , r_i , and r_p to figure out $UID_i^* = h(ID_i^* \| r_i^*)$, $UPW_i^* = h(PW_i^* \| r_i^* \| r_p^*)$, and $USC_3 = h(UID_i \| UPW_i \| \sigma_{Ui} \| r_i \| r_p)$ and confirms the correctness of the formula $USC_3^* = USC_3$. Nevertheless, the assailant cannot obtain password and sensitive parameters at the same time [4]. The smart card will suspend the session promptly after the assailant inputs the speculated password and sensitive parameters. The second possibility is that the assailant only knows password and biometric. Although the assailant has no ability to regain σ_{Ui} by the formula $REP(BIO_{Ui}, \tau_{Ui}) = \sigma_{Ui}$, he is able to capture the communication information $(SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SNj})$. Even if the assailant obtains the correct password and biometric, he still cannot pass the verification of the smart card and cannot simulate the communication information. The third possibility is that the assailant only knows smart card and password. Assume that the assailant captures $(GE_i, GF_i, USC_1, USC_2, USC_3)$ kept in smart card, where $USC_1 = r_i \oplus h(ID_i \| PW_i \| \sigma_{Ui})$, $USC_2 = r_p \oplus h(\sigma_{Ui} \| r_i)$, and $USC_3 = h(UID_i \| UPW_i \| \sigma_{Ui} \| r_i \| r_p)$. Due to the uniqueness of biometric, the assailant has no ability to regain σ_{Ui} by the formula $GEN(BIO_{Ui}) = (\sigma_{Ui}, \tau_{Ui})$. Without obtaining accurate biometric information to figure out USC_1 , USC_2 , and USC_3 , it is impossible for the assailant to imitate user to log into the gateway.

5.3. Perfect Forward and Backward Security. In our designed protocol, the private session key in case 1 is $SK = h(r_{SNj} \| r_{GWNh} \| UID_i \| h(r_{SCn} \| SCN_i) SNX_j)$ and it is counted by the stochastic digits r_{SNj} , r_{GWNh} , r_i , and r_{SCn} , the identities ID_i , SCN_i , and ID_{SNj} , and the private key SX_{SNj} . The private session key in case 2 is $SK = h(r_{SNj} \| r_{GWNf} \| UID_i \| h(r_{ui} \| UID_i))$ and it is counted by the stochastic digits r_{SNj} , r_{GWNh} , r_{ui} , and r_i and the identity ID_i . The private session key is counted by the hash function and the stochastic digits are variable in each session. Even if the assailant compromises the private session key SK in case 1 and case 2, he is unable to obtain any previous or future private session keys. Consequently, our designed protocol is capable of achieving perfect forward and backward security.

5.4. Resist Sensor Node Capture Attack. In our designed protocol, the assailant is able to capture the sensor node and obtain the information (ID_{SNj}, SNX_j) kept in the sensor nodes, since the sensor nodes are placed in an unattended environment. SNX_j is calculated as $SNX_j = h(ID_{SNj} \| SX_{SNj})$ and SX_{SNj} is the private key of sensor node that is only known to himself. Even if the assailant compromises the information kept in the sensor nodes, he is unable to accurately figure out the private parameters in sensor nodes and create the effective information in the communication process. Consequently, our designed protocol is capable of resisting sensor node capture attack.

5.5. Resist Stolen Smart Card Attack. In our designed protocol, smart card is one of the three factors; hence, the case where the smart card is stolen is supposed to be taken into consideration. Smart card includes GE_i , GF_i , USC_1 , USC_2 , and USC_3 , where $GE_i = h(UID_i \| UPW_i)$, $GF_i = GE_i \oplus GUID_i \oplus UID_i$, $USC_1 = r_i \oplus h(ID_i \| PW_i \| \sigma_{Ui})$, $USC_2 = r_p \oplus h(\sigma_{Ui} \| r_i)$, and $USC_3 = h(UID_i \| UPW_i \| \sigma_{Ui} \| r_i \| r_p)$; r_i and r_p are stochastic digits picked by U_i ; and σ_{Ui} is counted by GEN . Assume that the smart card is stolen by the assailant through power analysis method and the information $(GE_i, GF_i, USC_1, USC_2, USC_3)$ kept in smart card is available to the assailant. The assailant is unable to speculate ID_i , PW_i , and σ_{Ui} through USC_1 and is also unable to speculate r_{GWNh} and SX_{GWNh} through $GUID_i$. Without these important parameters, the assailant is unable to imitate the smart card information. Thus, our designed protocol is capable of resisting stolen smart card attack.

5.6. Resist User Impersonation Attack. In our designed protocol, assume that the login request information $(SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SNj})$ is known by the assailant. In order to compute SCG_1 , the assailant has to calculate $GUID_i$ and SCN_i . In order to compute SCG_2 , the assailant has to calculate r_{SCn} and $h(SCN_i \| T_{sc})$. In order to compute SCG_4 , the assailant has to calculate $GUID_i$, SCG_3 , r_{SCn} , and SCN_i . To implement impersonation attack, the assailant has to speculate accurate parameters $(r_{SCn}, SCN_i, T_{sc}, r_{GWNh}, SX_{GWNh}, ID_{GWNh}, ID_i, PW_i, r_i, r_p)$. However, it is impossible for the assailant to gain these parameters. Without these important parameters, the assailant is unable to imitate the user to participate in the communication process. Thus, our designed protocol is capable of resisting user impersonation attack.

5.7. Resist Gateway Impersonation Attack. In our designed protocol, when U_i delivers the registration request (UID_i, UPW_i) to GWN_H , where $UID_i = h(ID_i \| r_i)$ and $UPW_i = h(PW_i \| r_i \| r_p)$, the assailant is able to capture this registration information and demands to reply information (GE_i^*, GF_i^*) to U_i , where $GF_i^* = GE_i^* \oplus GUID_i^* \oplus UID_i^*$, $GE_i^* = h(UID_i^* \| UPW_i^*)$, and $GUID_i^* = h(r_{GWNh} \| SX_{GWNh} \| ID_{GWNh}^*) \oplus UID_i^*$. In order to accurately calculate these parameters, the assailant needs to speculate $(r_{GWNh}, r_i, r_p, ID_i, PW_i, SX_{GWNh}, ID_{GWNh})$. As the stochastic

digits (r_{GWNh}, r_i, r_p) are variable in each session, this reply will not be successful. Consequently, our designed protocol is capable of resisting gateway impersonation attack.

5.8. Resist Sensor Node Impersonation Attack. In our designed protocol, the assailant is able to capture the information $(HSN_1, HSN_2, HSN_3, HSN_4, T_{gwnh})$ and counts $UID_i^* = SNX_j \oplus HSN_1$, $h(r_{SCn} \| SCN_i)^* = ID_{SNj} \oplus HSN_2$, and $r_{GWNh}^* = HSN_1 \oplus UID_i^* \oplus HSN_3$. Then, the assailant chooses stochastic digit r_{ASSk} and time T_{ass} to count $SK_s = h(r_{ASSk} \| r_{GWNh} \| UID_i \| h(r_{SCn} \| SCN_i) \| SNX_j)$, $SHN_1 = r_{ASSk} \oplus h(h(r_{SCn} \| SCN_i) \| r_{GWNh} \| T_{ass})$, and $SHN_2 = h(r_{GWNh} \| UID_i \| SK_s \| SNX_j \| T_{ass})$ as the valid sensor nodes. Nevertheless, SNX_j includes the private key SX_{SNj} of SN; hence, the assailant is unable to count the accurate information (SHN_1, SHN_2, T_{ass}) and the session key SK_A . The aforementioned sensor node impersonation attack is in case 1, and case 2 is identical to case 1. Consequently, our designed protocol is capable of resisting the sensor node impersonation attack.

5.9. Resist Reply Attack. In our designed protocol, we apply the time stamp in our communication information to resist reply attack. Suppose that the assailant captures the foregone communication information $(SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SNj})$ and intends to imitate the legitimate user to reply the information. GWN_H computes the freshness of the information by the formula $|T_{gwnh} - T_{sc}| \leq \Delta T$. If it is not right, GWN_H terminates the session promptly. Suppose that the assailant captures the foregone communication information $(HSN_1, HSN_2, HSN_3, HSN_4, T_{gwnh})$ and intends to imitate the legitimate gateway to reply the information. SN_j calculates the freshness of the information by the formula $|T_{snj} - T_{gwnh}| \leq \Delta T$. If it is not right, SN_j terminates the session promptly. Consequently, our designed protocol is capable of resisting reply attack.

5.10. Resist Privileged Insider Attack. In our designed protocol, U_i delivers UID_i and UPW_i to GWN_H as the registration request in registration section, where $UID_i = h(ID_i \| r_i)$ and $UPW_i = h(PW_i \| r_i \| r_p)$. If the identity and password are leaked to any privileged insider at GWN_H , this will lead to abundant security risks. The privileged insider is unable to extract the accurate identity ID_i and password PW_i from UID_i and UPW_i in the registration section on account of the irreversible one-way hash function $h(\cdot)$. Unaware of the stochastic digits r_i and r_p , the privileged insider is also unable to extract the accurate identity ID_i and password PW_i from UID_i and UPW_i in the registration section. Consequently, our designed protocol is capable of resisting privileged insider attack.

5.11. Resist Online Password-Guessing Attack. In our designed protocol, password PW_i never emerges in the delivered information in the communication process. Although the assailant is able to capture the communication information $(SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SNj})$,

$(HSN_1, HSN_2, HSN_3, HSN_4, T_{gwnh})$, (SHN_1, SHN_2, T_{snj}) , and $(HSN_3, GSC_1, GSC_2, GSC_3, GSC_4, T_{hgwn1})$, all the communication information does not directly associate with password PW_i . The aforementioned condition is in case 1, and case 2 is identical to case 1. Consequently, our designed protocol is capable of resisting online password-guessing attack.

5.12. Resist Offline Password-Guessing Attack. In our designed protocol, the assailant is able to capture the smart card and obtain the kept information GE_i, GF_i, USC_1, USC_2 , and USC_3 . The smart card contents containing password are $USC_1 = r_i \oplus h(ID_i \| PW_i \| \sigma_{Ui})$ and $USC_3 = h(UID_i \| UPW_i \| \sigma_{Ui} \| r_i \| r_p)$. For the purpose of speculating the password accurately, the assailant has to obtain ID_i and σ_{Ui} at the same time for USC_1 and has to obtain ID_i, r_i, r_p , and σ_{Ui} at the same time for USC_3 . It is impossible for the assailant to accurately compute these parameters at the same time. Consequently, our designed protocol is capable of resisting offline password-guessing attack.

5.13. Resist User Tracking Attack. In our designed protocol, parameter $GUID_i$ computed by the gateway node for the user is transformed into $GUID_i^{new} = GSC_3 \oplus h(UID_i \| UPW_i)$ after finishing the authentication process in case 1. Parameter $GUID_i$ computed by the gateway node for the user is transformed into $GUID_i^{new} = h(r_{GWNf}^{new} \| SX_{GWNf} \| ID_{GWNf}) \oplus UID_i$ after finishing the authentication process in case 2. Without knowing the relevant parameter, only known U_i , the assailant is unable to figure out the following $GUID_i^{new}$. Consequently, our designed protocol is capable of resisting user tracking attack.

5.14. Biometric Template Protection. In our designed protocol, the biometric information kept in smart card is first counted via $GEN(BIO_{U_i}) = (\sigma_{U_i}, \tau_{U_i})$ and the masked with the irreversible one-way hash function $USC_1 = r_i \oplus h(ID_i \| PW_i \| \sigma_{U_i})$, $USC_2 = r_p \oplus h(\sigma_{U_i} \| r_i)$, and $USC_3 = h(UID_i \| UPW_i \| \sigma_{U_i} \| r_i \| r_p)$. Even though the smart card is captured by the assailant, he is incapable of gaining any useful biometric information because the hash function is irreversible operation. Consequently, our designed protocol is capable of protecting the biometric template.

5.15. Mutual Authentication. In our designed protocol, U_i delivers the login request $(SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SNj})$ to GWN_H . After reception of the information, GWN_H authenticates U_i by computing SCG_4^* . GWN_H transmits $(HSN_1, HSN_2, HSN_3, HSN_4, T_{gwnh})$ to SN_j . After reception of the information, SN_j authenticates GWN_H by computing HSN_4^* . SN_j dispatches (SHN_1, SHN_2, T_{snj}) to GWN_H . After reception of the information, GWN_H authenticates SN_j by computing SHN_2^* . GWN_H transmits $(HSN_3, GSC_1, GSC_2, GSC_3, GSC_4, T_{hgwn1})$ to U_i . After reception of the information, U_i authenticates GWN_H by computing GSC_4^* . The aforementioned mutual authentication is in case 1, and case 2 is identical to case 1.

Consequently, our designed protocol is capable of achieving the mutual authentication.

5.16. User Anonymity. In our designed protocol, the assailant is able to capture the login request ($SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SN_j}$) and obtain the kept information GE_i, GF_i, USC_1, USC_2 , and USC_3 in the stolen smart card. The assailant will figure out identity ID_i via $h(UID_i || UPW_i) = GF_i \oplus SCG_3$, where $UID_i = h(ID_i || r_i)$. In order to figure out GF_i , the assailant has to speculate parameters r_{GWN_H} and SX_{GWN_H} , which are only known to GWN_H . Moreover, UPW_i includes parameters PW_i and r_p , which are only known to U_i . Consequently, our designed protocol is capable of achieving user anonymity.

6. Performance Comparison

In this section, we will demonstrate performance comparisons of our proposed mutual authentication protocol with other related mutual authentication protocols in terms of security, computation time, and communication cost.

6.1. Security Comparison. The security comparison result is shown in Table 1. From [1], we know that [25] cannot resist offline and online password-guessing attack. As shown in [25], the authors' security analysis does not mention or refer to IF5, IF7, IF10, and IF13. As shown in [46], the authors' security analysis does not mention or refer to IF2, IF4, and IF11. From [1], we know that [45] and [9] cannot resist IF5 and cannot achieve IF16 and IF3. As shown in [50], the authors' security analysis does not mention or refer to IF2, IF4, IF11, IF12, and IF14. As shown in [8], the authors' security analysis does not mention or refer to IF3, IF5, IF7, and IF14. From [47], we know that [48] cannot resist reply and sensor node capture attacks. As shown in [47], the authors' security analysis does not mention or refer to IF2, IF11, and IF12. As shown in [49], the authors' security analysis does not mention or refer to IF2, IF13, IF14, and IF15.

6.2. Computation Time Comparison. The computation time comparison result is presented in Table 2. We directly obtain the communication costs in the corresponding references as shown in Table 2. We can see that some references [47–49] add fingerprint operations to communication cost, while some references [8, 9, 25, 45] do not. In order to make a unified communication cost comparison, we will not add the fingerprint operations to communication cost. In this comparison, we specify that H represents the time of hash function operation, E/D represents the time of encryption and decryption operation, MM represents the time of modular multiplication operation, and EM represents the time of ECC point multiplication operation. We apply the experimental results of $EM = 0.0171$ s [46], $H = 0.00032$ s [7], $E/D = 0.0056$ s [7], and $MM = 0.0002586$ s [47] to compute computation cost. The total communication time in our designed protocol is $27H = 0.00864$ s in case 1 and

TABLE 1: Security comparison.

	[45]	[9]	[46]	[25]	[47]	[48]	[8]	[49]	[50]	Ours
IF1	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
IF2	Y	Y	N	Y	N	Y	Y	N	N	Y
IF3	N	N	Y	Y	Y	Y	N	Y	Y	Y
IF4	Y	Y	N	Y	Y	N	Y	Y	N	Y
IF5	N	N	Y	N	Y	Y	N	Y	Y	Y
IF6	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
IF7	Y	Y	Y	N	Y	Y	N	Y	Y	Y
IF8	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
IF9	Y	Y	Y	Y	Y	N	Y	Y	Y	Y
IF10	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
IF11	Y	Y	N	N	N	Y	Y	Y	N	Y
IF12	Y	Y	Y	N	N	Y	Y	Y	N	Y
IF13	Y	Y	Y	N	Y	Y	Y	N	Y	Y
IF14	Y	Y	Y	Y	Y	Y	N	N	N	Y
IF15	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
IF16	N	N	Y	Y	Y	Y	Y	Y	Y	Y

IF1: session key security; IF2: three-factor security; IF3: perfect forward and backward security; IF4: resist sensor node capture attack; IF5: resist stolen smart card attack; IF6: resist user impersonation attack; IF7: resist gateway impersonation attack; IF8: resist sensor node impersonation attack; IF9: resist reply attack; IF10: resist privileged insider attack; IF11: resist online password-guessing attack; IF12: resist offline password-guessing attack; IF13: resist user tracking attack; IF14: biometric template protection; IF15: mutual authentication; IF16: user anonymity; Y: yes; N: no or not mentioned.

$43H = 0.0137$ s in case 2. Although the communication cost is higher than the communication time in [7], our designed protocol has higher level of security. Compared with other authentication protocols, no matter in case 1 or in case 2, our designed protocol has higher level of computation cost and is more suitable for the resource-constrained wireless sensor networks.

6.3. Communication Cost Comparison. The communication cost comparison result is revealed in Table 3. In order to make a unified and thorough communication cost comparison, we make the following assumptions that the identity of user is 160 bits, the identity of gateway node or base station is 160 bits, the identity of sensor node is 32 bits, the stochastic digit is 128 bits, the result of symmetric encryption/decryption is 128 bits, the time stamp size is 32 bits, the result of hash function is 160 bits, and the result of ECC point multiplication operation is 160 bits.

In case 1, the communication cost of the information ($SCG_1, SCG_2, SCG_4, T_{sc}, UID_i, ID_{SN_j}$) delivered from U_i to GWN_H is 160 bits + 160 bits + 160 bits + 32 bits + 160 bits + 32 bits = 704 bits; the communication cost of the information ($HSN_1, HSN_2, HSN_3, HSN_4, T_{gwnh}$) transmitted from GWN_H to SN_j is 160 bits + 160 bits + 128 bits + 160 bits + 32 bits = 640 bits; the communication cost of the information (SHN_1, SHN_2, T_{snj}) dispatched from SN_j to GWN_H is 160 bits + 160 bits + 32 bits = 352 bits; and the communication cost of the information ($HSN_3, GSC_1, GSC_2, GSC_3, GSC_4, T_{hgwn1}$) transmitted from GWN_H to U_i is 160 bits + 160 bits + 160 bits + 160 bits + 160 bits + 32 bits = 832 bits.

TABLE 2: Computation time comparison.

	User	Sensor node	Home gateway/base station	Foreign gateway/base station	Total time
[45]	$6H + 2E/D$	$5H + 1E/D$	$8H + 3E/D$	$5H + 2E/D$	$24H + 8E/D = 0.0524$ s
[9]	$14H$	$4H$	$17H$	0	$35H = 0.0112$ s
[46]	$8H + 3EM$	0	$7H + 3EM$	0	$15H + 6EM = 0.107$ s
[25]	$8H + 3EM$	$4H + 2EM$	$8H + EM$	0	$20H + 6EM = 0.109$ s
[47]	$9H + 2EM$	$5H$	$10H + 1EM$	0	$24H + 3EM = 0.0589$ s
[48]	$9H + 1E/D$	$4H + 2E/D$	$6H + 3E/D + 2MM$	0	$19H + 6E/D + 2MM = 0.0412$ s
[8]	Case 1: $13H$ Case 2: $18H$	Case 1: $6H$ Case 2: $6H$	Case 1: $17H$ Case 2: $10H$	Case 1: $0H$ Case 2: $14H$	Case 1: $36H = 0.0115$ s Case 2: $48H = 0.0153$ s
[49]	Case 1: $12H + 3EM$ Case 2: $12H + 4EM$	Case 1: $5H$ Case 2: $5H$	Case 1: $6H + 3EM$ Case 2: 0	Case 1: 0 Case 2: $8H + 3EM$	Case 1: $23H + 6EM = 0.1099$ s Case 2: $25H + 6EM = 0.111$ s
[50]	Case 1: $7H + 2EM$ Case 2: $7H + 2EM$	Case 1: $4H + 2EM$ Case 2: $4H + 2EM$	Case 1: $11H$ Case 2: $9H + 1EM$	Case 1: $0H$ Case 2: $8H + 1EM$	Case 1: $22H + 4EM = 0.0754$ s Case 2: $28H + 6EM = 0.1116$ s
Ours	Case 1: $12H$ Case 2: $16H$	Case 1: $5H$ Case 2: $7H$	Case 1: $12H$ Case 2: $6H$	Case 1: $0H$ Case 2: $14H$	Case 1: $27H = 0.00864$ s Case 2: $43H = 0.0137$ s

TABLE 3: Communication cost comparison.

	User (bits)	Sensor node (bits)	Home gateway/base station (bits)	Foreign gateway/base station (bits)	Total cost (bits)
[45]	608	352	448	736	2144
[9]	983	352	1344	0	2679
[46]	864	0	512	0	1376
[25]	640	480	960	0	2080
[47]	928	416	1312	0	2656
[48]	512	160	1440	0	2112
[8]	Case 1: 864 Case 2: 512	Case 1: 352 Case 2: 352	Case 1: 1344 Case 2: 1184	Case 1: 0 Case 2: 1376	Case 1: 2560 Case 2: 3424
[49]	Case 1: 1024 Case 2: 1024	Case 1: 352 Case 2: 352	Case 1: 800 Case 2: 544	Case 1: 0 Case 2: 800	Case 1: 2176 Case 2: 2720
[50]	Case 1: 864 Case 2: 864	Case 1: 1728 Case 2: 1728	Case 1: 640 Case 2: 832	Case 1: 0 Case 2: 1504	Case 1: 3232 Case 2: 4928
Ours	Case 1: 704 Case 2: 512	Case 1: 352 Case 2: 352	Case 1: 1472 Case 2: 480	Case 1: 0 Case 2: 1920	Case 1: 2528 Case 2: 3264

In case 2, the communication cost of the information ($FHN_1, FHN_2, FHN_3, FHN_4$) transmitted from GWN_F to GWN_H is 160 bits + 160 bits + 128 bits + 160 bits = 608 bits; the communication cost of the information (FHN_2, GSC_6, GSC_7) transmitted from GWN_H to U_i is $160 + 160 + 160 = 480$ bits; the communication cost of the information ($SCF_5, SCF_6, SCF_7, T_{ui}$) delivered from U_i to GWN_F is 160 bits + 160 bits + 160 bits + 32 bits = 512 bits; the communication cost of the information ($FSN_1, FSN_2, FSN_3, FSN_4, T_{gwnf}$) transmitted from GWN_F to SN_j is 160 bits + 160 bits + 160 bits + 160 bits + 32 bits = 672 bits; the communication cost of the information (SFN_2, SFN_3, T_{snj}) dispatched from SN_j to GWN_F is 160 bits + 160 bits + 32 bits = 352 bits; and the communication cost of the information ($FSC_1, FSC_2, FSC_3, FSC_4, T_{gwnf1}$) transmitted from GWN_F to U_i is 128 bits + 160 bits + 160 bits + 32 bits = 640 bits.

Compared with the other authentication protocols, the total communication cost in our protocol is a bit higher than those in the other protocols [25, 45, 46, 48, 49]. During the authentication process, the number of information exchanges in the protocols in [46, 48, 49] is less than ours and the sensor nodes require more communication cost than the

gateway node in the protocol in [50]. Because the sensor nodes are resource-constrained, the communication costs of the sensor nodes shall be reduced. The sensor nodes' communication costs in our protocol are lower than those in the other comparison protocols. The communication cost is acceptable as our designed authentication protocol achieves additional security features and has lower computation time.

7. Conclusion

To overcome the problems that the sensor nodes need to execute heavy calculation and communication consumption during the authentication process and cannot resist node capture attack and that the protocols also cannot provide perfect forward and backward security and cannot resist replay attack, we put forward a novel multifactor user authentication and key agreement scheme for multigateway wireless sensor networks in this paper. In our authentication protocol, we apply the lightweight hash function and given biometric information to achieve a higher level of security and efficiency, as well as a larger communication coverage area. Our authentication protocol meets sixteen evaluation criteria. We separately apply BAN logic, random oracle

model, and AVISPA tool to validate the security of our authentication protocol. Our authentication protocol is able to achieve higher security and is more efficient in communication and computation costs as compared with the related authentication protocols.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China, under Grant no. 61962052, IoT Innovation Team Foundation of Qinghai Office of Science and Technology, under Grant no. 2020-ZJ-903, Key Laboratory of IoT of Qinghai (2020-ZJ-Y16), and Hebei IoT Monitoring Center (3142016020).

References

- [1] C. Wang, D. Wang, Y. Tu et al., "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.
- [2] D. Singh, B. Kumar, S. Singh et al., "Evaluating authentication schemes for real-time data in wireless sensor network," *Wireless Personal Communications*, vol. 114, no. 3, 2020.
- [3] Y. Lu, G. Xu, L. Li et al., "Robust privacy-preserving mutual authenticated key agreement scheme in roaming service for global mobility networks," *IEEE Systems Journal*, vol. 13, pp. 1–12, 2019.
- [4] D. Wang, W. Li, and P. Wang, "Measuring two-factor Authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.
- [5] C. Wang, Y. Wang, Y. Chen, H. Liu, and J. Liu, "User authentication on mobile devices: approaches, threats and trends," *Computer Networks*, vol. 170, no. 2, Article ID 107118, 2020.
- [6] S. Jangirala, D. A. Kumar, W. Mohammad, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial internet of things," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, p. 1, 2018.
- [7] M. Wazid, A. K. Das, V. Odelu et al., "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, p. 1, 2017.
- [8] H. Guo, Y. Gao, T. Xu et al., "A secure and efficient three-factor multi-gateway authentication protocol for wireless sensor networks," *Ad Hoc Networks*, vol. 95, Article ID 101965.1, 2019.
- [9] R. Amin, S. K. H. Islam, N. Kumar, and K. K. R. Choo, "An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks," *Journal of Network and Computer Applications*, vol. 104, 2018.
- [10] P. Gope and T. Hwang, "Lightweight and energy-efficient mutual authentication and key agreement scheme with user anonymity for secure communication in global mobility networks," *IEEE Systems Journal*, vol. 10, no. 4, pp. 1–10, 2016.
- [11] G. Xu, J. Liu, Y. Lu, X. Zeng, Y. Zhang, and X. Li, "A novel efficient MAKKA protocol with desynchronization for anonymous roaming service in Global Mobility Networks," *Journal of Network and Computer Applications*, vol. 107, pp. 83–92, 2018.
- [12] W. Fan, L. Xu, S. Kumari et al., "An enhanced mutual authentication and key agreement scheme for mobile user roaming service in global mobility networks," *Annals of Telecommunications*, vol. 72, no. 3–4, pp. 1–14, 2016.
- [13] G. Mohit and S. C. Narendra, "Anonymous two factor authentication protocol for roaming service in global mobility network with security beyond traditional limit," *Ad Hoc Networks*, vol. 84, pp. 56–67, 2019.
- [14] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4900–4913, 2018.
- [15] S. Hussain and S. A. Chaudhry, "Comments on "biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment"" *IEEE Internet of Things Journal*, vol. 6, no. 6, Article ID 10936, 2019.
- [16] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Computer Networks*, vol. 101, pp. 42–62, 2016.
- [17] A. O. Sharif, H. Arshad, M. Nikooghadam et al., "Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme," *Future Generation Computer Systems*, vol. 100, 2019.
- [18] F. Wu, X. Li, L. Xu et al., "A novel three-factor Authentication protocol for wireless sensor networks with IoT notion," *IEEE Systems Journal*, vol. 15, no. 99, pp. 1–10, 2020.
- [19] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Information Sciences*, vol. 321, pp. 263–277, 2015.
- [20] S. Kumari, X. Li, F. Wu, A. K. Das, H. Arshad, and M. K. Khan, "A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps," *Future Generation Computer Systems*, vol. 63, pp. 56–75, 2016.
- [21] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 76, pp. 37–48, 2016.
- [22] L. Xiong, J. Niu, S. Kumari et al., "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network and Computer Applications*, vol. 103, 2018.
- [23] A. K. Das, "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," *International Journal of Communication Systems*, vol. 30, no. 1, Article ID e2933.1, 2017.
- [24] Y. Lu, G. Xu, L. Li et al., "Anonymous three-factor authenticated key agreement for wireless sensor networks," *Wireless Networks*, vol. 25, 2017.
- [25] X. Li, J. Peng, M. S. Obaidat et al., "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems," *IEEE Systems Journal*, vol. 14, no. 99, pp. 1–12, 2019.

- [26] A. Ruhul, S. K. HafizulIslam, G. P. Biswas, M. Khurram Khan, and N. Kumar, "A robust and anonymous patient monitoring system using wireless medical sensor networks," *Future Generations Computer Systems Fgcs*, vol. 80, 2018.
- [27] J. Mo and H. Chen, "A lightweight secure user authentication and key agreement protocol for wireless sensor networks," *Security and Communication Networks*, vol. 2019, no. 4, pp. 1–17, 2019.
- [28] L. Xiong, J. Niu, S. Kumari et al., "A robust biometrics based three-factor authentication scheme for Global Mobility Networks in smart city," *Future Generation Computer Systems*, vol. 83, pp. 607–618, 2017.
- [29] L. Xiong, A. K. Sangaiah, S. Kumari, F. Wu, J. Shen, and M. K. Khan, "An efficient authentication and key agreement scheme with user anonymity for roaming service in smart city," *Personal & Ubiquitous Computing*, vol. 21, 2017.
- [30] B. Yza, B. Dha, L. A. Li et al., "A lightweight authentication and key agreement scheme for Internet of Drones," *Computer Communications*, vol. 154, pp. 455–464, 2020.
- [31] M. Wazid, A. K. Das, N. Kumar et al., "Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of Drones deployment," *IEEE Internet of Things Journal*, vol. 6, p. 1, 2018.
- [32] P. Vijayakumar, V. Chang, L. J. Deborah et al., "Computationally efficient privacy preserving anonymous mutual and batch Authentication schemes for vehicular ad hoc networks," *Future Generation Computer Systems*, vol. 78, 2016.
- [33] C. Xu, X. Huang, M. Ma, and H. Bao, "An anonymous handover authentication scheme based on LTE-A for vehicular networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 6251219, 15 pages, 2018.
- [34] S. Chatterjee, S. Roy, A. K. Das et al., "Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 99, p. 1, 2016.
- [35] T. Sudhakar, V. Natarajan, M. Gopinath, and J. Saranyadevi, "An enhanced authentication protocol for multi-server environment using password and smart card," *Wireless Personal Communications*, vol. 115, no. 2, 2020.
- [36] S. Qiu, D. Wang, G. Xu et al., "Practical and provably secure three-factor Authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.
- [37] W. Ding, C. Haibo, H. Debiao, and W. Ping, "On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices," *IEEE Systems Journal*, vol. 12, 2016.
- [38] L. Xiong, W. Fan, M. K. Khan et al., "A secure chaotic map-based remote authentication scheme for telecare medicine information systems," *Future Generation Computer Systems*, vol. 84, pp. 149–159, 2017.
- [39] J. Mo, W. Shen, and W. Pan, "An improved anonymous authentication protocol for wearable health monitoring systems," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 5686498, 13 pages, 2020.
- [40] A. Irshad, S. A. Chaudhry, Q. Xie et al., "An enhanced and provably secure chaotic map-based authenticated key agreement in multi-server architecture," *Arabian Journal for Science and Engineering*, vol. 43, 2017.
- [41] R. Amin and G. P. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Networks*, vol. 36, no. 1, pp. 58–80, 2016.
- [42] S. Jangirala, S. Mukhopadhyay, and A. K. Das, "A multi-server environment with secure and efficient remote user authentication scheme based on dynamic ID using smart cards," *Wireless Personal Communications*, vol. 95, 2017.
- [43] J. Srinivas, S. Mukhopadhyay, and D. Mishra, "Secure and efficient user authentication scheme for multi-gateway wireless sensor networks," *Ad Hoc Networks*, vol. 54, pp. 147–169, 2017.
- [44] D. Kumar, S. Chand, and B. Kumar, "Cryptanalysis and improvement of an authentication protocol for wireless sensor networks applications like safety monitoring in coal mines," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 1, pp. 1–20, 2018.
- [45] A. Rifaqat, P. Alrup Kumar, K. Saru, K. Marimuthu, and C. Mauro, "A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring," *Future Generation Computer Systems*, vol. 84, 2018.
- [46] Y. Chen and J. Chen, "A secure three-factor-based authentication with key agreement protocol for e-Health clouds," *The Journal of Supercomputing*, vol. 77, no. 3, 2020.
- [47] H. Far, M. Bayat, A. K. Das et al., "LAPTAS: lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT," *Wireless Networks*, vol. 27, no. 4, pp. 1–24, 2021.
- [48] R. Vinoth, L. J. Deborah, P. Vijayakumar, and N. Kumar, "Secure multifactor Authenticated key agreement scheme for industrial IoT," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3801–3811, 2021.
- [49] A. K. Sutrala, A. K. Das, N. Kumar, A. G. Reddy, A. V. Vasilakos, and J. J. P. C. Rodrigues, "On the design of secure user authenticated key management scheme for multigateway-based wireless sensor networks using ECC," *International Journal of Communication Systems*, vol. 31, no. 8, Article ID e3514, 2018.
- [50] J. Guo and Y. Du, "A secure three-factor anonymous roaming authentication protocol using ECC for space information networks," *Peer-to-Peer Networking and Applications*, vol. 14, no. 9, pp. 1–19, 2021.

Research Article

From Unknown to Similar: Unknown Protocol Syntax Analysis for Network Flows in IoT

Yichuan Wang ^{1,2}, Han Yu ^{1,2}, Xinhong Hei ^{1,2}, Binbin Bai,^{1,2} and Wenjiang Ji^{1,2}

¹*Xi'an University of Technology, School of Computer Science and Engineering, Xi'an, China*

²*Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an, China*

Correspondence should be addressed to Xinhong Hei; heixinhong@xaut.edu.cn

Received 7 May 2021; Revised 20 June 2021; Accepted 17 July 2021; Published 2 August 2021

Academic Editor: Qing Yang

Copyright © 2021 Yichuan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Internet of Things (IoT) is the development and extension of computer, Internet, and mobile communication network and other related technologies, and in the new era of development, it increasingly shows its important role. To play the role of the Internet of Things, it is especially important to strengthen the network communication information security system construction, which is an important foundation for the Internet of Things business relying on Internet technology. Therefore, the communication protocol between IoT devices is a point that cannot be ignored, especially in recent years; the emergence of a large number of botnet and malicious communication has seriously threatened the communication security between connected devices. Therefore, it is necessary to identify these unknown protocols by reverse analysis. Although the development of protocol analysis technology has been quite mature, it is impossible to identify and analyze the unknown protocols of pure bitstreams with zero a priori knowledge using existing protocol analysis tools. In this paper, we make improvements to the existing protocol analysis algorithm, summarize and learn from the experience and knowledge of our predecessors, improve the algorithm ideas based on the Apriori algorithm idea, and perform feature string finding under the idea of composite features of CFI (Combined Frequent Items) algorithm. The advantages of existing algorithm ideas are combined together to finally propose a more efficient OFS (Optimal Feature Strings) algorithm with better performance in the face of bitstream protocol feature extraction problems.

1. Introduction

As the global economy continues to develop, the impact of scientific and technological advances on the daily lives of people around the world is gradually increasing. The Internet of Things (IoT) technology, which is derived from the advancement of science and technology, has also been developed significantly and has been applied in various industries around the world. The Internet of Things (IoT) has emerged in the context of information technology development, and its degree of development has been influenced by the processing power of information technology in the information age [1–4]. The popularization and development of IoT technology mark the comprehensive and integrated development of network information technology for the whole human race, which has laid a solid hardware foundation for the complete interconnection of all countries around the world.

IoT technology is characterized by a major shift in the mode of accessing and applying information among people, as well as a gradual change and subversion of people's behavioral patterns such as clothing, food, housing, and transportation [5–8]. Especially, in the fields of smart home, autonomous driving, and health care, IoT is already quietly changing people's lifestyles and will even have a further impact on all details of human life in the future [9–12]. It is worth everyone's attention that despite the good prospects and speed of development of IoT technology in developing and developed countries around the world in recent years, cybersecurity issues between IoT devices are frequent, which can have a certain impact on the global economic and technological development [13, 14].

As economic globalization continues, network communication technology has turned the whole world into a global village. Currently, the number of Internet users

worldwide has exceeded 5 billion, which shows that Internet devices are essential in everyone's daily life, and communication protocols play a very important role as a bridge for data communication between networked devices, so the classification and identification of these communication protocols has been a popular topic. Moreover, various network security incidents have appeared in people's view recently, and the endless malicious network attacks have brought a lot of economic losses and psychological panic to people [15, 16]. Whether the communication security of the network can be ensured is related to the fundamental interests of individuals, enterprises, and the country. It is significant to carry out reasonable and effective network maintenance and network regulation at this time [17–19]. Therefore, it is necessary to analyze and identify the unknown protocols in the network in order to better regulate the network security.

The three basic elements of network protocols are semantics, syntax, and timing. The inference of the protocol message format and the determination of its field contents belong to the protocol syntax analysis [20]. The analysis and extraction of protocol syntax is the basis of protocol analysis and identification. It requires analysis of the control statements of protocol messages and extraction of the protocol semantics based on data mining and sequence ratio methods. The purpose of protocol syntax rule inference is to build a logical model of the protocol syntax, focusing on the intrinsic logical relationships between protocol messages. How protocols interact must follow certain syntax rules.

Analyzing the role of network protocol specifications in the field of network regulation can help us to obtain information about the network traffic in the target network [21, 22]. By classifying the traffic generated by these protocols, network usage can be identified, network expansion plans can be developed, and bandwidth for specific protocols can be controlled. Protocol analysis can provide useful information to firewalls and intrusion detection prevention systems to help analyze network vulnerabilities and thus prevent and detect unknown attacks.

The bitstream protocol format analyzer works at the bottom of the network environment, analyzing the content of the acquired bitstream protocol data in real time by parsing the data and analyzing the protocol format. The current network protocol analysis method, with the huge number of analyzed protocol frames and the complexity of the data frames themselves, can take a long time for the algorithm to run, and how to optimize the algorithm is a research direction that needs to be continuously studied.

In general, the rapid development of IoT has brought us some opportunities and challenges, and the security of communication between IoT devices is now the biggest challenge; specifically, there are still a lot of defects about the unknown protocol analysis. When processing a large number of protocols, the complexity of data processing is large and the system response speed is still slow and needs to be optimized in order to play a role in the actual scenario. It needs to be optimized to be useful in practical scenarios. When capturing data, there may be more than one protocol type, the length may be inconsistent, and many protocol

identification methods will be greatly affected. Therefore, to propose an algorithm to solve the problem of intelligent inverse analysis of unknown protocols in connection with practical difficulties is the main research of this paper.

The rest of this paper is organized as follows: the first part introduces the current state of development and security issues of IoT technology worldwide, and the second part presents our work related to the unknown protocol parsing. The third part proposes a new protocol format analysis algorithm. The fourth part analyzes the performance of the new algorithm from several aspects and compares it with other algorithms. The fifth part concludes the work.

2. Related Work

Most of the existing studies on protocol identification have been based on content, port, and behavior-based protocol identification techniques [23]. The earliest protocol identification method used is the port number-based protocol identification technique. Port number-based protocol identification is well guaranteed in terms of correctness and efficiency in identifying traditional TCP/IP protocols. The algorithm principle of this technique is to use the service port number of TCP/IP protocol to identify the underlying protocol, and then compare the identified port number with the port number issued by IANA (Internet Assigned Numbers Authority) [24], and find the correspondence between the port and the protocol by cross-referencing to know the identified protocol's type. However, this technique also has certain defects because the port numbers managed by IANA are not all static, and some port numbers are dynamic, and dynamic port numbers can be easily controlled by Trojan horse programs to carry out network attacks and endanger the security of the Internet. With the continuous development of the Internet, new protocols are born that tend to use dynamic port numbers and no longer use IANA to register port numbers, at which time port-based protocol identification methods are no longer efficient and accurate. The reasons for the failure of this technique have been analyzed in the related literature because the lack of necessary semantic information and corresponding protocol specifications for unknown bitstream protocols, not to mention the unavailability of any information about the protocol ports, makes the port number-based protocol analysis technique nowadays not applicable to the field of reverse identification of unknown protocols for bitstreams.

Multiple sequence comparison techniques in genetics [25–28] can extract similar segments in DNA, and in the field of bitstream protocol, reverse identification also requires the extraction of format-specific message segments from messages; therefore, multiple sequence comparison techniques can also be applied to protocol format inference, where researchers infer and analyze message format information by extracting variable and immutable fields in identified messages. Pi, ScriptGen, Discoverer, and Netzob use bioinformatics-based sequence matching techniques to determine message similarities and cluster them. They, then, separate messages by identifying common parts between messages in the same group. The amount of data has a

significant impact on the quality of the protocol specification, but multiple sequence matching has exponential complexity because sequence matching algorithms use only two messages at a time as input [29].

Zhang et al. [30–32] studied and proposed a feature extraction method combining multipattern matching and association rules by investigating the bitstream protocol feature extraction technique to divide the bitstream protocol multiprotocol data frames into single-protocol data frames. The work is done for offline data, which cannot meet the real-time nature of bitstream data analysis and identification. Moreover, although the accuracy of the method is high, it still takes a long time to run for a large amount of data centrally and consumes more resources.

Youxiang et al. proposed a semiautomatic protocol inverse analysis method based on artificial knowledge [33], which suggests that sociological engineering and artificial guessing can be used to obtain a priori knowledge such as “field semantics,” length, and boundaries in the process of protocol format identification. The method first separates the segments of the message sequence associated with the a priori knowledge and then uses them as the basis for subsequent format inference, deriving the semantic inference of the fields using the a priori knowledge and verifying the results. After experimental validation, it is concluded that the method can verify the semantics and format of the obtained fields and improve the accuracy of the initial clustering, thus greatly improving the accuracy of format inference for unknown protocols.

Xiao-Li et al. conducted intensive research on existing protocol recognition techniques [34–37] to enumerate the algorithms and principles related to pattern matching and data mining by analyzing their strengths and weaknesses. A detailed presentation is also provided for the analysis of a considerable amount of bitstream data using the relevant theories of data mining to analyze the meaning relationships in order to find all possible candidate strings. Then, pattern matching algorithms are used for further analysis.

In addition to sequence comparison techniques, data mining techniques can also be used to perform inverse analysis of protocols. Karimov et al. used the Apriori algorithm to extract the keywords and message format of the protocol. The Aho-Corasick algorithm was first used for keyword extraction of protocols [38], followed by the frequent pattern FP-Growth algorithm to extract the format of messages [39]. Unlike the sequence comparison technique, the data mining technique takes all messages as input at once, which directly leads to a large computational cost for candidate selection. In addition, it is crucial to learn how to optimize the results to make them intuitive and clear.

3. Protocol Feature Extraction Algorithm

In order to avoid and improve the shortcomings of the algorithm described in the previous section, OFS, a protocol feature extraction algorithm based on the idea of the Apriori algorithm and the idea of composite features of CFI, is proposed, which is different from the idea of CFI algorithm. The previous algorithm tends to iterate to find feature strings

from nothing, while the OFS algorithm tends to find the range of possible feature strings at one time and then go to search feature strings within the range. This chapter will introduce the idea and steps of the OFS algorithm.

3.1. Algorithm Ideas

3.1.1. Algorithm-Related Definitions. To better illustrate the algorithm, some definitions are introduced and presented here.

Definition 1. Minimum support: A reasonable threshold defined by the user to measure the magnitude of support, which in a statistical sense represents the minimum standard of the importance of the data; here, we use Min_sup to represent it.

Definition 2. Frequent substring: If there are N data frame messages, these are sequences of bits of length $L1$, there exists a substring α of length $L2$ ($L1 > L2$), and if substring α has occurrences in M of the N data frames, the probability of occurrence of α is $P(M/N)$. If the probability of a substring occurring is greater than or equal to Min_Sup, then the string is called a frequent substring:

$$\text{Seq} = \{\alpha | P(\alpha) \geq \text{Min_sup}\}. \quad (1)$$

Definition 3. Minimum frequent substring length: A user-defined value where the length of a frequent substring is filtered out if it is less than the length of the least frequent substring, denoted as Min_sup.

Definition 4. Protocol feature: If frequent substring α appears frequently at one or more specific locations in the protocol data frame, it is considered likely that the frequent substring is a protocol feature of the protocol.

3.1.2. Algorithm Data Initialization. Algorithm data initialization is a five-step process:

- (1) Enter the support threshold Min_sup, traverse the data set, and record the length of the longest data in the data set as Min_sup (the length of the longest data in the data set).
- (2) Define a one-dimensional vector Vector and initialize all elements of it to 0 with Max_len.
- (3) Traverse all the data frames in the data set and record whether the elements of each position of each data are 0. If the value of the data at a position is 0, let the value of the one-dimensional vector at the corresponding position of that data be added by 1. For example, if the i -th position data $[i]$ of a data is equal to 0, then the value of the one-dimensional vector at its corresponding position is added by 1, that is, Vector $[i]$ is added by 1.
- (4) The support of each position is calculated by traversing the vector Vector once. If the position support $\text{sup} \geq \text{Min_Sup}$ or $\text{sup} \leq 1 - \text{Min_Sup}$

(assuming $\text{Min_Sup} > 0.5$), it means that the position may exist in a feature string; it is impossible for the position to exist in a feature string.

After calculating the support for each position, two more important definitions need to be stated to complement the data initialization process of the algorithm, Definition 5 and 6.

Definition 5. Bad characters: If the support of a position is not within the range specified in step (4), the character at that position is considered to be a bad character and is denoted as C_i .

Definition 6. Ideal string: The substring that appears between two adjacent bad characters in a one-dimensional vector Vector is called the ideal string. If there is only one bad character B_1 in a certain data frame, the substring from the beginning of the vector Vector0 all the way to B_1 (containing the character at 0 but not at B_1) is considered as the ideal string and similarly from B_1 all the way to the end of the vector Vector is also considered as an ideal string. The minimum frequent substring Min_len can be used to filter part of the ideal string.

- (5) After the processing of the above steps, all ideal strings will be obtained by Vector and the position of bad characters, and then these ideal string records are put into a set prunSet (the set of ideal strings).

3.1.3. Data Reprocessing. After the initialization of the algorithm data, we get the data set prunSet after preprocessing; the data set contains all the possible locations of the feature string, but the range of the occurrence of each feature string is too large, which is not convenient enough for the specific search of the feature string afterwards. Because this operation of frequency statistics for each position ignores the continuity of the string, the range obtained is relatively large, so we use the continuous property of the string to perform in processing in a good way; the specific steps are as follows:

- (1) Iterate over each data Str (string in the ideal string set) of the data set prunSet to get the Str length of each data and use this length to build a one-dimensional vector Vector, so that its value is 0.
- (2) Reiterate the data set dateSet (the original data set), intercept the string date (the string in the original data set) with the same length and the same position as Str in the data set, cut Str and date using the length of the least frequent substring Min_len , and judge whether they are equal. If they are equal, then the one-dimensional vector $\text{Vector}[i]$ corresponding to the cut position is added by one; if not equal, there is no operation.
- (3) Then, we refer to steps 3, 4, and 5 of the algorithm data initialization to obtain the updated prunSet , and the data processing operation of the algorithm is completed.

3.2. Algorithm Flow. The entire flow of the algorithm is described in Table 1, and Figure 1 shows the algorithm flowchart.

From Table 1, we can see that the OFS algorithm divides the data processing into two stages: preprocessing and reprocessing. Although the data is preprocessed to obtain the approximate range of the ideal string, only the fixed position of the ideal string is obtained without using the continuity of the ideal string, so the range of the ideal string is very large, which contains a large amount of useless information. Therefore, the role of data reprocessing at this time is to use the definition of the minimum frequent substring length to further reduce the range of the ideal string obtained by preprocessing, the operation will eliminate a large number of useless information, making the subsequent data operations much more efficient.

3.2.1. The Process of Obtaining the Set of Items. From the above operation, we can conclude that prunSet is a collection of all ideal strings, so naturally the frequent substrings must also be obtained from the ideal string. Suppose a certain ideal string is "0010001000010001001001#47"; the string intercepted from the corresponding position of the data frame set dateSet is 0010001001010001001001#47. By comparison, we can see that the two strings differ only in the characters of position 56. The comparison shows that the two strings differ only in the characters at position 56. Since this is the case, we can separate the two substrings "001000100#47" and "010001001001#57" from this data. Put them into a new set of singleMap (ideal string of substring collection); all the data frames of the data frame collection dateSet to intercept and compare the separation can get the ideal string of the substring collection and all the ideal string to obtain the operation can get all the ideal string of singleMap.

3.2.2. Removing the Include Operation. After getting the item set of the ideal string, singleMap belonging to the ideal string is obtained, and singleMap needs to be removed from the containment operation. In an ideal string, there are two substrings: "000010001001#223" and "010001001#226." Obviously, the substring at position 226 is a true suffix of the substring 223, and the case becomes a postinclusion. Similarly, if a substring is a true prefix of another substring, it is a preinclusion. If the true prefix of a substring is the true suffix of another substring, the case is called mutual inclusion.

Postinclusion can cause the number of substrings to be counted incorrectly, which can lead to missing frequent substrings. This is because the counts of them are counted separately in singleMap. Consider an extreme case where "00100010110#402" appears in the first 50% of the data frames of the data Set and "0010110#406" appears in the last 50% of the data frames of the data Set. If Min_sup is 0.7 at this time, then both substrings cannot be used as frequent substrings. However, the string "0010110#406" is obviously a feature string, because it actually appears in 100% of the data. Therefore, when dealing with such cases, we need to add the number of times the string "00100010110#402" is in the

TABLE 1: Description of OFS algorithm.

Input	The ideal string set of data preprocessing is prun Set; the data frame set is data Set; the minimum support is Min_Sup
Output	Frequent item set feature Map
(1)	Judge whether prun Set is empty
(2)	If it is empty, the algorithm ends; otherwise, the prun Set is reprocessed
(3)	Judge whether the prun Set is empty. If it is, the algorithm ends. Otherwise, traverse the prun Set
(4)	Traverse each piece of data in the prun Set in the data frame set data Set
(5)	From each data frame in the data Set, the strings with the same position and length as the data are intercepted and compared. If any substring matches successfully, it will be added to single Map
(6)	Whether the data Set traversal is finished. If not, return to step 4. Otherwise, the data in single Map will be removed and included
(7)	Add the data in single Map to feature Map
(8)	Judge whether the traversal of prun Set is finished. If not, return to step 3. Otherwise, filter the support of feature Map
(9)	Deduplicate feature Map
(10)	In the end, the feature Map is output as the final frequent item set

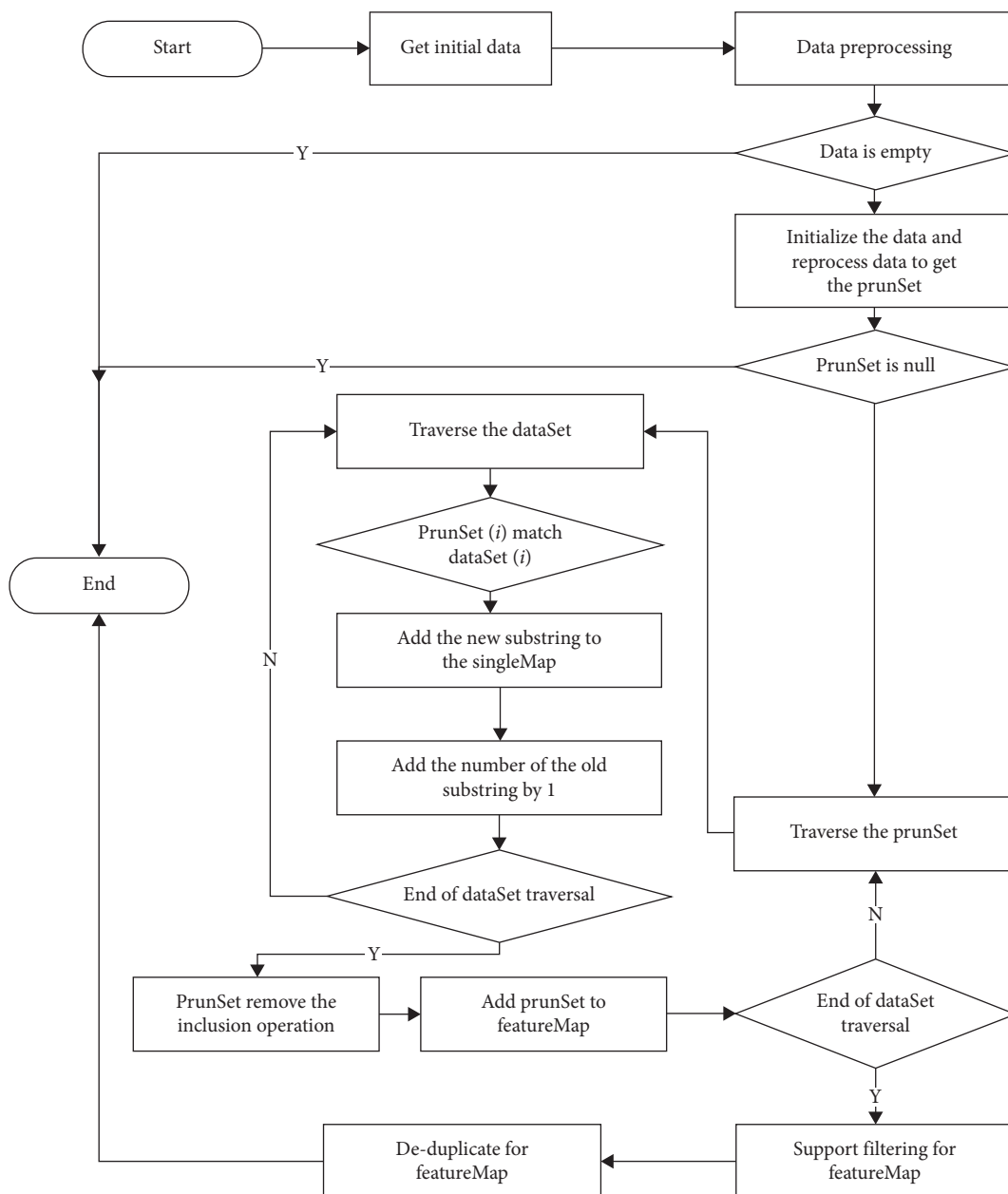


FIGURE 1: Alerts filter and identification model.

single Map to the string “0010110#406”, so that the statistics are complete. Similarly, for postinclusion, the number of times the longer substring is in the single Map should be added to the other substring. For mutual inclusion, we need to intercept the mutual inclusion part of two strings plus position information to form a new substring and add the number of times both are in the single Map to the new substring.

Before handling these three cases, the single Map is copied to tmp Single Map (a collection of substrings of the temporary ideal string), and either adding times or adding new strings is done in tmp Single Map, so the single Map needs to be updated after processing.

3.2.3. Get Frequent Substrings. After all ideal strings in the pruned Set are subjected to item set acquisition and inclusion removal operations, each substring and corresponding count in the respective single Map of each ideal string is added to the feature Map (feature string set).

Then, the support is calculated for each substring in single Map, and all substrings with support less than Min_Sup are deleted.

For example, consider a case where both the string “00001110100110#153” and the string “01110100110#153” have support greater than the minimum support. Neither of them will be removed, but it is obvious that for strings in the same position, only the longer ones should be left.

At this point, the final set of frequent items of data Set, a collection of data frames, has been obtained.

The association rule generation, however, still follows the association rule analysis method of the Apriori algorithm.

3.3. Algorithm Evaluation. Evaluating the merits of an algorithm requires several perspectives. The most common means is to calculate the time complexity and space complexity of the algorithm.

3.3.1. Time Complexity. Suppose the data Set has n data frames and the average length of the data frames is m . Then, first iterate through the data Set to initialize the vector Vector with $O(mn)$ time complexity. The time complexity is $O(m)$ to obtain the ideal string set pruned Set by Vector. Then, the length of all the ideal strings in pruned Set does not exceed m . For each ideal string of pruned Set to compare with data Set and get substrings, the time complexity of this operation is $O(mn)$. Overall, the final time complexity of the algorithm is $O(mn)$. This also shows the superiority of the new algorithm.

3.3.2. Spatial Complexity. Assuming that the average length of data frame is m , the one-dimensional vector Vector is initialized with Max_len, and the length of Max_len is taken as the average length m , so the one-dimensional vector Vector has m elements, all operations are based on the Vector obtained from the initial data preprocessing work, and the subsequent data reprocessing operations are all for the ideal string cross-matching work. Therefore, the space of

all operations after data preprocessing does not exceed the Vector, so the space complexity of the algorithm is $O(m)$.

4. Analysis of Experimental Results

The content focuses on testing the OFS algorithm to ensure the correctness of the algorithm. And the OFS algorithm is compared with the CFI algorithm to derive the correctness and superiority of the optimization direction of the OFS algorithm.

4.1. Support and Coverage Testing. This step focuses on testing the algorithm by two means. The first one is the extraction of frequent substrings from the OFS algorithm using the set of data frames, and then the extraction results are taken out for separate check counts, thus testing the correctness of the OFS algorithm for extracting frequent substrings in terms of support counts. In the second test, the OFS algorithm is compared with the CFI algorithm implemented to extract frequent substrings from the same set of data frames, and the results of the two algorithms are compared to see if the frequent item sets of the two algorithms are the same in number and correspond to each other. This further tests the correctness of the OFS algorithm in terms of the range and support of the extracted frequent substrings.

As shown in Table 2, the data shows the comparison of the frequent item set extraction results for DNS protocols using the two matching methods, and it is obvious from the corresponding entries that the algorithm results are consistent with the test results of the brute force method.

As shown in Table 3, the data shows the comparison of the frequent item set extraction results for the HTTP protocol using the two methods, and it is obvious from the corresponding entries that the algorithm results are consistent with the test results of the brute force method.

The test results from the comparison of the two sets of tables show that the OFS algorithm has the same results as after the brute force search. This indicates that the OFS algorithm possesses some correctness in counting the support of frequent substrings.

As shown in Table 4, the data shows the comparison of frequent item set extraction results for both protocols using the CFI algorithm; it can be seen that using the same data for CFI algorithm testing, the OFS algorithm and CFI algorithm extract the exact same frequent item set under the same condition of the data frame set, which shows that the coverage of OFS algorithm in terms of frequent substring acquisition is comprehensive and once again correct in terms of support counting.

4.2. Algorithm Time Comparison Analysis. We try to demonstrate whether the OFS algorithm has an advantage over the CFI algorithm in terms of recognition speed by comparing the time used for feature extraction using both OFS and CFI algorithms for seven different data frames. These seven data frames are the data sets of seven common communication protocols. The size of the ICMP protocol file

TABLE 2: DNS protocol frequent item set results comparison.

Matching program	Frequent item set	Quantity (pieces)
OFS algorithm matching search results	000000000000000#160	24335
	000000000000000#398	17248
	000000000000000#416	19914
	00#399	18847
	00#361	19084
	00001000100#183	19580
	00010001000#184	17182
	010000000#302	17565
	010000100000000000010001010000000000000000000#94	19603
	100000000#303	18269
Violent match search results	000000000000000#160	24335
	000000000000000#398	17248
	000000000000000#416	19914
	00#399	18847
	00#361	19084
	00001000100#183	19580
	00010001000#184	17182
	010000000#302	17565
	010000100000000000010001010000000000000000000#94	19603
	100000000#303	18269

TABLE 3: HTTP protocol frequent item set results comparison.

Matching program	Frequent item set	Quantity (pieces)
OFS algorithm matching search results	000000000000000#160	24335
	000000000#381	11511
	000000000000000#416	10701
	0000000000000001#162	9725
	000001100#184	10135
	00001010011001100000101101111000000000#240	10177
	010000000000000#160	10239
	0101000000#368	13688
Violent match search results	000000000000000#160	24335
	000000000#381	11511
	000000000000000#416	10701
	0000000000000001#162	9725
	000001100#184	10135
	00001010011001100000101101111000000000#240	10177
	010000000000000#160	10239
	0101000000#368	13688

is 800 kB, QICQ protocol file is 20015 kB, DNS protocol file is 9264 kB, and SSDP protocol file is 6889 kB. The specific size of each protocol and the running time of both algorithms are detailed in Tables 5 and 6. All seven protocol data sets are intercepted by Wireshark and both algorithms are performed in conducted in CodeBlocks, and the runtimes are derived from the execution times of the console programs.

As shown in Table 6, we can see that the OFS algorithm has a significant advantage in speed compared to the CFI algorithm, but the CFI algorithm in the SSDP protocol file has a recognition time of 2095.6s and combined with the overall data in Table 6 to compare, it is clear that the CFI algorithm for the SSDP protocol has too long a recognition time, so this time is defined as bad data. The running time of the two groups of algorithms is compared as a line graph,

and the difference between the two can be seen more clearly. This is shown in Figure 2.

4.3. Algorithm Accuracy Comparison Analysis. Figure 3 shows the comparison results of the three algorithms for different protocols after the accuracy test, respectively. We know from Section 4.2 that the running time of the OFS algorithm is greatly shortened compared to the CFI algorithm, but it can be seen from Figure 3 that the accuracy of the OFS algorithm is still close to the CFI algorithm, so it can be seen that the OFS algorithm has a considerable advantage when performing unknown protocol analysis.

Figures 4 and 5 show the experimental plots comparing the *F1* values and accuracy of the OFS algorithm with the Relim algorithm and the FP growth algorithm.

TABLE 4: DNS protocol and HTTP protocol frequent item set extraction result table.

Matching program	Frequent item set	Quantity (pieces)
DNS protocol	0000000000000000#160	24335
	0000000000000000#398	17248
	0000000000000000#416	19914
	000#399	18847
	000#361	19084
	00001000100#183	19580
	00010001000#184	17182
	0100000000#302	17565
	010000100000000000010001010000000000000000000#94	19603
	1000000000#303	18269
HTTP protocol	0000000000000000#160	24335
	00000000#381	11511
	0000000000000000#416	10701
	00000000000000001#162	9725
	000001100#184	10135
	00001010011001100000101101111000000000#240	10177
	0100000000000000#160	10239
	0101000000#368	13688

TABLE 5: Protocol data set details.

Protocol type	Total number of data frames (pieces)	Total data frame size (kB)
SSDP protocol	11687	6889
QICQ protocol	47812	20015
SSDP protocol	1705	800
ICMP2 protocol	9056	40642
HTTP protocol	9056	43337
HTTP2 protocol	9651	4337
DNS protocol	24340	9364
Training set	125117	126978

TABLE 6: OFS algorithm and CFI algorithm running time comparison.

File size (kB)	6889	20015	800	6031	40632	43337	9264
CFI algorithm time (s)	2095.6	391.7	9.2	160.2	502.6	5730.4	77.0
OFS algorithm time (s)	0.5	3.7	0.4	1.2	15.9	17.8	1.6

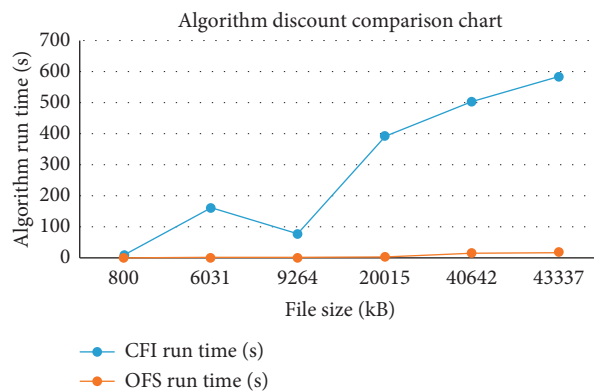


FIGURE 2: Algorithm running timeline comparison chart.

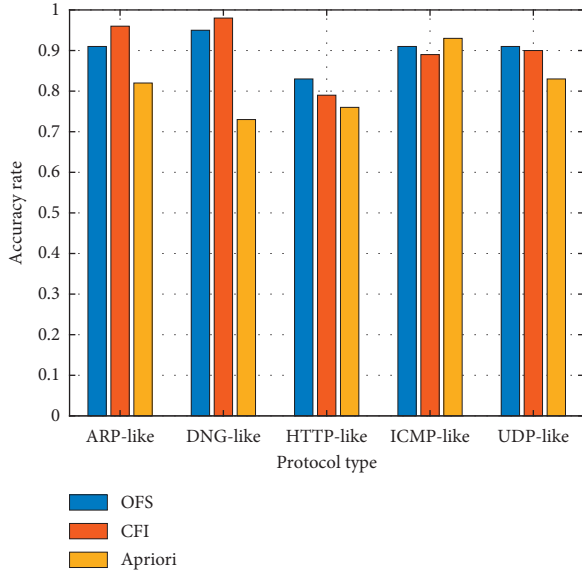


FIGURE 3: Accuracy comparison bar chart of the old algorithm.

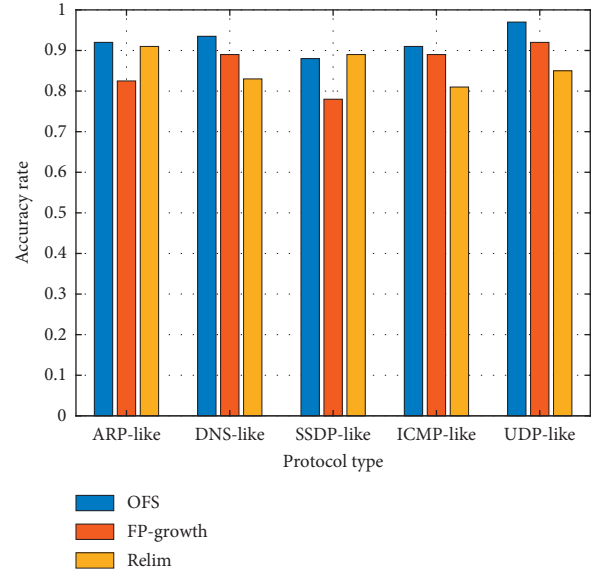


FIGURE 5: Comparison bar chart of F1 values for the algorithm.

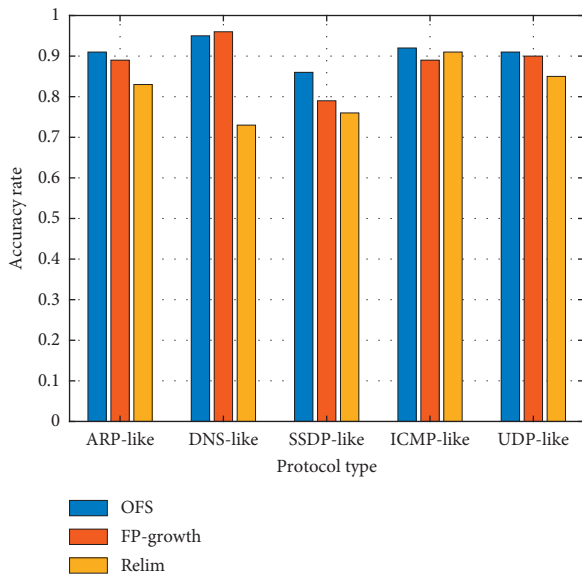


FIGURE 4: Accuracy comparison bar chart of the algorithm.

As can be seen from the accuracy comparison plot in Figure 4, the OFS algorithm is more stable than the other algorithms. From the $F1$ value comparison plot in Figure 5, it can be seen that the OFS algorithm has a slightly higher performance evaluation than the FP growth algorithm and the REIM algorithm, which is about 4% higher than the FP growth algorithm.

4.4. Support and Similarity Tests. The OFS algorithm is embedded into an unknown protocol syntax inverse analysis system to detect the effect of the size of support on the number and length of feature strings in a protocol by performing feature extraction tests with different support degrees on a set of protocol data sets by the OFS algorithm.

TABLE 7: Comparison of feature extraction of the OFS algorithm with different support degrees.

Support level	Feature string
0.6	11111111#10.***6814
	0100001010011001100#222.***10802
	00000000#258.***6814
0.7	0100001010011001100#302.***8627
	0100001010011001100#222.***10802
0.8	0100001010011001100#302.***8627
	0100001010011001100#222.***10802
	00001010011001100#304.***10802

TABLE 8: Similarity comparison of 10 ARP protocol data frames with different support degrees.

Protocol type	Similarity
ARP (0.6)	4.4
ARP (0.7)	6.67
ARP (0.8)	7

TABLE 9: Similarity comparison of 100 ARP protocol data frames with different support degrees.

Protocol type	Similarity
ARP (0.6)	32.8
ARP (0.7)	45.33
ARP (0.8)	50

TABLE 10: Similarity comparison of 1000 ARP protocol data frames with different support degrees.

Protocol type	Similarity
ARP (0.6)	412.8
ARP (0.7)	556
ARP (0.8)	622

Table 7 shows the details of the feature strings obtained after feature extraction by the OFS algorithm for the same ARP protocol data set with support degrees of 0.6, 0.7, and

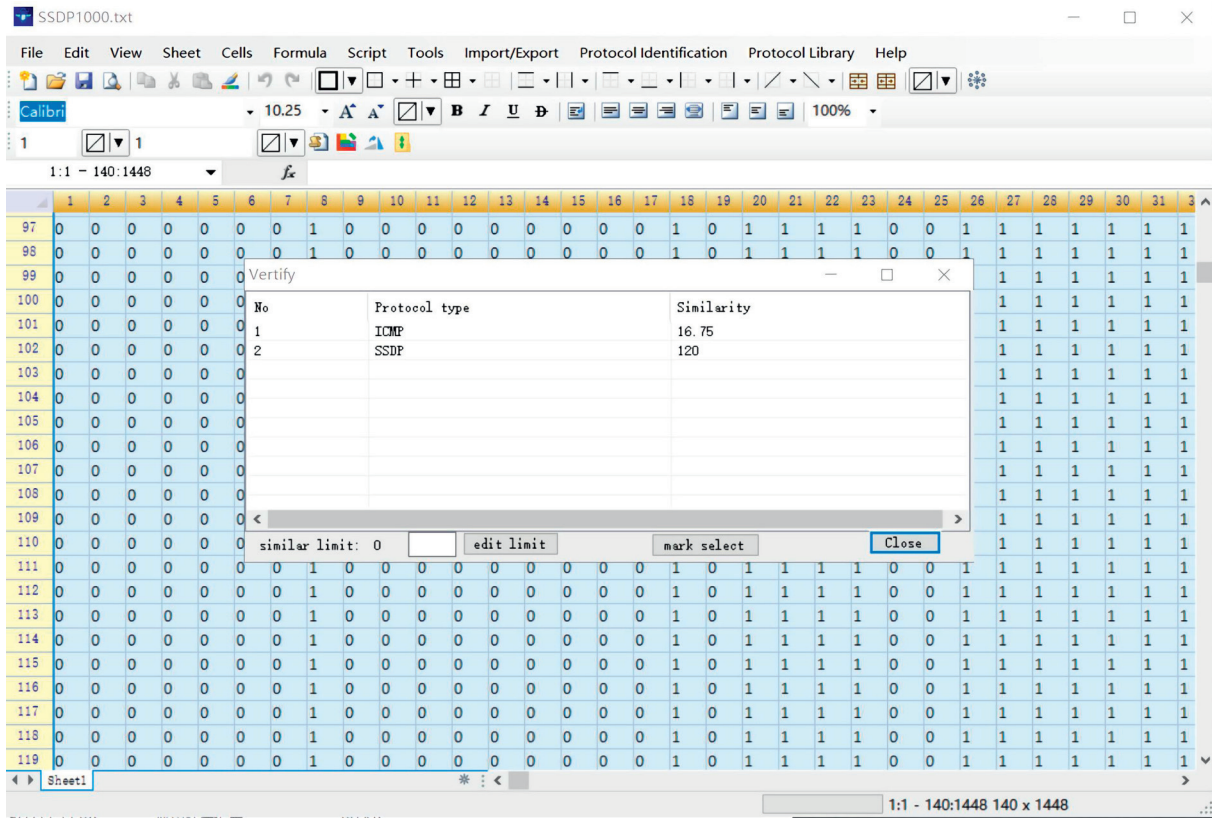


FIGURE 6: System main interface diagram.

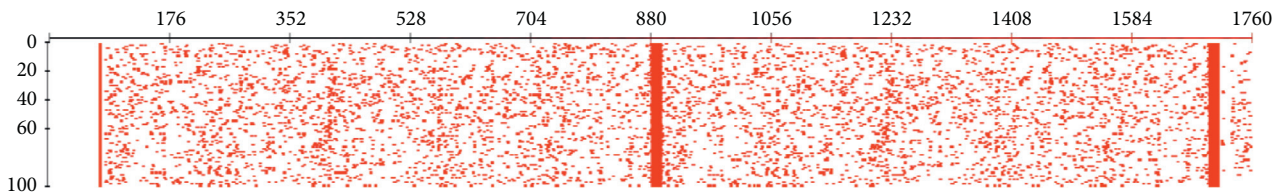


FIGURE 7: String color marker export chart.

0.8, respectively. From Tables 7, it can be seen that when the support degree is 0.6, the protocol syntax inverse analysis system extracts 4 feature strings after feature extraction, and when the support degree is 0.7 and 0.8, the system can only extract 2 feature strings, and the feature strings at the support degree of 0.8 are shorter than those at the support degree of 0.7. Therefore, it can be concluded that the number of feature strings will gradually decrease as the support degree increases, and the length of the feature strings will also become shorter. Therefore, it can be concluded that the number of feature strings decreases as the support increases and the length of the feature strings becomes shorter, which proves that the core idea of the OFS algorithm is correct and can yield the expected results.

Tables 8–10 show that the corresponding feature strings are extracted by the OFS algorithm with different support degrees and then used to identify and match the protocols to

obtain the matching similarity. In Tables 8–10, 10, 100, and 1000 data frames are randomly selected from 10802 data frames of the ARP protocol data set for similarity testing, and the results are shown in Tables 8–10. This also proves that the core idea of the OFS algorithm is correct and can achieve the expected results.

5. Experimental Development Configuration

The operation of the OFS algorithm relies on the Unknown Bitstream Protocol Intelligent Reverse Analysis System to implement the system, which has integrated features including piecewise import and export, protocol analysis, known protocol libraries, and some other essential modules. Data import and export include opening MAT format files, opening binary TXT files, opening Wireshark files, saving MAT format files, and exporting PNG format files.

5.1. Development Environment. The prototype system development and implementation environment for intelligent reverse analysis of BitTorrent protocol syntax is configured as follows:

Operating system: Windows 10 Inter(R) Core(TM)
CPU 64-bit operating system.

Memory: 16.0 GB

Debugging environment: Microsoft Visual Studio 2019.

Development language: C# language.

Protocol analysis tool: Wireshark.

5.2. Data Source. The experimental data set source of this system is divided into two main parts. The first part is the real-time data frames captured using the Wireshark tool, and the data frames are classified and saved in pcap format and TXT text format.

5.3. System Experiment Interface. The main interface of the system is shown in Figure 6, and its foreground display is an Excel-like display control.

The feature mining function of the system relies on the core idea of the OFS algorithm. The system first converts the hexadecimal strings of the protocol data set into binary strings and then displays them in the main interface, and then the feature mining module calls the OFS algorithm embedded in it to read the feature strings of the protocol data set and introduces the classical association rule mining algorithm to analyze the relationship between items in the frequent item analysis results. The frequent substrings with the lowest recognition rate are removed. Finally, the more discriminative results are stored in the protocol feature library.

The protocol identification module mainly includes two parts: protocol type determination and marking protocol features. Protocol type determination mainly relies on the protocol features generated by feature mining. When matching, the system compares the selected protocol data set with the features of each protocol in the feature library, calculates the similarity based on the number of features that can be matched to each protocol, and outputs the protocol types that satisfy the threshold value.

In the protocol type determination result, select the protocol type you want to view, and you can color the selected data set with the protocol feature marker so that you can view different protocol feature distributions according to different types. For example, after marking all "1111" strings in the file and exporting to PNG, the result is shown in Figure 7.

6. Conclusion

In this paper, we analyze the security risks of network communication in today's high-speed development of IoT technology and propose a reverse analysis method for protocol feature extraction and identification, which first extracts the feature strings from the protocol data frames and deposits

them in the protocol library, mainly for the purpose of matching with the new unknown data. The main purpose of this method is to compare and match with the new unknown data frames to identify the true identity of the unknown protocol and to achieve the purpose of maintaining the security of communication between IoT devices. The method is named the OFS algorithm, which is born by improving the existing Apriori algorithm and combining it with the idea of finding feature strings in the CFI algorithm. Combining the advantages of previous algorithms, the OFS algorithm can extract the frequent items set in the protocol data set more efficiently. Experimental results show that the OFS algorithm has a good improvement in the accuracy and speed of protocol identification and greatly improves the efficiency of the algorithm based on the original CFI algorithm, which has a good effect in the field of reverse identification of protocols.

Data Availability

All the data and methods have been presented in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research work was supported by the National Joint Funds of China (U20B2050), National Key R&D Program of China (2018YFB1201500), National Natural Science Funds of China (62072368, 61773313, and 61702411), and Key Research and Development Program of Shaanxi Province (2020GY-039, 2021ZDLGY05-09, 2017ZDXMGY-098, and 2019TD-014).

References

- [1] V. G. Semin, E. R. Khakimullin, A. S. Kabanov, and A. B. Los, "Problems of information security technology the "internet of things"" in *Proceedings of the 2017 International Conference "Quality Management, Transport and Information Security Information Technologies"*, pp. 110–113, Saint Petersburg, Russia, September 2017.
- [2] W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf, and Y. A. Bangash, "An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10250–10276, 2020.
- [3] Y. Yang, W. Zhang, F. Dang, L. Yan, and H. Liang, "Research on computer network information security and protection strategy based on internet of things," in *Proceedings of the 2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, pp. 688–691, Chongqing, China, November 2020.
- [4] A. M. Mohamed and Y. A. M. Hamad, "IoT security: review and future directions for protection models," in *Proceedings of the 2020 International Conference on Computing and Information Technology (ICCIT-1441)*, pp. 1–4, Tabuk, Saudi Arabia, September 2020.
- [5] F. Ni, J. Wei, and J. Shen, "An internet of things (IoTs) based intelligent life monitoring system for vehicles," in *Proceedings of the 2018 IEEE 3rd Advanced Information Technology*,

- Electronic and Automation Control Conference (IAEAC)*, pp. 532–535, Chongqing, China, October 2018.
- [6] H. Chen, “Application of internet of things technology in ship’s personal life,” in *Proceedings of the 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC)*, pp. 1318–1321, Sanya, China, June 2017.
 - [7] S. Yury and E. Samoylova, “The internet of things as socio-technological institution of civil society in post-informational era,” in *Proceedings of the 2017 2nd International Conference on Computer and Communication Systems (ICCCS)*, pp. 142–145, Krakow, Poland, July 2017.
 - [8] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, “Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles,” *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
 - [9] O. Al-Mahmud, K. Khan, R. Roy, and F. Mashuque Alamgir, “Internet of things (IoT) based smart health care medical box for elderly people,” in *Proceedings of the 2020 International Conference for Emerging Technology (INCET)*, pp. 1–6, Belgaum, India, June 2020.
 - [10] S. S. Mishra and A. Rasool, “IoT health care monitoring and tracking: a survey,” in *Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1052–1057, Tirunelveli, India, April 2019.
 - [11] M. A. Mahmud, K. Bates, T. Wood, A. Abdelgawad, and K. Yelamarthi, “A complete internet of things (IoT) platform for structural health monitoring (SHM),” in *Proceedings of the IEEE 4th World Forum on Internet of Things (WF-IoT)*, pp. 275–279, Singapore, February 2018.
 - [12] J. Xiong, R. Ma, L. Chen et al., “A personalized privacy protection framework for mobile crowdsensing in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.
 - [13] Y. Xu, J. Liu, Y. Shen, J. Liu, X. Jiang, and T. Taleb, “Incentive jamming-based secure routing in decentralized internet of things,” *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 3000–3013, 2021.
 - [14] Y. Xu, J. Liu, Y. Shen, X. Jiang, Y. Ji, and N. Shiratori, “QoS-aware secure routing design for wireless networks with selfish jammers,” *IEEE Transactions on Wireless Communications*, p. 99, 2021.
 - [15] W. Wang, X. Zhang, L. Dong, Y. Fan, X. Diao, and T. Xu, “Network attack detection based on domain attack behavior analysis,” in *Proceedings of the 2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 962–965, Chengdu, China, October 2020.
 - [16] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, “An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.
 - [17] Mulyadi and D. Rahayu, “Indonesia national cybersecurity review: before and after establishment national cyber and crypto agency (BSSN),” in *Proceedings of the 2018 6th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1–6, Parapat, Indonesia, August 2018.
 - [18] M. R. Egas, G. Ninahualpa, D. Molina, M. Ron, G. Ninahualpa, and J. Diaz, “National cybersecurity strategy for developing countries: case study: Ecuador proposal,” in *Proceedings of the 2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–7, Seville, Spain, June 2020.
 - [19] J. Xiong, J. Ren, L. Chen et al., “Enhancing privacy and availability for data clustering in intelligent electrical service of IoT,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2019.
 - [20] Y. Hu, L. Pang, Q. Pei, and X. A. Wang, “Analyze network protocol’s hidden behavior,” in *Proceedings of the 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pp. 403–406, Krakow, Poland, November 2015.
 - [21] L. Gergo, “Message format and field semantics inference for binary protocols using recorded network traffic,” in *Proceedings of the 2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–6, Split, Croatia, September 2018.
 - [22] B. D. Sija, Y. Goo, K. Shim, S. Kim, M. Choi, and M. Kim, “Survey on network protocol reverse engineering approaches, methods and tools,” in *Proceedings of the 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 271–274, Seoul, South Korea, September 2017.
 - [23] T. Gu, A. Abhishek, H. Fu, H. Zhang, D. Basu, and P. Mohapatra, “Towards learning-automation IoT attack detection through reinforcement learning,” in *Proceedings of the 2020 IEEE 21st International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM)*, pp. 88–97, Cork, Ireland, September 2020.
 - [24] Iana: <http://www.iana.org/assignments/port-numbers.?18?>.
 - [25] H. Seo and D. Cho, “A new alignment free genome comparison algorithm based on statistically estimated feature frequency profile,” in *Proceedings of the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4265–4268, Jeju Island, Republic of Korea, July 2017.
 - [26] R. Singh, D. Rai, R. Prasad, and R. Singh, “Similarity detection in biological sequences using parameterized matching and Q-gram,” in *Proceedings of the 2018 Recent Advances on Engineering, Technology and Computational Sciences (RAETCS)*, pp. 1–6, Allahabad, India, February 2018.
 - [27] M. H. Neamatollahi and M. Naghibzadeh, “Simple and efficient pattern matching algorithms for biological sequences,” *IEEE Access*, vol. 8, pp. 23838–23846, 2020.
 - [28] Y. Tian, Z. Wang, J. Xiong, and J. Ma, “A blockchain-based secure key management scheme with trustworthiness in DWSNs,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
 - [29] A. Wichmann and S. Schupp, “Matching machine-code functions in executables within one product line via bio-informatic sequence alignment,” in *Proceedings of the IEEE 5th Workshop on Mining Unstructured Data (MUD)*, pp. 1–5, Bremen, Germany, May 2015.
 - [30] L. Zhang, *Research on Feature Extraction and Identification Method of Bitstream Protocol*, Xi’an University of Technology, Xi’an, China, 2019.
 - [31] X. Hei, B. Bai, Y. Wang, L. Zhang, L. Zhu, and W. Ji, “Feature extraction optimization for bitstream communication protocol format reverse analysis,” in *Proceedings of the 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, pp. 662–669, Rotorua, New Zealand, August 2019.
 - [32] W. Wang, B. Bai, Y. Wang, X. Hei, and L. Zhang, “Bitstream protocol classification mechanism based on feature extraction,” in *Proceedings of the 2019 International Conference on*

- Networking and Network Applications (NaNA)*, pp. 241–246, Daegu, Republic of Korea, October 2019.
- [33] D. U. Youxiang, W. Li-fa, H. Zheng, and P. Fan, “A semi-automatic protocol reverse method based on message sequence analysis,” *Computer Engineering*, vol. 38, no. 19, pp. 277–280, 2012.
- [34] M. Xiao-Li and Z. Xiao-Lei, “The application of data mining technology in computer network security,” in *Proceedings of the 2015 7th International Conference on Measuring Technology and Mechatronics Automation*, pp. 1126–1129, Nanchang, China, June 2015.
- [35] Y.-H. Goo, K.-S. Shim, U.-J. Baek, J.-T. Park, M.-G. Shin, and M.-S. Kim, “An automatic protocol reverse engineering approach from the viewpoint of the TCP/IP reference model,” in *Proceedings of the 2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 43–48, Daegu, Republic of Korea, September 2020.
- [36] L. Cai, R. Shi, and D. Xu, “Communication protocol identification based on data mining and automatic reasoning,” in *Proceedings of the IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pp. 211–216, Beijing, China, March 2017.
- [37] Z. Jie and L. Jianping, “Feature identification of unknown protocol,” in *Proceedings of the 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 147–149, Chengdu, China, December 2016.
- [38] M. Karimov, K. Tashev, and S. Rustamova, “Application of the Aho-Corasick algorithm to create a network intrusion detection system,” in *Proceedings of the 2020 International Conference on Information Science and Communications Technologies (ICISCT)*, pp. 1–5, Karachi, Pakistan, February 2020.
- [39] J. Sivapriya, R. Roy, M. Biswas, and S. Mandal, “Comparative study of APRIORI and FP algorithm for decision making,” in *Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1058–1061, Tirunelveli, India, April 2019.

Research Article

Improved Outsourced Provable Data Possession for Secure Cloud Storage

Haibin Yang,¹ Zhengge Yi ,¹ Ruifeng Li,¹ Zheng Tu,¹ Xu An Wang ,^{1,2} Yuanyou Cui,¹ and Xiaoyuan Yang¹

¹Engineering University of People's Armed Police, Xi'an, China

²Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang, China

Correspondence should be addressed to Xu An Wang; wangxazjd@163.com

Received 1 May 2021; Accepted 8 July 2021; Published 23 July 2021

Academic Editor: Qing Yang

Copyright © 2021 Haibin Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the advent of data outsourcing, how to efficiently verify the integrity of data stored at an untrusted cloud service provider (CSP) has become a significant problem in cloud storage. In 2019, Guo et al. proposed an outsourced dynamic provable data possession scheme with batch update for secure cloud storage. Although their scheme is very novel, we find that their proposal is not secure in this paper. The malicious cloud server has ability to forge the authentication labels, and thus it can forge or delete the user's data but still provide a correct data possession proof. Based on the original protocol, we proposed an improved one for the auditing scheme, and our new protocol is effective yet resistant to attacks.

1. Introduction

Since 2007, as one of the most interesting topics in the computer field, cloud computing has experienced rapid development and has become a key research direction for large-scale enterprises and institutions. Its high flexibility, scalability, high performance ratio, and other characteristics make it serve storage, medical, financial, education and other aspects [1–3]. Among them, cloud storage is an emerging technology developing in cloud computing in terms of data storage [4]. Compared with traditional data storage methods, cloud storage has the advantages of high performance and low cost. Cloud storage uses data storage and data management as its basic functions, allowing users to connect at any location and store local data and information on the cloud, facilitating users' management of resources.

However, with the widespread application of cloud storage technology, its security has received more and more attention from users and has gradually become the key to the sustainable development of cloud storage technology. On the one hand, cloud service providers (CSPs) may delete users' data stored in order to free up storage space for their interests or may want to obtain users' data privacy [5]. On

the other hand, the CSP has great openness and complexity, and it is easy to become the central target of various malicious attacks, leading to the loss, leakage, tampering, or damage of users' data. Therefore, cloud storage integrity audits have emerged to solve the problem. Users regularly audit the integrity of their own data information stored in the cloud, discover whether their data have been discarded or tampered with, and take corresponding remedial measures.

1.1. Related Work. In the early years, cloud audit-related research was mainly about the integrity verification of remote data. Users do not own the original data and can only verify the integrity of the data stored on the cloud server through the protocol. In 2003, Deswarte et al. [6] proposed the first audit scheme that supports remote data integrity verification. The scheme is based on the Diffie–Hellman key exchange protocol using the homomorphic characteristics of RSA signatures and the difficulty of calculating discrete logarithms as a security basis. The entire file is represented by a large number and then subjected to modular exponentiation to achieve remote data integrity audit. However,

this solution will generate a great computing overhead, which is a heavy burden for users. In 2006, Filho et al. [7], based on RSA's homomorphic hash function, used the hash function to compress large data files into small hash values before performing operations. This scheme reduces the expense of calculation, but it is also not suitable for large-scale data storage in a cloud storage environment. The scheme put forward the important role of homomorphic hash function in remote data integrity verification, which is the biggest contribution of it. In 2008, Sebe et al. [8] based on the idea of partitioning to improve the previous scheme. The scheme divides the large data file into blocks and then each data block is calculated, which greatly reduces the computational expense. But the prover still needs to access all the data when generating the evidence, so this scheme is also not suitable for large data files.

The above schemes all require the user as a verifier to maintain a metadata set for verification. On the one hand, it is easy for users to lose or leak these metadata, which leads to the disclosure of private data. On the other hand, for users with limited computing resources, huge outsourcing data will increase the computing overhead in the audit process. In addition, in the event of a data corruption accident, the user or CSP will shirk each other's responsibilities and cannot provide effective evidence to confirm who should be responsible for the accident. Thus, scholars have introduced an absolutely impartial third-party auditor to audit on behalf of users. Auditors are more professional than users in terms of data preservation and computing performance, and in the event of an accident, they can be held accountable for solving problems in a fair manner. Therefore, the audit scheme has gradually changed from a private audit between users and CSP to a public audit between users, CSP, and third-party auditors (TPAs). In 2007, Shah et al. [9] proposed a public audit scheme based on the difficulty of discrete logarithm calculation to audit ciphertext data and key integrity. The scheme uses a hash function with a key to precalculate a certain number of response values stored by the auditor. During the audit process, the auditor only needs to match the evidence provided by the server with the prestored response value. However, the number of audits in this scheme will be limited by the number of prestored response values.

The amount of calculation required for the integrity audit of all data is not a small expense even for professional third-party auditors. Scholars have been studying how to increase audit efficiency to reduce computational overhead, but from another aspect, reducing the data content that needs to be audited can also achieve the goal. In 2005, Noar and Rothblum [10] proposed an online memory detection scheme. The scheme studied the sublinear authentication and proposed related authentication protocols. The basic idea of the sublinear authentication is to verify the integrity of all the original data by verifying the integrity of a small part of the data block specified randomly. In 2007, Ateniese et al. [11] proposed the first probabilistic provable data possession (PDP) auditing scheme with both safety and practicality. The scheme is based on RSA's homomorphic authentication label, which realizes the audit of outsourced data. The metadata of multiple data blocks can be aggregated

into one value, which effectively reduces the communication overhead, and a random sampling strategy is adopted to check the user's remote data instead of verifying all the user's data, so the calculation cost is effectively reduced.

With the continuous improvement of the audit program, some expansion requirements are constantly raised, for example, audit programs that support privacy protection or batch audits. In 2010, Wang et al. [12] proposed an audit scheme supporting privacy protection through the integration of homomorphic authentication tags and random mask technologies for the first time, in which the bilinear signature is used to support batch audits. In 2013, Yang et al. [13] proposed an audit solution based on the index table technology that supports dynamic data update, and the tag aggregation technology is used to process multiple audit requests from different users to support batch audits in a multi-user multi-cloud environment. In 2015, Hui et al. [14] proposed a public audit scheme based on dynamic hash table (DHT), which can record the attribute information of data blocks to support dynamic data update and improve efficiency. The program also supports privacy protection and batch auditing.

In 2007, Juels and Jr [15] proposed an original proof of retrievability (POR) scheme. Different from the PDP scheme described above, the POR scheme can repair the corrupted data when data are detected to be damaged. The scheme uses sampling and error correction codes to perform fault-tolerant preprocessing on outsourced data files, which can restore data with a certain probability when the data are damaged. In 2008, Shacham and Waters [16] proposed a compact POR scheme. The scheme draws on the idea of homomorphic authentication tags and effectively aggregates the evidence into a smaller value, allowing the verifier to perform any number of audits, while also reducing the communication overhead in the verification process. POR and PDP have their own application scenarios. The former can recover damaged data, and the latter is more flexible which can be applied to privacy protection, dynamic auditing, and batch auditing. Cloud auditing schemes are constantly being improved based on users' needs. When scholars are studying how to reduce computing and communication costs, they also try to expand functions horizontally or combine them with different technologies for innovation. In 2013, Zhao et al. [17] proposed the first identity-based cloud audit scheme, which uses random mask technology to achieve privacy protection. In the identity-based cloud auditing scheme, only the private key generator (PKG) holds a public-private key pair and its public key certificate. The public keys of other users can be calculated based on the identity information, and the private key is generated by PKG, which will reduce the calculation and communication overhead of the scheme. In 2015, Zhang and Dong [18] proposed the first certificateless cloud audit scheme that can resist malicious auditors. Thus, the concept of malicious auditors was introduced into the cloud audit program for the first time. The certificateless cloud audit scheme can solve the certificate management problem in the certificate-based cloud audit solution and the key escrow problem in the identity-based audit solution. In 2016, Xin

et al. [19] combined the transparent watermarking technology with the auditing scheme, proposing a scheme to audit the integrity of static multimedia data, which can greatly save multimedia data calculation and storage costs.

1.2. Our Contribution. Recently, an outsourced dynamic provable data possession scheme with batch update for secure cloud storage (ODPDP) was proposed by Wei et al. [20]. However, we find that there are security problems in their scheme. The adversary can easily forge authentication labels. Even if all the outsourced data have been deleted by the cloud server, CSP can still give a correct data possession proof. And malicious auditors do not carry out auditing work but can conspire with the cloud server forging audit log to deceive client. Finally, we propose an improved secure auditing protocol, and roughly analysis shows that our new protocol is secure and can be used in practical settings.

1.3. Organization. This paper is organized as follows. In Section 2, we describe the system model of our scheme. In Section 3, we review Guo et al.'s outsourced dynamic provable data possession scheme with batch update for secure cloud storage. In Section 4, we give our attacks to the original scheme to show that it is not secure. In Section 5, we give our improved secure auditing scheme and roughly analyze its security. Finally, in Section 6, we draw some conclusions.

2. System Model

First of all, for the convenience of understanding, the notations and their corresponding meanings of this paper are described in Table 1.

There are three entities in the system model of ODPDP scheme: CSP (cloud service provider), client, and auditor, as depicted in Figure 1. The following three entities are involved:

- (1) CSP (cloud service provider): the service provider, which has abundant computing power and physical storage capacity, realizes the maintenance and management of the received data from client. This part is honest and curious.
- (2) Client: the data owner, which outsources the data that needs to be calculated and stored to the CSP, concern the integrity of the outsourced data, and checks whether the auditor is honest in the audit work regularly.
- (3) Auditor: the third-party auditor accepts audit task from the client and is responsible for ensuring the integrity of the data of the client stored in CSP.

The protocols used in the ODPDP scheme are as follows:

- (1) *Setup* $(1^k) \longrightarrow \{\text{client}: sk_c, vk_c, sk, pk; \text{auditor}: sk_a, vk_a; \text{CSP}: sk_{CSP}, vk_{CSP}\}$: random key generation protocol. The users input a security parameter K and then it generates pairs of signing-verifying keys (SKP, VKP) for each participant. For the

convenience of expression, we assume that all the participants involved in each subsequent protocol always take the owners' public key and its own secret key as input.

- (2) *Store* (client: M) $\longrightarrow v\{\text{client}: P, C; \text{auditor}: P, CT; \text{CSP}: P, M\}$: the interactive protocol among the three parties. It takes the keys of the three participants as input and the data M owner by client, and then outputs the processed data $\mathcal{M} = \{M, \Sigma\}$ for the CSP. Σ is generated by the client through the secret key sk as the tag vector of M . And for the auditor, it outputs a RBMTT based on M . Besides, it outputs a public parameter P that is confirmed by three participants and a contract C between the client and the auditor.
- (3) *AuditData* (auditor: Q, T ; CSP: \mathcal{M}) $\longrightarrow \{\text{auditor}: dec_a, L\}$: the interactive protocol between the CSP and auditor to make the auditor to be sure that the integrity of M in CSP is in good condition. The auditor takes the functionality of Bitcoin to extract pseudo-random challenge Q and then sends it to the CSP. The CSP computes a proof of data possession based on Q and M and then sends it to the auditor for verification. The auditor verifies the proof from the CSP through Q, T, pk and then outputs a binary value dec_a as a response to indicate whether the auditor accepts the proof or not and a log entry L to record the auditing behavior.
- (4) *AuditLog* (client: B ; auditor: T, Λ) $\longrightarrow \{\text{client}: decc\}$: the interactive protocol between the client and auditor, which can help the client to audit a log file Λ consisting of the log entries recorded by the auditor. The aim of this protocol is to check whether the auditor accomplished the auditing task or not. After the auditor receives the random subset B of the Bitcoin block index released by the client, it calculates the proof of the specified log based on B, T, λ and sends the proof to the client. The client checks the received proof and then outputs a binary value $decc$, which indicates if it admits the proof. Compared with the AuditData protocol, the frequency is much lower and the computational efficiency is much higher in this protocol.

3. Review of Guo et al.'s Scheme

In Guo et al.'s scheme, three parties are involved, which are the user, the auditor, and the CSP. In their scheme, they used the rank-based Merkle tree (RBMT) to protect the integrity of data block hashes, while the hash values and tags protect the integrity of data blocks. Then, they proposed a multi-leaf-authenticated (MLA) solution for RBMT to authenticate multiple leaf nodes and their indices all together without storing status value and height value. At the same time, they proposed an efficient homomorphic verifiable tag (EHVT) based on BLS signature to reduce clients' log verification effort. For the specific implementation of these technologies, one can refer to the original paper [20]. Concretely, the following algorithms are involved in their scheme.

TABLE 1: Notations.

Notations	Descriptions
G	A multiplicative cyclic group
H	A secure hash function such that $H(\cdot): \{0, 1\}^* \rightarrow Z_p$
Z_p	A prime field
g	The generator of group G
M	The user's data file
$m\{m_{11}, \dots, m_{12}\}$	The user's data file with n blocks and s slices
α_i, x	The secret random values of user
λ	A random value that makes up the public key
sk	The secret key of user
pk	The public key of user
σ_i	The authentication label for the i -th data block
Σ	The collection of authentication labels
M	The processed data of user which include M and Σ
Q	The challenge set sent by the auditor to CSP
T	A rank-based Merkle tree built from user's data
Λ	The local log file of the auditor
L	A record of an auditor's auditing work
ρ	The proof generated by the CSP or auditor
\sqcup_p	A multi-proof based on multiple challenged leaf nodes

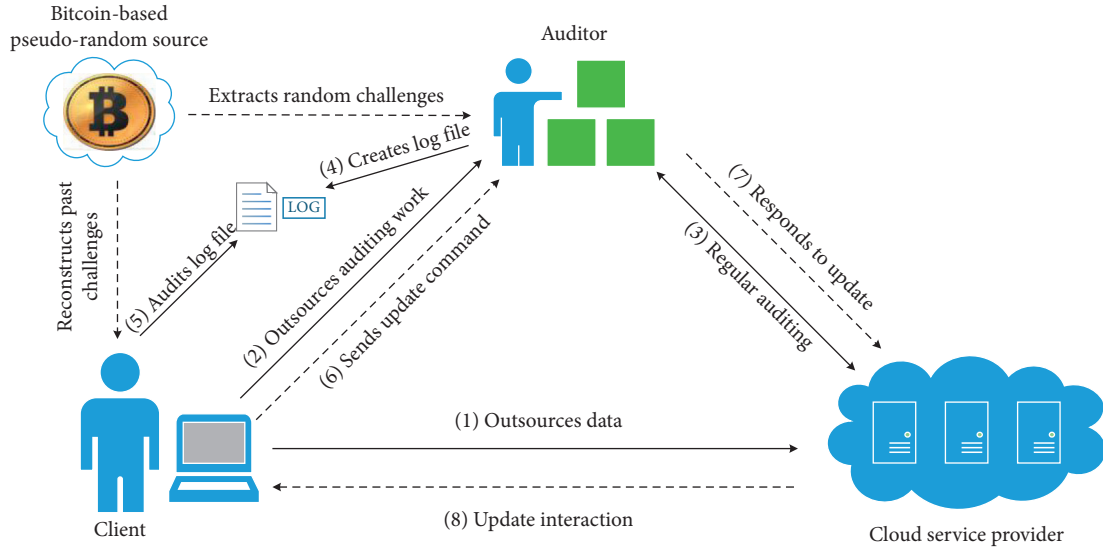


FIGURE 1: System model.

3.1. *Setup Protocol.* Each participant $P \in \{\text{CSP}, \text{client}, \text{auditor}\}$ performs Key Gen to obtain skP and vkP . In addition, the client samples $s + 1$ random elements $\alpha_1, \alpha_2, \dots, \alpha_s, x \in Z_q$ and computes $g_1 = g^{\alpha_1}, g_2 = g^{\alpha_2}, \dots, g_s = g^{\alpha_s}, y = g^x \in G$. Then, the client chooses a random element $\lambda \in G$, and the secret key and public key are denoted as $sk = (\alpha_1, \alpha_2, \dots, \alpha_s, x)$ and $pk = (g, \lambda, g_1, g_2, \dots, g_s, y)$.

3.2. *Store Protocol.* The data file is divided into $M = (m_1, m_2, \dots, m_n)$, and each data block consists of s sectors and has the form $m_i = m_{i1}m_{i2} \dots \| m_{is}$ ($1 \leq i \leq n$) such

that each sector $m_{iz} \in Z_q$ ($1 \leq z \leq s$), where $\|$ denotes concatenation.

Constructing RBMT. With all data blocks, the client first computes $h_i = H_2(m_i)$ ($1 \leq i \leq n$). Then, the client constructs RBMT T on top of the ordered hash values, meaning that each leaf node w_i stores the corresponding hash value h_i .

Computing EHVT. Based on g, λ and secret key sk , the client computes

$$\sigma_i = \left(\lambda^{h_i} \cdot g \sum_{z=1}^s \alpha_z m_{iz} \right)^x \in G \quad (1 \leq i \leq n). \quad (1)$$

Then, the client generates the processed data $\mathcal{M} = \{M, \Sigma\}$, where $\Sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$.

3.2.1. Outsourcing Data. The client sends \mathcal{M} and $\text{Sig}_{\text{skc}}(\mathcal{M})$ to CSP. CSP verifies $\text{Sig}_{\text{skc}}(\mathcal{M})$, and if the verification is passed, CSP accepts \mathcal{M} .

3.2.2. Outsourcing Auditing Work. Auditing work is outsourced to the auditor and CSP sends T with $\text{Sig}_{\text{skc}}(T)$ to the auditor. Then, the auditor verifies $\text{Sig}_{\text{skc}}(T)$.

3.2.3. Agreeing Parameters. A public parameter $\mathbb{P} = \{n, h_{\text{root}}\}$ needs to be agreed on by three participants, where n denotes the number of data blocks and h_{root} denotes the Merkle root of T . In addition, the client and auditor also need to agree on a contract $C = \{\text{BI}, F, l\}$, where C denotes the auditor's checking policy. The auditing work will start from a Bitcoin block index BI, the auditing frequency depends on F , and l dictates the number of challenged data blocks for each checking.

Then, the client deletes \mathcal{M} and T from its local storage, and she only maintains a constant amount of metadata.

3.3. AuditData Protocol. The scheme leverages the Bitcoin blockchain as a time-dependent pseudo-random source to generate periodic challenges. The auditor inputs the time $t \in \tau$ to obtain a hash value $\text{hash}^{(b)} \in \{0, 1\}^{\text{hash}}$ of the latest block that has appeared since time t in Bitcoin blockchain. Then, PRBG is invoked on the input $\text{hash}^{(b)}$ to acquire pseudo-random bits, which will be used by the auditor to select a pair of keys $k_{\pi}^{(b)}, k_f^{(b)}$. At last, the auditor generates a challenge $Q^{(b)} = \{b, k_{\pi}^{(b)}, k_f^{(b)}\}$ and sends it to CSP, where the block b corresponds to the time t .

Upon receiving the challenge $Q^{(b)}$, CSP first computes the challenged indices and coefficients as follows:

$$\begin{aligned} i_{\eta} &= \pi_{k_{\pi}^{(b)}}(\eta), \\ a_{\eta} &= f_{k_f^{(b)}}(\eta) \quad (1 \leq \eta \leq l). \end{aligned} \quad (2)$$

Then, CSP computes the proof of data possession to prove the integrity of the challenged data blocks as follows:

$$\begin{aligned} \mu_z^{(b)} &= \sum_{\eta=1}^l a_{\eta} m_{i_{\eta} z} \in \mathbb{Z}_q, \quad 1 \leq z \leq s, \\ \sigma^{(b)} &= \prod_{\eta=1}^l \sigma_{i_{\eta}}^{a_{\eta}} \in G. \end{aligned} \quad (3)$$

Finally, CSP responses the auditor with the proof $\rho^{(b)} = \{\mu_1^{(b)}, \mu_2^{(b)}, \dots, \mu_s^{(b)}, \sigma^{(b)}\}$. Then, the auditor verifies the correctness of $\rho^{(b)}$. First, the auditor computes the challenged indices and coefficients. Second, the auditor computes the value with T as follows:

$$h^{(b)} = \lambda \sum_{\eta=1}^l a_{\eta} h_{i_{\eta}} \in G. \quad (4)$$

Third, the auditor verifies the proof $\rho^{(b)}$ by checking the following equation:

$$e(\sigma^{(b)}, g) \stackrel{?}{=} e\left(h^{(b)} \cdot \prod_{z=1}^s g_z^{\mu_z^{(b)}}, y\right). \quad (5)$$

If the equation holds, the auditor assures that the challenged data blocks are intact. Lastly, the auditor saves the log entry in the log file Λ to record the auditing work as follows:

$$L^{(b)} = \{t, Q^{(b)}, h^{(b)}, \rho^{(b)}, \text{Sig}_{\text{skc}_{\text{CSP}}}(\rho^{(b)})\}. \quad (6)$$

3.4. AuditLog Protocol. The client chooses a random subset B of indices of Bitcoin blocks and sends it to the auditor. Once receiving B , the auditor finds $Q^{(b)}, h^{(b)}$, and $\rho^{(b)}$ from his log file Λ for each $b \in B$ and computes.

$$\begin{aligned} h^{(B)} &= \prod_{b \in B} h^{(b)} \in G, \\ \sigma^{(B)} &= \prod_{b \in B} \sigma^{(b)} \in G, \\ \mu_z^{(B)} &= \sum_{b \in B} \mu_z^{(b)} \in \mathbb{Z}_q, \quad 1 \leq z \leq s. \end{aligned} \quad (7)$$

In addition, for each $b \in B$, the auditor reads $k_{\pi}^{(b)}$ from $Q^{(b)}$ and computes the challenged indices i_{η} ($1 \leq \eta \leq l$) by

invoking $\pi_{k_{\pi}^{(b)}}(\eta)$. After eliminating the repetitive indices, the last ordered challenge index vector is denoted by

$C = (i_1, i_2, \dots, i_c)$. Then, the auditor obtains the corresponding multi-proof \sqcup_p . At last, the auditor generates the proof of appointed logs as follows:

$$\rho^{(B)} = \{\sqcup_p, h^{(B)}, \mu_1^{(B)}, \mu_2^{(B)}, \dots, \mu_s^{(B)}, \sigma^{(B)}\}, \quad (8)$$

and sends it to the client with $\text{Sig}_{sk_a}(\rho^{(B)})$.

After verifying $\text{Sig}_{sk_a}(\rho^{(B)})$, for each $b \in B$, the client first invokes PRBG($\text{hash}^{(b)}$) to get $Q^{(b)}$ and reconstructs the challenged indices and coefficients i_η, a_η ($1 \leq \eta \leq l$). Then, the client verifies the correctness of \sqcup_p . If the verification is passed, it means that all the challenged leaf nodes w_{i_j} ($1 \leq j \leq c$) in \sqcup_p are authenticated, and then the corresponding hash value h_{i_j} stored in leaf node w_{i_j} can be accepted by the client. Finally, with λ and all authenticated h_{i_j} , the client verifies $h(B)$ by checking the following equation:

$$h^{(B)} \stackrel{?}{=} \lambda \sum_{b \in B} \left(\prod_{\eta=1}^l a_\eta h_{i_\eta} \right). \quad (9)$$

If this verification passes, the client checks the last equation by using her secret key sk and the verified $h^{(B)}$ as follows:

$$\sigma^{(B)} \stackrel{?}{=} \left(h^{(B)} \cdot g \sum_{z=1}^s \alpha_z \mu_z^{(B)} \right)^x. \quad (10)$$

If the above equation holds, the client assures that the auditor audited CSP for all the past challenged data blocks appointed by B honestly. The correctness of equation can be elaborated as follows:

$$\begin{aligned} \sigma^{(B)} &= \prod_{b \in B} \prod_{\eta=1}^l \sigma_{i_\eta}^{a_\eta} \\ &= \prod_{b \in B} \prod_{\eta=1}^l \left(\lambda^{h_{i_\eta}} \cdot g \sum_{z=1}^s \alpha_z m_{i_\eta z} \right)^{a_\eta x} \\ &= \left(\prod_{b \in B} \lambda \sum_{\eta=1}^l a_\eta h_{i_\eta} \cdot g \sum_{z=1}^s \alpha_z \left(\sum_{\eta=1}^l a_\eta m_{i_\eta z} \right) \right)^x \\ &= \left(\lambda \sum_{b \in B} \left(\sum_{\eta=1}^l (a_\eta h_{i_\eta}) \right) \cdot g \sum_{z=1}^s \alpha_z \left(\sum_{b \in B} \mu_z^{(b)} \right) \right)^x \\ &= \left(h^{(B)} \cdot g \sum_{z=1}^s \alpha_z \mu_z^{(B)} \right)^x. \end{aligned} \quad (11)$$

4. Our Attack

In Guo et al.'s auditing protocol, their security model indicates that the malicious CSP cannot forge false proof to pass the challenger's verification and the client can resist malicious CSP and auditor collusion attacks. However, we find that we can extract some key information from the client's pk , data blocks, and their corresponding tags which are known to CSP. In this section, we firstly show how CSPs extract key information and how to use this information to forge "correct" data blocks and their corresponding tags. Then, we will show how malicious CSP and auditor collude to use false proof to pass the client's verification.

4.1. Attack I. Our attack is based on the following observation: the public key of the client is

$$pk = (g, \lambda, g_1, g_2, \dots, g_s, \gamma), \quad (12)$$

and this public key is known to all, and thus the adversary can easily use it to forge authentication label. Concretely, the adversary launches the following attack:

- (1) In the Store Protocol, CSP can receive M from client, which includes the data of the client and its corresponding authentication tags. The adversary can get a large number of authentication tags as follows:

$$\begin{aligned} \sigma_1 &= \left(\lambda^{h_1} \cdot g \sum_{z=1}^s \alpha_z m_{1z} \right)^x, \\ \sigma_2 &= \left(\lambda^{h_2} \cdot g \sum_{z=1}^s \alpha_z m_{2z} \right)^x, \\ &\vdots \\ \sigma_n &= \left(\lambda^{h_n} \cdot g \sum_{z=1}^s \alpha_z m_{nz} \right)^x. \end{aligned} \quad (13)$$

- (2) The above equations can be rewritten as follows:

$$\begin{aligned}
\sigma_1 &= (\lambda^x)^{h_1} \cdot (g^{\alpha_1 x})^{m_{11}} \cdot (g^{\alpha_2 x})^{m_{12}} \dots (g^{\alpha_s x})^{m_{1s}}, \\
\sigma_2 &= (\lambda^x)^{h_2} \cdot (g^{\alpha_1 x})^{m_{21}} \cdot (g^{\alpha_2 x})^{m_{22}} \dots (g^{\alpha_s x})^{m_{2s}}, \\
&\vdots \\
\sigma_n &= (\lambda^x)^{h_n} \cdot (g^{\alpha_1 x})^{m_{n1}} \cdot (g^{\alpha_2 x})^{m_{n2}} \dots (g^{\alpha_s x})^{m_{ns}}.
\end{aligned} \tag{14}$$

The CSP knows the data blocks of client, and it can calculate the corresponding hash values through $H(m_i)$:

$$h_1, h_2, \dots, h_n. \tag{15}$$

In order to simplify the attack process, let $s = 2$, $A = \lambda^x$, $B = g^{\alpha_1 x}$, $C = g^{\alpha_2 x}$, and take three linear irrelevant tags $\sigma_1, \sigma_2, \sigma_3$ as follows:

$$\begin{aligned}
\sigma_1 &= A^{h_1} \cdot B^{m_{11}} \cdot C^{m_{12}}, \\
\sigma_2 &= A^{h_2} \cdot B^{m_{21}} \cdot C^{m_{22}}, \\
\sigma_3 &= A^{h_3} \cdot B^{m_{31}} \cdot C^{m_{32}}.
\end{aligned} \tag{16}$$

(3) With these equations, the adversary can compute A , B , and C . Concretely, the adversary first computes

$$\begin{aligned}
\sigma_1^{h_2} &= A^{h_1 h_2} \cdot B^{m_{11} h_2} \cdot C^{m_{12} h_2}, \\
\sigma_1^{h_3} &= A^{h_1 h_3} \cdot B^{m_{11} h_3} \cdot C^{m_{12} h_3}, \\
\sigma_1^{h_1} &= A^{h_2 h_1} \cdot B^{m_{21} h_1} \cdot C^{m_{22} h_1}, \\
\sigma_1^{h_1} &= A^{h_3 h_1} \cdot B^{m_{31} h_1} \cdot C^{m_{32} h_1},
\end{aligned} \tag{17}$$

and then computes

$$\begin{aligned}
\frac{\sigma_1^{h_2}}{\sigma_2^{h_2}} &= \frac{A^{h_1 h_2} \cdot B^{m_{11} h_2} \cdot C^{m_{12} h_2}}{A^{h_2 h_1} \cdot B^{m_{21} h_1} \cdot C^{m_{22} h_1}} \\
&= \frac{B^{m_{11} h_2} \cdot C^{m_{12} h_2}}{B^{m_{21} h_1} \cdot C^{m_{22} h_1}} \\
&= B^{m_{11} h_2 - m_{21} h_1} \cdot C^{m_{12} h_2 - m_{22} h_1}, \\
\frac{\sigma_1^{h_3}}{\sigma_3^{h_1}} &= \frac{A^{h_1 h_3} \cdot B^{m_{11} h_3} \cdot C^{m_{12} h_3}}{A^{h_3 h_1} \cdot B^{m_{31} h_1} \cdot C^{m_{32} h_1}} \\
&= \frac{B^{m_{11} h_3} \cdot C^{m_{12} h_3}}{B^{m_{31} h_1} \cdot C^{m_{32} h_1}} \\
&= B^{m_{11} h_3 - m_{31} h_1} \cdot C^{m_{12} h_3 - m_{32} h_1}.
\end{aligned} \tag{18}$$

Next, the adversary computes

$$\begin{aligned}
\left(\frac{\sigma_1^{h_2}}{\sigma_2^{h_2}} \right)^{m_{11} h_3 - m_{31} h_1} &= B^{(m_{11} h_2 - m_{21} h_1)(m_{11} h_3 - m_{31} h_1)} \cdot C^{(m_{12} h_2 - m_{22} h_1)(m_{11} h_3 - m_{31} h_1)}, \\
\left(\frac{\sigma_1^{h_3}}{\sigma_3^{h_1}} \right)^{m_{11} h_2 - m_{21} h_1} &= B^{(m_{11} h_2 - m_{21} h_1)(m_{11} h_3 - m_{31} h_1)} \cdot C^{(m_{12} h_3 - m_{32} h_1)(m_{11} h_2 - m_{21} h_1)}, \\
\frac{(\sigma_1^{h_2} / \sigma_2^{h_2})^{m_{11} h_3 - m_{31} h_1}}{(\sigma_1^{h_3} / \sigma_3^{h_1})^{m_{11} h_2 - m_{21} h_1}} &= \frac{B^{(m_{11} h_2 - m_{21} h_1)(m_{11} h_3 - m_{31} h_1)} \cdot C^{(m_{12} h_2 - m_{22} h_1)(m_{11} h_3 - m_{31} h_1)}}{B^{(m_{11} h_2 - m_{21} h_1)(m_{11} h_3 - m_{31} h_1)} \cdot C^{(m_{12} h_3 - m_{32} h_1)(m_{11} h_2 - m_{21} h_1)}} \\
&= C^{(m_{12} h_2 - m_{22} h_1)(m_{11} h_3 - m_{31} h_1) - (m_{12} h_3 - m_{32} h_1)(m_{11} h_2 - m_{21} h_1)}.
\end{aligned} \tag{19}$$

From this equation, we can know

$$C = \left\{ \frac{(\sigma_1^{h_2} / \sigma_2^{h_2})^{m_{11} h_3 - m_{31} h_1}}{(\sigma_1^{h_3} / \sigma_3^{h_1})^{m_{11} h_2 - m_{21} h_1}} \right\}^{(1 / ((m_{12} h_3 - m_{32} h_1)(m_{11} h_2 - m_{21} h_1) - (m_{12} h_2 - m_{22} h_1)(m_{11} h_3 - m_{31} h_1)))}. \tag{20}$$

The value of B can be obtained by substituting the value of C into formula (3):

$$\begin{aligned}
B &= \left\{ \left(\frac{\sigma_1^{h_3}}{\sigma_3^{h_1}} \right)^{m_{11}h_2 - m_{21}h_1} \cdot C^{-(m_{12}h_3 - m_{32}h_1)(m_{11}h_2 - m_{21}h_1)} \right\}^{(1/(m_{11}h_2 - m_{21}h_1)(m_{11}h_3 - m_{31}h_1))} \\
&= \left\{ \left(\frac{\sigma_1^{h_3}}{\sigma_3^{h_1}} \right)^{m_{11}h_2 - m_{21}h_1} \cdot \left[\frac{(\sigma_1^{h_2}/\sigma_1^{h_2})^{m_{11}h_3 - m_{31}h_1}}{(\sigma_1^{h_3}/\sigma_1^{h_3})^{m_{11}h_2 - m_{21}h_1}} \right]^{(1/(m_{12}h_3 - m_{32}h_1)(m_{11}h_2 - m_{21}h_1) - (m_{12}h_2 - m_{22}h_1))} \right\}^{(1/(m_{11}h_2 - m_{21}h_1)(m_{11}h_3 - m_{31}h_1))}
\end{aligned} \tag{21}$$

According to formula (3), the following results can be obtained:

$$A = \left(\frac{\sigma_1}{B^{m_{11}} \cdot C^{m_{12}}} \right)^{(1/h_1)}. \tag{22}$$

The value of A can be obtained by substituting the value of B and C into the above equation.

Through the above process, the adversary can obtain the value of $\lambda^x, g^{\alpha_1 x}, g^{\alpha_2 x}$. Significantly, when $s > 2$, the value of $g^{\alpha_3 x}, g^{\alpha_4 x}, \dots, g^{\alpha_s x}$ can be calculated in the same way. In this way, the adversary can get the key parameters of tags.

(4) Now, the malicious cloud server modifies data blocks

$$m_{11}, m_{12}, \dots, m_{1s}; m_{21}, m_{22}, \dots, m_{2s}, \dots, m_{n1}, m_{n2}, \dots, m_{ns} \tag{23}$$

to be any other data blocks

$$\bar{m}_{11}, \bar{m}_{12}, \dots, \bar{m}_{1s}; \bar{m}_{21}, \bar{m}_{22}, \dots, \bar{m}_{2s}, \dots, \bar{m}_{n1}, \bar{m}_{n2}, \dots, \bar{m}_{ns}. \tag{24}$$

(5) The adversary knows the value of

$$\lambda^x, g^{\alpha_1 x}, g^{\alpha_2 x}, \dots, g^{\alpha_s x}. \tag{25}$$

and thus it can compute the forged authentication label for modified data blocks as follows:

$$\begin{aligned}
\bar{\sigma}_1 &= (\lambda^x)^{h_1} \cdot (g^{\alpha_1 x})^{\bar{m}_{11}} \cdot (g^{\alpha_2 x})^{\bar{m}_{12}} \dots (g^{\alpha_s x})^{\bar{m}_{1s}}, \\
\bar{\sigma}_2 &= (\lambda^x)^{h_2} \cdot (g^{\alpha_1 x})^{\bar{m}_{21}} \cdot (g^{\alpha_2 x})^{\bar{m}_{22}} \dots (g^{\alpha_s x})^{\bar{m}_{2s}}, \\
&\vdots \\
\bar{\sigma}_n &= (\lambda^x)^{h_n} \cdot (g^{\alpha_1 x})^{\bar{m}_{n1}} \cdot (g^{\alpha_2 x})^{\bar{m}_{n2}} \dots (g^{\alpha_s x})^{\bar{m}_{ns}}.
\end{aligned} \tag{26}$$

4.2. Attack II. Our attack II is based on the following observation: even if the CSP does not store any data blocks, the auditor does not need to carry out the audit work and store RBMT T , and the CSP and the auditor can conspire to generate the correct log file, which makes the client believe

that CSP stores data integrally and the auditor performs the audit work honestly. Concretely, the attack is the following:

- (1) Store Protocol: after receiving the client's data M and its corresponding tag collection Σ , the CSP first verifies the correctness of their signatures according to the original scheme. The malicious cloud server can get the value of $\lambda^x, g^{\alpha_1 x}, g^{\alpha_2 x}, \dots, g^{\alpha_s x}$ through doing the same as the attack I. Then, CSP deletes the client's data and its corresponding tags. After receiving the auditing work and T from client, the auditor does the same as original scheme but deletes T .
- (2) Audit Data Protocol: in this step, the malicious cloud server and the auditor complete the interactive process of challenge and response, but the CSP does not have the real data, and the auditor does not need to complete the verification work.
 - (a) The auditor generates a challenge $Q^{(b)} = (b, k_\pi^{(b)}, k_f^{(b)})$ as original scheme and sends it to CSP.
 - (b) After receiving the challenge $Q^{(b)}$, CSP first computes the challenged indices as follows:
$$i_\eta = \pi_{k_\pi^{(b)}}(\eta),$$

$$\alpha_\eta = f_{k_f^{(b)}}(\eta) (1 \leq \eta \leq l). \tag{27}$$
 - (c) To generate the proof, CSP randomly chooses g and $\hat{h}_j \in Z_q (j \in i_\eta)$ and computes a combination of the challenged blocks as $\hat{\mu}_z^{(b)} = \sum_{\eta=1}^l \alpha_\eta m_{i_\eta z} \in Z_q (1 \leq z \leq s)$.
 - (d) For any $\hat{m}_j \in Z_q (j \in i_\eta)$, the malicious cloud server computes the forged tags as $\hat{\sigma}_j = (\lambda^x)^{h_j} (g^{\alpha_1 x})^{m_{j1}} (g^{\alpha_2 x})^{m_{j2}} \dots (g^{\alpha_s x})^{m_{js}}$ and aggregates the tags as $\hat{\sigma}^{(b)} = \prod_{\eta=1}^l \sigma_{i_\eta}^{\alpha_\eta} \in G$.
 - (e) CSP responds the auditor with the proof $\hat{\rho}^{(b)} = \{\hat{\mu}_1^{(b)}, \hat{\mu}_2^{(b)}, \dots, \hat{\mu}_s^{(b)}, \hat{\sigma}^{(b)}\}$ and its signature $\text{Sig}_{sk_{\text{CSP}}}(\hat{\rho}^{(b)})$. Note that CSP also need to send the forged hash value $\hat{h}_j \in Z_q (j \in i_\eta)$ to the auditor.
 - (f) The auditor verifies the validity of $\text{Sig}_{sk_{\text{CSP}}}(\hat{\rho}^{(b)})$, and if it is correct, the auditor does the next

calculation. First, with the value of $\hat{h}_j \in Z_q$ ($j \in i_\eta$) received from CSP, the auditor computes the value as follows:

$$\hat{h}^{(b)} = \lambda^{\sum_{\eta=1}^l a_\eta h_{i_\eta}} \in G. \quad (28)$$

Finally, the auditor does not need expand a lot of computational expanse to verify the proof $\hat{\rho}^{(b)}$, but creates the following log directly:

$$\hat{L}^{(b)} = \{t, Q^{(b)}, \hat{h}^{(b)}, \hat{\rho}^{(b)}, \text{Sig}_{\text{skCSP}}^{(b)}\}. \quad (29)$$

(3) Audit Log Protocol: here we show that the malicious auditor has the ability to generate correct log file, which can convince the client that he has honestly performed the auditing work and that CSP has honestly stored all data.

(a) The auditor finds $Q^{(b)}$, $\hat{h}^{(b)}$, and $\hat{\rho}^{(b)}$ from his log file Λ for each $b \in B$ with the random subset B of indices of Bitcoin blocks and computes

$$\begin{aligned} \hat{h}^{(B)} &= \prod_{b \in B} \hat{h}^{(b)} \in G, \hat{\sigma}^{(B)} = \prod_{b \in B} \hat{\sigma}^{(b)} \in G, \\ \hat{\mu}_z^{(B)} &= \sum_{b \in B} \hat{\mu}_z^{(b)} \in Z_q \quad (1 \leq z \leq s). \end{aligned} \quad (30)$$

$$\begin{aligned} \hat{\sigma}^{(B)} &= \prod_{b \in B} \prod_{\eta=1}^l \hat{\sigma}_{i_\eta}^{a_\eta} = \prod_{b \in B} \prod_{\eta=1}^l \left(\hat{h}_{i_\eta} \cdot g^{\sum_{z=1}^s \alpha_z \hat{m}_{i_\eta}} \right)^{a_\eta x} \\ &= \left(\prod_{b \in B} \left(\hat{h}_{i_\eta} \right)^{\sum_{\eta=1}^l a_\eta} \cdot g^{\sum_{z=1}^s \alpha_z \left(\sum_{\eta=1}^l a_\eta \hat{m}_{i_\eta} \right)} \right)^x = \left(\hat{h}^{(B)} \cdot g^{\sum_{z=1}^s \alpha_z \left(\sum_{b \in B} \hat{\mu}_z^{(b)} \right)} \right)^x = \left(\hat{h}^{(B)} \cdot g^{\sum_{z=1}^s \alpha_z \hat{\mu}_z^{(B)}} \right)^x. \end{aligned} \quad (33)$$

5. Improved Secure Auditing Protocol

In this section, we give an improved secure auditing protocol.

5.1. Setup Protocol. Each participant $P \in \{\text{CSP, client, auditor}\}$ performs KeyGenss to obtain skp and vkp . In addition, the client chooses $s+1$ random elements $\alpha_1, \alpha_2, \dots, \alpha_s, x \in Z_q$ and computes $g_1 = g^{\alpha_1}, g_2 = g^{\alpha_2}, \dots, g_s = g^{\alpha_s}, y = g^x \in G$. The client's secret keys and public keys are denoted as $\text{sk} = (\alpha_1, \alpha_2, \dots, \alpha_s, x)$ and $\text{pk} = (g, g_1, g_2, \dots, g_s, y)$.

5.2. Store Protocol. The data file held by the client is divided into n data blocks as $M = (m_1, m_2, \dots, m_n)$, and each m_i consists of s sectors. More precisely, the data block has the

Then, the auditor generates the proof of appointed logs as original scheme as follows:

$$\hat{\rho}^{(B)} = \left\{ \sqcup_p, \hat{h}^{(B)}, \hat{\mu}_1^{(B)}, \hat{\mu}_2^{(B)}, \dots, \hat{\mu}_s^{(B)}, \hat{\sigma}^{(B)} \right\}, \quad (31)$$

and sends it to the client with $\text{Sig}_{\text{ska}}(\hat{\rho}^{(B)})$.

(b) The client first verifies the correctness of $\text{Sig}_{\text{ska}}(\hat{\rho}^{(B)})$ and \sqcup_p and then verifies the following equations:

$$\begin{aligned} h^{(B)} &\stackrel{?}{=} \lambda^{\sum_{b \in B} \left(l \sum_{\eta=1}^l a_\eta h_{i_\eta} \right)}, \\ \sigma^{(B)} &\stackrel{?}{=} \left(h^{(B)} \cdot g^{\sum_{z=1}^s \alpha_z \mu_z^{(B)}} \right)^x. \end{aligned} \quad (32)$$

Here we can verify that the forged proof $\hat{\sigma}^{(B)}$ is a valid one if the following equation holds:

form $m_i = m_{i1} m_{i2} \dots \| m_{is}$ ($1 \leq i \leq n$) such that each sector $m_{iz} \in Z_q$ ($1 \leq z \leq s$), where $\|$ denotes concatenation.

Constructing RBMT. With all data blocks, the client computes hash values $h_i = H_2(m_i)$ ($1 \leq i \leq n$). Then, it constructs RBMT T on top of the ordered hash values, meaning that each leaf node w_i stores the corresponding hash value h_i .

Computing EHVT. Based on g and sk , the client computes

$$\sigma_i = \left(h_i \cdot g^{\sum_{z=1}^s \alpha_z m_{iz}} \right)^x \in G \quad (1 \leq i \leq n). \quad (34)$$

Then, the client generates the processed data $\mathcal{M} = \{M, \Sigma\}$, where $\Sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$. $\mathcal{M} = \{M, \Sigma\}$.

5.2.1. *Outsourcing Data.* The client sends \mathcal{M} and $\text{Sig}_{skc}(\mathcal{M})$ to CSP. CSP verifies $\text{Sig}_{skc}(\mathcal{M})$; if $\text{Sig}_{skc}(\mathcal{M})$ passes, CSP accepts \mathcal{M} .

5.2.2. *Outsourcing Auditing Work.* The client outsources auditing work to the auditor by sending T with $\text{Sig}_{skc}(T)$. If $\text{Sig}_{skc}(T)$ is passed, the auditor accepts T .

5.2.3. *Agreeing Parameters.* A public parameter $\mathbb{P} = \{n, h_{\text{root}}\}$ needs to be agreed on by three participants, where n is the total number of data blocks and h_{root} is the Merkle root of T . In addition, the client and auditor need to further agree on a contract $\mathbb{C} = \{\text{BI}, F, l\}$ that specifies the checking policy for the auditor. BI denotes a Bitcoin block index from which the auditing work starts, F denotes the auditing frequency, and l denotes the number of challenged data blocks for each auditing. Now the client deletes \mathcal{M} and T from the local storage.

5.3. *AuditData Protocol.* The scheme leverages the Bitcoin blockchain as a time-dependent pseudo-random source to generate periodic challenges. The auditor first inputs the time $t \in \tau$ to obtain a hash value $\text{hash}^{(b)} \in \{0, 1\}^{l_{\text{hash}}}$ of the latest block that has appeared since time t in Bitcoin blockchain. Then, PRBG is invoked on the $\text{hash}^{(b)}$ to obtain long enough pseudo-random bits, which will be sequentially used by the auditor to select a pair of keys $k_{\pi}^{(b)}, k_f^{(b)}$. At last, the auditor generates a challenge $Q^{(b)} = \{b, k_{\pi}^{(b)}, k_f^{(b)}\}$ and sends it to the CSP, where the block b corresponds to the time t .

Upon receiving the challenge $Q^{(b)}$, the CSP first computes the challenged indices and coefficients as follows:

$$\begin{aligned} i_{\eta} &= \pi_{k_{\pi}^{(b)}}(\eta), \\ a_{\eta} &= f_{k_f^{(b)}}(\eta) \quad (1 \leq \eta \leq l). \end{aligned} \quad (35)$$

Then, the CSP computes the proof of data possession to verify the integrity as follows:

$$\begin{aligned} \mu_z^{(b)} &= \sum_{\eta=1}^l a_{\eta} m_{i_{\eta} z} \in \mathbb{Z}_q \quad (1 \leq z \leq s), \\ \sigma^{(b)} &= \prod_{\eta=1}^l \sigma_{i_{\eta}}^{a_{\eta}} \in G. \end{aligned} \quad (36)$$

Finally, the CSP responses the auditor with the proof $\rho^{(b)} = \{\mu_1^{(b)}, \mu_2^{(b)}, \dots, \mu_s^{(b)}, \sigma^{(b)}\}$, and then the auditor verifies the correctness of $\rho^{(b)}$. First, the auditor computes the challenged indices and coefficients. Second, with the corresponding hash values stored in his local T , the auditor computes the value

$$h^{(b)} = \prod_{\eta=1}^l (h_{i_{\eta}})^{a_{\eta}} \in G. \quad (37)$$

Third, the auditor checks the following equation to verify the proof:

$$e(\sigma^{(b)}, g) \stackrel{?}{=} e\left(h^{(b)} \cdot \prod_{z=1}^s g_z^{\mu_z^{(b)}}, y\right). \quad (38)$$

If the equation holds, it means that the challenged data blocks are intact. Lastly, the auditor creates the following log entry that records his auditing work:

$$L^{(b)} = \{t, Q^{(b)}, h^{(b)}, \rho^{(b)}, \text{Sig}_{skc_{\text{CSP}}}(\rho^{(b)})\}, \quad (39)$$

and saves it in his local log file Λ . The correctness of the equality can be elaborated as follows:

$$\begin{aligned} e(\sigma^{(b)}, g) &= e\left(\prod_{\eta=1}^l (h_{i_{\eta}} \cdot g_{\sum_{z=1}^s \alpha_z m_{i_{\eta} z}})^{a_{\eta}}, g^x\right) \\ &= e\left(\prod_{\eta=1}^l (h_{i_{\eta}})^{a_{\eta}} \cdot \prod_{\eta=1}^l \left(\prod_{z=1}^s g_z^{a_{\eta} m_{i_{\eta} z}}\right), y\right) \\ &= e\left(h^{(b)} \cdot \prod_{z=1}^s g_z^{\sum_{\eta=1}^l a_{\eta} m_{i_{\eta} z}}, y\right) \\ &= e\left(h^{(b)} \cdot \prod_{z=1}^s g_z^{\mu_z^{(b)}}, y\right). \end{aligned} \quad (40)$$

5.4. *AuditLog Protocol.* The client chooses a random subset B of indices of Bitcoin blocks and sends it to the auditor. Once receiving B , the auditor finds $Q^{(b)}, h^{(b)}$, and $\rho^{(b)}$ from his log file Λ for each $b \in B$ and computes

$$\begin{aligned} h^{(B)} &= \prod_{b \in B} h^{(b)} \in G, \\ \sigma^{(B)} &= \prod_{b \in B} \sigma^{(b)} \in G, \\ \mu_z^{(B)} &= \sum_{b \in B} \mu_z^{(b)} \in \mathbb{Z}_q \quad (1 \leq z \leq s). \end{aligned} \quad (41)$$

In addition, for each $b \in B$, the auditor reads $k_{\pi}^{(b)}$ from $Q^{(b)}$ and computes the challenged indices i_{η} ($1 \leq \eta \leq l$) by invoking $\pi_{k_{\pi}^{(b)}}(\eta)$. After eliminating the repetitive indices, the last ordered challenge index vector is denoted by $C = (i_1, i_2, \dots, i_c)$. Then, the auditor runs Gen Multi Proof (T, C) to obtain the corresponding multi-proof \sqcup_p . At last, the auditor generates the proof of appointed logs as below:

$$\rho^{(B)} = \{\sqcup_p, h^{(B)}, \mu_1^{(B)}, \mu_2^{(B)}, \dots, \mu_s^{(B)}, \sigma^{(B)}\}, \quad (42)$$

and sends it to the client with $\text{Sig}_{sk_a}(\rho^{(B)})$.

After verifying $\text{Sig}_{sk_a}(\rho^{(B)})$, for each $b \in B$, the client first invokes PRBG($\text{hash}^{(b)}$) to get $Q^{(b)}$ and reconstructs the challenged indices and coefficients i_{η}, a_{η} ($1 \leq \eta \leq l$). Then, the client verifies the correctness of \sqcup_p by calling VerMulti Proof ($\sqcup_p, n, h_{\text{root}}, C$), where C can be obtained by utilizing her own constructed indices for all $b \in B$. If the verification is passed, it means that all the challenged leaf nodes w_{i_j} ($1 \leq j \leq c$) in \sqcup_p are authenticated. Finally, with all

authenticated h_{i_η} , the client verifies $h(B)$ by checking the following equation :

$$h^{(B)} \stackrel{?}{=} \prod_{b \in B} \prod_{\eta=1}^l \left(h_{i_\eta} \right)^{a_\eta}. \quad (43)$$

If this verification passes, the client checks the last equation by using sk and $h^{(B)}$:

$$\sigma^{(B)} \stackrel{?}{=} \left(h^{(B)} \cdot g^{\sum_{z=1}^s \alpha_z \mu_z^{(B)}} \right)^x. \quad (44)$$

If the above equation holds, it means that the auditor audited the past challenged data blocks appointed by B honestly. The correctness of the equation can be elaborated as below:

$$\begin{aligned} \zeta^{(B)} &= \prod_{b \in B} \prod_{\eta=1}^l \sigma_{i_\eta}^{a_\eta}, \\ &= \prod_{b \in B} \prod_{\eta=1}^l \left(h_{i_\eta} \cdot g^{\sum_{z=1}^s \alpha_z m_{i_\eta z}} \right)^{a_\eta x} \\ &= \left(\prod_{b \in B} \left(h_{i_\eta} \right)^{\sum_{\eta=1}^l a_\eta} \cdot g^{\sum_{z=1}^s \alpha_z \left(\sum_{\eta=1}^l a_\eta m_{i_\eta z} \right)} \right)^x \\ &= \left(h^{(B)} \cdot g^{\sum_{z=1}^s \alpha_z \left(\sum_{b \in B} \mu_z^{(b)} \right)} \right)^x \\ &= \left(h^{(B)} \cdot g^{\sum_{z=1}^s \alpha_z \mu_z^{(B)}} \right)^x. \end{aligned} \quad (45)$$

In our improved protocol, the construction of data tags is

$$\sigma_i = \left(h_i \cdot g^{\sum_{z=1}^s \alpha_z m_{iz}} \right)^x \in G \quad (1 \leq i \leq n), \quad (46)$$

instead of

$$\sigma_i = \left(\lambda^{h_i} \cdot g^{\sum_{z=1}^s \alpha_z m_{iz}} \right)^x \in G \quad (1 \leq i \leq n). \quad (47)$$

Therefore, the adversary cannot forge the authentication tags as attack I. Furthermore, malicious CSP and auditor cannot conspire to deceive the client through attack II.

6. Conclusion

In this paper, we point out that Guo et al.'s outsourced dynamic provable data possession scheme with batch update for secure cloud storage is not secure. The authentication tags can be easily forged, and thus the cloud server can modify or delete the data arbitrarily, and the auditor cannot carry out auditing work. In all these attacks, the cloud server can still give correct data possession proofs, and the auditor can still give correct auditing log files. Finally, an improved secure cloud storage auditing protocol is given. We remark that Guo et al.'s outsourced dynamic provable data

possession scheme with batch update for secure cloud storage is very novel but has some design flaw, and we hope similar shortcoming can be avoided in future scheme designs to improve the security of public auditing protocols.

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 study was supported by the National Key Research and Development Program of China (grant no. 2017YFB0802000), National Natural Science Foundation of China (grant nos. U1636114 and 61572521), Foundation of Guizhou Provincial Key Laboratory of Public Big Data (no. 2019BDKFJJ008), Engineering University of PAP's Funding for Scientific Research Innovation Team (grant no. KYTD201805), and Engineering University of PAP's Funding for Key Researcher (no. KYGG202011).

References

- [1] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
- [2] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNS," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
- [3] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, "A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2347–2360, 2020.
- [4] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, "A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2347–2360, 2019.
- [5] J. Xiong, M. Zhao, M. Bhuiyan, L. Chen, and Y. Tian, "An ai-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of iot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–923, 2019.
- [6] Y. Deswarte, J. J. Quisquater, and A. Saïdane, *Remote Integrity Checking*, Springer, New York, NY, USA, 2004.
- [7] D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfer," IACR, Lyon, France, Report 2006/150, 2006.
- [8] F. Seb e, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1034–1038, 2008.
- [9] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in *Proceedings of the Hotos07: Workshop on Hot Topics in Operating Systems*, DBLP, San Diego, CA, USA, May 2007.

- [10] M. Naor and G. N. Rothblum, "The complexity of online memory checking," *Journal of the ACM*, vol. 56, no. 1, pp. 1–46, 2007.
- [11] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the Acm Conference on Computer & Communications Security*, October 2007.
- [12] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.
- [13] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1717–1726, 2012.
- [14] T. Hui, Y. Chen, C. C. Chang et al., "Dynamic-hash-table based public auditing for secure cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 710–714, 2017.
- [15] A. Juels and B. S. K. Jr, "Proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 584–597, ACM, Alexandria, VA, USA, October 2007.
- [16] H. Shacham and B. Waters, *Compact Proofs of Retrievability*, Springer, Berlin, Germany, 2008.
- [17] J. Zhao, C. Xu, F. Li, and W. Zhang, "Identity-based public verification with privacy-preserving for data storage security in cloud computing," *IEICE - Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E96.A, no. 12, pp. 2709–2716, 2013.
- [18] J. Zhang and Q. Dong, "Efficient id-based public auditing for the outsourced data in cloud storage," *Information Sciences*, vol. 344, pp. 1–14, 2016.
- [19] T. Xin, Y. Qi, and Y. Huang, *Fragile Watermarking Based Proofs of Retrievability for Archival Cloud Data*, Springer, New York, NY, USA, 2016.
- [20] G. Wei, Z. Hua, S. Qin et al., "Outsourced dynamic provable data possession with batch update for secure cloud storage," *Future Generation Computer Systems*, vol. 95, pp. 309–322, 2019.

Research Article

Anticollusion Attack Strategy Combining Trust Metrics and Secret Sharing for Friendships Protection

Junfeng Tian  and Yue Li 

School of Cyberspace Security and Computer Institute, Hebei University, Baoding 071000, China

Correspondence should be addressed to Yue Li; 670186807@qq.com

Received 2 May 2021; Accepted 26 June 2021; Published 5 July 2021

Academic Editor: James Ying

Copyright © 2021 Junfeng Tian and Yue Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Online social networks provide users with services such as online interaction, instant messaging, and information sharing. The friend search engine, a new type of social application, provides users with the service for querying the list of other individuals' friends. Currently, the existing research focuses on independent attacks for friend search engines while ignoring the more complicated collusion attacks, which can expose more friendships that users are not willing to share. Compared with independent attackers, collusion attackers share query results by cooperating with each other. In this article, we propose a resistance strategy against collusion attacks to protect the friendship privacy. The proposed trust metric is based on users' behaviors and is combined with Shamir's secret sharing system, which can transform friendships into secrets. Through secret distribution and reconfiguration, only the participants who meet the query requirements can successfully reconstruct the secret, while the participants who do not meet the query conditions cannot successfully obtain the secret fragments even if they obtain the secret fragments. Experiments are conducted to verify the effectiveness of the proposed strategy and proved that this strategy can greatly limit the number of malicious attackers, greatly reduce the probability of successful collusion attacks, and reduce the number of victims.

1. Introduction

Friendship, as the beginning of social networks, is one of the most important factors in the development of online social networks (OSNs). Friendship is also the basis of social relationships. The friend search engine was born with the development of social networks. It provides a service for users in social networks to browse other users' friends list. According to "finder mind," the top 25 friend search engines help users find anyone for free and with high quality [1]. The powerful function of searching friends with friend search engines provides great convenience for users to search for familiar or interested friends and potentially attracts more people to join social networks.

1.1. Problem Identification. Friend search engines may reveal more friendships than users are willing to share, which is considered a privacy violation. Without a proper

protection strategy to address such privacy leakage, friendships that users do not want to display are always revealed, which will lead to users no longer using OSNs. Currently, available protection schemes [2] have been shown to resist malicious queries of friendships by independent attackers. The social network can record the query history of each individual requester, and when a query is made to the same user, the attacker always obtains the same friends list as a result of the query. With a defence strategy, an independent attacker cannot query for friendships outside the user's privacy settings.

A complicity attack by multiple queries has emerged [3]. It is accomplished by multiple malicious attackers who share query results by coordinating the query targets and sequences to make users reveal their friendships outside the privacy settings. Collusion attackers can gain access to the users' friendships that cannot be queried by independent attackers. Since existing defence strategies can only protect friendship privacy from independent attacks, they cannot

effectively resist conspiracy attacks. However, the Friend-Guard supports only two kinds of friend searches, including unweighted and popularity-based friend search [4]. The methods and characteristics of the conspiracy attacks need to be analyzed, and the query strategy needs to be studied and improved in order to protect the friendship privacy. The data sharing framework can resolve potential data leakage [5]. We focus on the design of an anticollusion attack strategy that is aimed at the privacy of users' friendships in OSNs.

1.2. Methods and Contributions. Friendships serve as the basis for interaction between users in social networks and as an extension of interpersonal relationships in the real world. Collusion attacks on friendships can lead to the leakage of interpersonal relationships outside the user's settings, which will cause a more serious impact on the stability of OSNs. To address such complicity attacks in social networks, we design a privacy protection strategy for friendships that can resist complicity attacks by combining the trust metric [6] with a (t, n) threshold function [7], drawing on the idea of secret sharing. The main contributions can be divided into the following three aspects:

A method to measure trust based on users' interaction behaviors is proposed. Combining the important features of users' interactions, the attributes that affect the trust metric are identified. Direct trust, recommendation trust, and comprehensive trust are calculated, and the friend queriers are classified according to the trust metric to control them when they query friendships.

A privacy-preserving strategy for friendships that can resist conspiracy attacks is proposed. The trust metric is combined with the Shamir Secret Sharing (SSS) system to transform friendships into secrets. The (t, n) threshold function is specially applied so that after sharing the secret, only a subset of the target user's friendships can be successfully queried by satisfying a specific condition, thus protecting the privacy of friendships.

The experimental design and implementation are described. First, the rationality of the trust metric is verified by a probabilistic random function. Then, the security is verified by experimenting against the collusion attack scheme. The feasibility and security are illustrated in terms of the limit rate, the number of victims, the number of attackers, and the probability of successful attacks.

2. Related Works

2.1. Attacks on the Privacy of Friendships. The number of users in OSNs continues to grow. Tens of thousands of users search for new friends and establish new contacts every day. Therefore, the privacy problem in friend search engines has attracted the attention of many researchers. Attacks against the privacy of friendships in OSNs can be divided into two

categories: attacks initiated by independent attackers and by colluding attackers.

Regarding independent attacks, research on modeling malicious attacks in OSNs showed that malicious individuals use the actual trust relationship between users and their family and friends to spread malware via OSNs [8]. By changing the display of malicious posts and personal information and hiding him/herself to avoid detection, an attacker in a chameleon attack, which is a new type of deception based on OSNs, is able to destroy users' privacy [9]. Studies have also shown that when the topology of OSNs does not contain cycles, malicious entities will violate users' privacy via active attacks if the network structure is not carefully designed [10]. Due to the rapid development of convolutional neural networks in recent years, applying them to social networks can result in very effective reasoning attacks and make high-precision predictions about private data [11]. In the heuristic attack model based on the Dopv attack [12], the attacker obtains the number of friends of the victim from two published social network graphs by spoofing the trust or browsing the homepage. The tag symmetry attack identifies a pair of friends by marking two fixed-point tags that connect the same edge [13]. An attacker can also identify the friendships of a pair of users by the number of their mutual friends [14]. Although OSN can hide the identity of the user by removing the user's identifier, an attacker can use other contextual information about the OSN to infer the identity of the target user [15].

Collusion attacks involve multiple malicious entities with the aim of launching a malicious attack through the coordination of multiple malicious entities to obtain more private information than is obtained in independent attacks. Multiple malicious entities can be fake accounts that are created by a single attacker or different real attackers [16, 17].

The router and users can maliciously collude to perform a collusion name guessing attack to compromise people's privacy [18]. Compared with independent attacks, collusion attacks are more complex and often exploit system vulnerabilities that independent attacks cannot detect. There is a complex collusion attack strategy in which multiple malicious users coordinate their queries, share the query results, and dynamically adjust their query based on the system's feedback to other malicious requestors [3].

2.2. Protection on the Privacy of Friendships. The actual parameter settings of social network providers have an impact on the display of users' personal information [19]. The personalized privacy measurement algorithm can calculate the user's privacy level, thereby protecting privacy data [20]. Moreover, the classification-anonymity model effectively guarantees the privacy of sensitive data [21]. Users' privacy is secured by encrypting data, and only authorized parties who have obtained the key can decrypt the encrypted content [22]. The blockchain-based secure key management scheme can improve trustworthiness more effectively and efficiently [23].

The additive secret sharing technique can encrypt raw data into two ciphertexts and construct two classes of secure

functions, which can be used to implement a privacy-preserving convolutional neural network [24]. Trust and identity are fundamental issues in social and online environments, and trust management can help users build trust and establish relationships with other users [25]. Existing relationships of users in social networks can be described as one-hop trust relationships, and further multihop trust relationships are built during the recommendation process [26]. When a user involves data items from multiple users, the trust value among users can be used to weigh the weight of user opinions to determine whether the data items are released or not, thus enabling collaborative privacy management [27]. In addition, a series of studies have proposed an unsupervised trust inference algorithm that is based on collaborative filtering in weighted social networks and a fast and robust trust inference algorithm [28, 29] to strengthen the security of social networks via trust inference and to satisfy the goal of differential privacy, a privacy and availability data clustering (PADC) scheme based on k -means algorithm and differential privacy is proposed, which can enhance the selection of the initial center points and the distance calculation method from other points to the center point [30].

However, researchers rarely consider privacy leakage problems caused by the friend search service provided by OSNs. Research on these problems can address the privacy needs of users' friends while ensuring the sociality of OSNs. The solution adopted by most OSNs is to allow each individual user to choose to completely display or completely hide their entire friend list. Moreover, OSNs often default their users to expose the entire friend list, of which most users are unaware [31]. It is conceivable that this setting aims to increase the sociality of the OSN. If users set their friend list to be completely hidden to protect the privacy of their friendships, this setting will substantially affect the sociality of OSNs. There are also some OSNs that set the users' friend list display to "show only a fixed number." For example, on Facebook, the number of friends displayed is set to 8, which limits the flexibility of users in changing their personal settings. However, some researchers have discovered that randomly displaying eight friends is sufficient for third parties to obtain data to estimate friend lists [32]. Moreover, regarding the different privacy settings of users, consider the following example: if A and B are friends, even if user A hides his or her friend list and the requestor cannot query the friend list of A , if user B is set to display his or her friend list, when the requestor queries the friend list of B , the friendships of B and A will be displayed and destroy A 's privacy. This problem is referred to as the "mutual effect" [2].

To better protect the privacy of users' friendships in OSNs, a privacy protection strategy in the friend search engine [2] was shown to successfully resist attacks initiated by independent attackers. However, the strategy was unable to defend against collusion attacks initiated by multiple malicious attackers. Subsequently, an advanced collusion attack strategy coordinated by multiple malicious requestors [3] showed that multiple malicious requestors with limited knowledge of OSNs can successfully destroy users' privacy settings in the friend search engine. Another study [33]

implemented web applications to detect malicious behavior such as collusion attacks in the friend search engine. However, few researchers have investigated how to resist collusion attacks initiated by malicious attackers in friend search engines.

In this article, we propose an anticollusion attack strategy to fill these research gaps. This strategy distinguishes trusted users from untrusted users based on the credibility among users in OSNs and uses the (t, n) threshold function to limit the querying of requestors in the friend search engine to resist malicious attacks initiated by colluding attackers in OSNs.

3. Collusion Attack Strategy

3.1. Related Definitions. In friend search engines, to strengthen the protection of the user's friendships, a certain number of friendships, such as k , will be displayed when responding to a query request. These k friends are defined as the most influential friends of the users in the OSN. Assume that node N_a exists in the OSN with direct friends $N_{a,i}$ and that set is $F_a^k (i < k)$. Requestor Q_1 wants to query N_a 's friendships; two nodes, N_1 and N_2 , exist, and $k = 1$. N_1 and N_2 are each user's most important friends.

3.1.1. Unpopular Node. N_a is an unpopular node if nodes $N_{a,i} \in F_a^k$ and $N_a \notin F_k^i$. As Figure 1 shows, N_0 is an unpopular node.

3.1.2. Popular Node. N_a is a popular node if $N_a \in F_a^k$ and $N_{a,i} \in F_k^i$. As Figure 2 shows, N_0 is a popular node.

3.1.3. Occupation. If requestor Q_1 queries node N_1 , based on the friend search engine display strategy, the query result is $E(N_1, N_2)$. At this time, N_1 has shown his or her most important friend N_2 , and N_1 is occupied.

3.1.4. Passive Display. Requestor Q_1 queries the important friend list of N_1 . Based on the friend search engine strategy, the query result is $E(N_1, N_2)$. The most important friend who exposes N_2 is N_1 , and N_2 is referred to as a passive display.

3.2. Attack Model

3.2.1. Maximum Number of Friends Displayed. Due to the different personal preferences of users in OSNs, their privacy settings will also be different. The maximum number of friends displayed, k , may also be different. This strategy assumes that all nodes have the same k value.

3.2.2. Attackers' Prior Knowledge. Typically, the success of a malicious requestor's attack is closely related to his or her knowledge of OSNs. The attack success rate of malicious requestors who know more about OSNs is expected to be higher. This article assumes that malicious requestors have

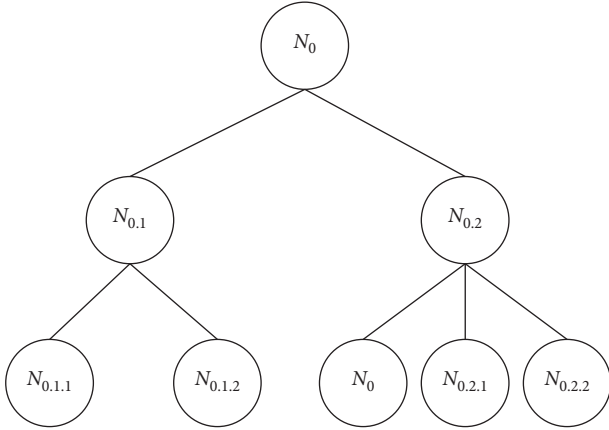


FIGURE 1: Unpopular node.

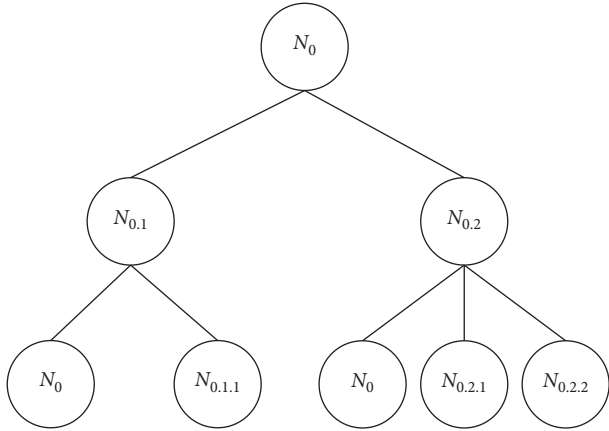


FIGURE 2: Popular node.

limited knowledge of OSNs and are limited only to target nodes.

3.2.3. Attack Target. The goal of the malicious query is to violate the privacy of the target user in the OSN (i.e., to query the $k + 1$ th friend of the target node). When the privacy of the target user is set to show a number of friends less than k , the privacy of the target user cannot be violated. Each malicious requestor's attack target is unique, and each collusion attack has only one victim node. Although malicious requestors may infringe the privacy of other users during the query process, only when the privacy of the target node is destroyed is the collusion attack considered successful.

3.2.4. Attack Strategy. Collusion attackers in OSNs can query users' friendships via the friend search engine and query the relationship between users and friends by coordinating the query sequence and query targets.

(1) Attacks on Unpopular Nodes. When the target user is an unpopular node, since there exists at least one node $N_{a,i} \in F_a^k$, and $N_a \notin F_{a,i}^k$. Therefore, the first malicious attacker obtains the set of its friends F_a^k by querying N_a 's

friends and shares the query result with the new attacker. The new malicious attacker can query N_a 's friends $N_{a,i}$ separately by the query result shared by the first malicious attacker and always find the node in $F_{a,i}^k$ where N_a does not exist. Suppose the node is $N_{a,x}$, and a query on $N_{a,x}$ can show its k friends so that it is occupied. At this point, if a query is performed again on the target node N_a , N_a will display its $k + 1$ th friend $N_{a,(k+1)}$ since $N_{a,x}$ is already occupied.

Suppose there exists a malicious attacker MR_i ($i = 1, 2, \dots$), $k = 1$. Taking the unpopular node in Figure 1 as an example to illustrate the attack process on the non-popular node. The results are shown in Table 1.

(2) Attacks on Popular Nodes. Analogous to the attack on unpopular nodes, the basic idea of the attack on popular nodes is also to expose the $k + 1$ th friend of the target node by occupying one of its top k friends. However, since both popular nodes and their friends are each other's first k friends, directly querying the friends of the target popular node cannot destroy its friendship privacy by appropriation. Therefore, $N_{a,i}$ is occupied k times by passively displaying $N_{a,i}$. However, when the target node and its friend $N_{a,1}$ are each other's first important friend, they cannot be passively displayed.

Taking N_0 as the target node in Figure 3 as an example, suppose there exist malicious attackers MR_i ($i = 1, 2, \dots$) with $k = 3$, The attack process of an attack on popular node N_0 is shown in Table 2.

4. Anticollusion Attack Strategy of Friendships Protection

The collusion attack compromises users' friendship privacy by coordinating the query order through multiple malicious requestors and dynamically adjusting the query target through the query results of others. To solve the problem, we investigate the strategy to resist collusion attacks. In this work, the access control of requestors in the friend search engine is considered, credibility is employed as the restriction condition for requestor queries, and the Shamir Secret Sharing (SSS) system is utilized to control queries.

4.1. Credibility Calculations. In OSNs, the interaction behaviors between users are an important factor that affects the trust metric between users. According to the relationship between users, the trust relationship between two users, i.e., the trust subject and the trust object, can be divided into three types, that are direct trust, recommendation trust, and comprehensive trust. There are four main attributes that are important for credibility calculations.

4.1.1. Number of Interactions. The greater the number of interactions between two users is, the higher the trust between the users is.

4.1.2. Interaction Evaluation. After each interaction, the user gives a corresponding evaluation based on the process, the results, and the importance of the interaction event. The

TABLE 1: Attack process on unpopular node N_0 .

Step	Requestor	Target	Result
1	MR ₁	N_0	$E(N_0, N_{0,1})$
2	MR ₂	$N_{0,1}$ N_0	$E(N_{0,1}, N_{0,1,1})$ $E(N_0, N_{0,2})$

evaluation value of the l th interaction is recorded as $C_l \in [0, 1]$.

4.1.3. Interaction Time. Interaction evaluations that are similar to the current time better reflect the user's recent behavior. The closer the evaluation is to the current time, the greater the impact is on direct credibility.

4.1.4. Interaction Events. The weight of the event of the l th interaction between two users is denoted as W_l .

4.1.5. Direct Trust (DT_{ij}). For two user nodes that have historical interactions in the OSN, the credibility of one user to another is referred to as direct trust. A user obtains the credibility evaluation of another based on the historical performance of the user who has interacted with him or her.

If node i and node j have interacted n times in the OSN, after the l th interaction is completed, node i evaluates node j to obtain evaluation value C_l and interaction event weight W_l . Subsequently, the l th interaction time t_l , importance of the l th interaction event W_l , evaluation value C_l of the interaction event of node i with node j , and the influence of the number n of interactions between node i and node j on the evaluation value are considered. The calculation formula of direct trust is expressed as follows:

$$DT_{ij} = \alpha \cdot \frac{\sum_{l=1}^n \Phi(t_l) \cdot C_l \cdot W_l}{n}, \quad (1)$$

where $\alpha = \sqrt{n/(n+1)}$ is a function of the number of interactions used to adjust the influence of the number of interactions on credibility. The user obtains a high degree of trust only when he or she obtains multiple satisfactory evaluation values. $\varphi(t_l) = \exp(-[(t_n - t_l)/T])$ is the time decay coefficient, where t_n is the n th interaction time (i.e., current interaction time), t_l is the l th interaction time, and T is the time period. The evaluation of an interaction event that is more similar to the current interaction time has a greater impact on credibility. W_l and C_l are the weight of the interaction event between node i and node j and the evaluation value of node i for the event, respectively. This approach can prevent malicious requestors from interacting with the target user by using events with a low weight to gain the trust of the target user while deceiving the user during interaction events with high weights.

4.1.6. Recommended Trust (RT_{ij}). If node i wants to gain a comprehensive understanding of node j , node i needs to obtain the recommended trust for node j via intermediate

node c , where node $c = \{c_1, c_2, c_3, \dots, c_n\}$. The calculation of recommended trust is expressed as follows:

$$RT_{ij} = \sum_{c=1}^n (DT_{ic} \cdot DT_{cj}), \quad (2)$$

where DT_{ic} is the direct trust of user i in user c , DT_{cj} is the direct trust of user c in user j , and the direct trust of user i in user c can be regarded as a recommendation for calculating the recommended trust weights.

4.1.7. Comprehensive Trust (OT_{ij}). The credibility of a user in the OSN must be integrated with his or her direct trust and the recommended trust of other users, which is referred to as comprehensive trust. The weights of direct trust and recommended trust are determined by experimental calculations. In real life, people are generally more inclined to believe their judgments, and the recommendations of others serve only as a reference. Thus, the calculation of comprehensive trust is expressed as follows:

$$OT_{ij} = u \cdot DT_{ij} + v \cdot RT_{ij} \quad (u + v = 1, u > v), \quad (3)$$

where OT_{ij} is the direct trust of node i in node j , RT_{ij} is the recommended trust of node i in node j , and u and v are the weight coefficients of direct trust and recommended trust, respectively.

4.2. Shamir Secret Sharing System. The SSS system is a specific secret sharing scheme designed by Shamir based on language interpolation polynomial theory [34, 35]. This scheme clearly illustrates how to divide data D into n segments so that D can be easily reconstructed from t segments and so that even if all $t - 1$ segments are mastered, D cannot be reconstructed.

In response to collusion attacks in OSNs, this article uses the SSS (t, n) threshold function to control the querying of users' friendships. The (t, n) threshold SSS consists of the following three stages.

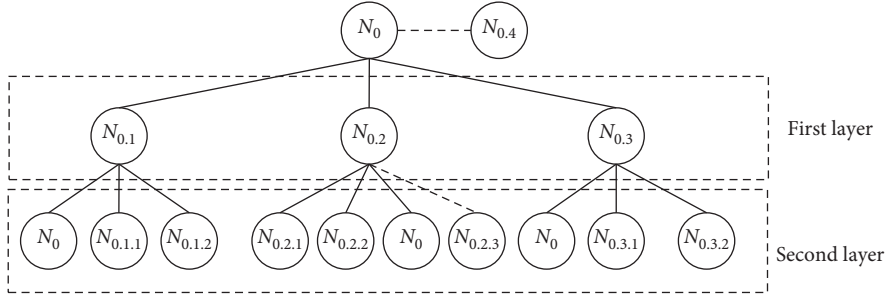
4.2.1. System Parameter Setting. n is the number of all participants, t is the threshold, p is a large prime number, and $s \in Z_p$ is the secret to be shared.

4.2.2. Secret Distribution. The secret distributor D chooses a random t degree polynomial.

$$a(x) = s + a_1x^1 + a_2x^2 + a_3x^3 + \dots + a_{t-1}x^{t-1} \pmod p, \alpha_j \in_R Z_p. \quad (4)$$

The condition $a(0) = s$ is satisfied. D sends $s_i = a(i)$ to participants $P_i, i = 1, 2, \dots, n$.

4.2.3. Secret Reconstruction. Any number of participants can reconstruct the secret using their secret fragments. Let t participants who want to reconstruct the secret be $P_i, i = 1, 2, \dots, t$, and let $A = \{1, 2, \dots, t\}$.

FIGURE 3: Friendship of popular node N_0 .TABLE 2: Attack process on popular node N_0 .

Step	Requestor	Target	Result
1	MR_1	N_0	$E(N_0, N_{0.1}), E(N_0, N_{0.2}), E(N_0, N_{0.3})$
2	MR_2	$N_{0.2}$	$E(N_{0.2}, N_{0.2.1}), E(N_{0.2}, N_{0.2.2}), E(N_{0.2}, N_0)$
3	MR_3	$N_{0.2.1}$ $N_{0.2.2}$ $N_{0.2}$	$E(N_{0.2.1}, N_{0.2.1.1}), E(N_{0.2.1}, N_{0.2}), E(N_{0.2.1}, N_{0.2.1.2})$ $E(N_{0.2.2}, N_{0.2.2.1}), E(N_{0.2.2}, N_{0.2.2.2}), E(N_{0.2.2}, N_{0.2.2.3})$ $E(N_{0.2}, N_{0.2.1}), E(N_{0.2}, N_0), E(N_{0.2}, N_{0.2.3})$
4	MR_4	$N_{0.2.3}$ $N_{0.2}$ N_0	$E(N_{0.2.3}, N_{0.2.3.1}), E(N_{0.2.3}, N_{0.2}), E(N_{0.2.3}, N_{0.2.3.2})$ $E(N_{0.2}, N_{0.2.1}), E(N_{0.2}, N_{0.2.2}), E(N_{0.2}, N_{0.2.3})$ $E(N_0, N_{0.1}), E(N_0, N_{0.3}), E(N_0, N_{0.4})$

λ_i is calculated based on the following formula:

$$\lambda_i = \prod_{j \in A(i)} \frac{j}{j-i}. \quad (5)$$

The original secret is restored based on the following formula:

$$s = \sum_{i \in A} s_i \lambda_i. \quad (6)$$

The security of the SSS depends on the assumption that the parties honestly perform the operations predetermined by the agreement. We consider reliable secret distributors and believe that the administrators of OSNs are honest in the strategy.

4.3. Friend Search Engine with the SSS System

4.3.1. Friendships Transform. When a querier queries the friends of a target user, the friend search engine will return the relationship of edges between nodes among users according to the display strategy. However, according to the SSS system and the requirement of the (t, n) threshold function, the shared secret is $s \in Z_p$ with p being a large prime number. The secret s to be shared in this strategy is the friendship of the target. Therefore, it is necessary to process the representation of an important user's friendship and transform it to the range of Z_p and then share it by the threshold function.

In order to transform the friendships into shareable secrets, we propose a friendship transform algorithm to convert the friendships to satisfy the secret sharing condition. According to the query goal of the querier, the IDs of

the first k friends of the target node are first obtained. The friendships transform algorithm is shown as Algorithm 1.

4.3.2. Friendships Protection. In OSNs, users can access the friendships of other users by friend search engines. Multiple malicious requestors can share their query results with each other by coordinating the query target and query sequence, which causes the target user to expose more friends than the user is willing to display. A friend search engine that has introduced the trust metric and SSS can control the queries of users. This control can guarantee that only users whose comprehensive trust reaches the trust threshold can successfully query the friendships of the target user.

Assume that secret distributor D is honest and that each anonymous requestor $P_i, i = 1, 2, \dots, n$ can obtain a correct secret fragment from D . The number of requestors is higher than the trust threshold for querying the friendships of the target user each time $n_A \geq 2$. The access control process of this solution is described as follows.

Obtain Comprehensive Trust. Requestors $P_i, i = 1, 2, \dots, n$ request querying the friendships of target user n_a , obtaining comprehensive trust T_{ai} of P_i , and sorting the results in descending order by value based on the interaction between target user n_a and requestor P_i in the OSN.

Classify the Query. Based on trust threshold TR , the requestors are divided into categories A and B . Category $A: T_{ai} \in [TR, 1]$ and category $B: T_{ai} \in [0, TR]$. The number of requestors in the two categories is denoted as n_A and n_B .

Confirm Threshold t . According to the definition of the (t, n) threshold function and the requirements of access control security, requestors who have not reached the

Input: ID_x : ID of the target user
Output: s : the secret to share

- (1) Get the IDs of the top k friends of the target node: $ID_1, ID_2, ID_3, \dots, ID_k$;
- (2) $SUM_{ID} = \sum_{i=1}^k ID_i$;
- (3) if SUM_{ID} is prime then
- (4) $s = SUM_{ID}$
- (5) else
- (6) $s = \text{find_next_prime}(SUM_{ID})$;
- (7) end if

ALGORITHM 1: Friendships_transform.

system trust threshold cannot successfully query the target user's friendships. Since $T_{ai} < TR$, it is necessary to ensure that requestors in category B cannot successfully query the friendships of the target user. Thus, in each query process, $t = n_B + 1$.

Secret Distribution. The secret distributor D chooses a random t degree polynomial $a(x) = s + a_1x^1 + a_2x^2 + a_3x^3 + \dots + a_{t-1}x^{t-1} \pmod p$, $\alpha_j \in_R Z_p$, $a(0) = s$. D sends $s_i = a(i)$ to participants $P_i, i = 1, 2, \dots, n$.

Secret Reconstruction. n_B requestors in category B , who are arranged in descending order of comprehensive trust, submit the secret fragments s_i obtained in reverse order, and n_A requestors and n_B requestors are divided into n_A groups for secret reconstruction.

Assume that requestor P_i ($i = 1, 2, \dots, n$), who queries the friendships of the target user, is arranged in descending order based on the comprehensive trust of the target user n_a . Category A is $P_1, P_2, P_3, \dots, P_m$, and category B is $P_{m+1}, P_{m+2}, \dots, P_n$. Threshold $t = n_B + 1$. As shown in Table 3, category A can be divided into m groups to reconstruct secret s .

The comprehensive trust of the first requestor among the m groups of requestors who participate in the secret reconstruction is greater than the trust threshold set by the target user (i.e., only users trusted by the target user can successfully query the target's friendships). During each secret reconstruction process, the users $P_{m+1}, P_{m+2}, \dots, P_n$ who have not reached the comprehensive trust level threshold must submit their secret fragments $s_{m+1}, s_{m+2}, \dots, s_n$ obtained from D . Users $P_1, P_2, P_3, \dots, P_m$ will submit $s_{m+1}, s_{m+2}, \dots, s_n$. The secret fragment s_i ($i \in [1, m]$) is secretly reconstructed. The threshold $t = n_B + 1$ can ensure that even if $P_{m+1}, P_{m+2}, \dots, P_n$ constitute the group of submitted secret fragments, the secret cannot be successfully reconstructed.

4.3.3. Punishment Mechanism. Multiple malicious requestors query the friendships of users by coordinating their query order and query target via the friend search engine. The proposed mechanism further protects the privacy of the target users' friendships by setting the punishment mechanism. When the user who has inquired about the friendships of the target user causes the privacy leak, the comprehensive trust of the inquirers will be reduced, which

will make the next query impossible. Assume that before querying the friendships of the target user, the malicious requestors MR_1 and MR_2 are disguised as trusted nodes. If malicious requestors MR_1 and MR_2 have comprehensive trust T_{ai} , ($T_{ai} > T_t$), during the first query, the malicious requestor MR_1 can successfully reconstruct target node n_a 's friendships by secret fragments submitted by category B users and the secret fragments obtained from D . After the malicious requestor MR_1 obtains the query result and shares it with MR_2 , malicious requestor MR_2 can require the other nodes based on the query result of MR_1 . If the final query result causes the target node to expose the $k + 1$ th friend, then the system punishes all nodes that are secretly reconstructed, which reduces the trust value of the user nodes for the reconstructed secret to 1/2 of the original value. The trust decay function is expressed as follows:

$$T'_i = \frac{T_i}{2}. \quad (7)$$

Taking the attack in Section 3.2.3 as an example, assume that the trust threshold is 0.5 and the comprehensive trust of MR_1 and MR_2 is the maximum value of 1. According to the collusion attack strategy, N_3 's privacy will be violated.

When the friend search engine detects that the privacy of user N_3 is breached, it will reduce the trust of all users who have queried at this time to punish them. The trust value of MR_2 was originally 1. After the punishment, its comprehensive trust is reduced according to the trust decay function, and the comprehensive trust of malicious requestors MR_1 and MR_2 is reduced from 1 to 0.5. The comprehensive trust obtained from the target user is now lower than the trust threshold, and the next query cannot be performed.

5. Experiment

In this section, we experimentally verify the effectiveness of the proposed anticollusion attack strategy. Our experimental research includes synthetic datasets to verify the validity of the credibility calculations and three large-scale real-world datasets to verify the security of the anticollusion attack strategy.

5.1. Datasets. We generate random numbers that satisfy the previously described conditions of the credibility calculation method, including data on 1000 groups of user interactions,

TABLE 3: Groups to reconstruct secret s .

Group number	Group member
1	$P_1, P_n, P_{n-1}, \dots, P_{m+1}$
2	$P_2, P_n, P_{n-1}, \dots, P_{m+1}$
3	$P_3, P_n, P_{n-1}, \dots, P_{m+1}$
...	...
m	$P_m, P_n, P_{n-1}, \dots, P_{m+1}$

and verify the correctness of the trust calculations. In addition, we use three real-world social network datasets to verify the security of the anticollusion attack strategy.

5.1.1. Synthetic Dataset. A random probability function is used to fit users' interactions in OSNs. The setting standards for the time interval of interactions between users and the weights of the interaction events are different for each OSN. We select the interaction data within the time interval ($\Phi(t_i) = 0.367879$) among users in the synthetic dataset. The number of interactions is set to 50; the weights of the interaction events take values in the range $[1, 20]$; and the interaction evaluation takes values in the range $(0, 1]$ as an example to verify the rationality of the trust calculations, that is, $W_i \in [1, 20]$, $C_i \in (0, 1]$, and $n \in [1, 50]$. The trust between two users may exceed 1 and should be normalized.

5.1.2. Facebook Dataset. In [36], the data from <https://Facebook.com> capture the friendships among users, which can be modeled as undirected graphs.

5.1.3. Slashdot Dataset. In [37], Slashdot is a technology-related news website and a specific user community, where users can submit and edit news about the current main technology. In 2002, Slashdot launched the Slashdot Zoo function, which enables users to mark each other as friends or enemies. The network establishes links between two friends or enemies among Slashdot users. Therefore, the data in this dataset are directional. This article uses 2009 Slashdot data, and the Slashdot dataset is converted to an undirected graph to reflect users' friendships. Regardless of the direction of the connection between two nodes in the network, an edge is created in the undirected graph for these two nodes.

5.1.4. Gowalla Dataset. In [38], Gowalla is a location-based social networking site in which users share their location by signing in. The friendships collected from Gowalla are undirected. The complete dataset consists of 19, 591 nodes and 950, 327 edges. Due to data size limitations, this program selects only a portion of the data for testing.

We list the main attributes of each dataset in Table 4. The synthetic dataset is used to verify the rationality of the credibility calculations, and the remaining three datasets are used to verify the security of the proposed anticollusion attack strategy.

5.2. Strategy Analysis

5.2.1. Collusion Attack Strategy Analysis. According to the collusion attack model in [3], the collusion attack model has different probabilities of success for collusion attacks on popular and unpopular nodes. The probability of a successful conspiracy attack is mainly related to four factors, such as the degree d of the query node, the number of friends k allowed to be displayed, the layer of the friend relationship tree, and the rank r of the query user among the friends in that layer.

Given a user node of degree d , assume that the probability that one of his friends is ranked among the top k is k/d . Randomly choose the victim node N_0 and one of his top k friends $N_{0,i}$ with degree $d_{0,i}$; then, the probability that N_0 is among the top k friends of $N_{0,i}$ is

$$\begin{cases} \frac{k}{d_{0,i}}, & d_{0,i} > k, \\ 1, & d_{0,i} \leq k. \end{cases} \quad (8)$$

Simplify it as $\min(k/d_{0,i}, 1)$.

Assuming that the probability of N_0 becoming one of the top k friends of any of its friends is independent, the probability of N_0 becoming a popular node is p . Then, p is denoted by

$$p(N_0) = \prod_{i=1}^k \min\left(\frac{k}{d_{0,i}}, 1\right). \quad (9)$$

If N_0 is an unpopular node, the probability of easily destroying the privacy of the target user's friendships by direct query at the first level is

$$p(\text{Attack at layer}_1) = 1 - \prod_{i=1}^k \min\left(\frac{k}{d_{0,i}}, 1\right). \quad (10)$$

The number of collusion attackers required is

$$\text{Num}(\text{Attackers for unpop}) = 1 + k. \quad (11)$$

If N_0 is a popular node, according to the attack flow of compromising the privacy of popular nodes, a malicious attacker cannot directly make N_0 reveal the $k + 1$ th friend by querying its first layer friends. Therefore, the collusion attackers make the target user N_0 's first friend $N_{0,1}$ occupied and thus compromise the target user's friendship privacy by passively displaying it with probability:

$$p(\text{Attack through } N_{0,1}) = 1 - \prod_{i=1}^{r_{0,1}} \min\left(\frac{k}{d_{0,1,i}}, 1\right), \quad (12)$$

where $r_{0,1}$ is the ranking of N_0 among the friends of $N_{0,1}$.

5.2.2. Anticollusion Attack Strategy Analysis. According to the analysis of the attack success probability of the collusion attack and the total number of malicious attackers required, the probability of successful attack is equation (10), and the

TABLE 4: Social network dataset property.

Dataset	Synthetic dataset	Facebook	Slashdot	Gowalla
Vertices	1000	63731	82168	196591
Edges	8997	817090	948464	582533
Average degree	—	25.773	12.273	9.668

number of collusion attackers required is $k + 1$. The attack on popular nodes is more complicated. Generally, this attack cannot destroy the privacy of the target user's friendships by querying the first friend only, and the probability of destroying the target user's privacy by occupying the first friend $N_{0,1}$ of the target user is equation (12), where the number of conspiracy attackers required is at least $k + 2$.

In the friendship protection strategy against collusion attacks, the querier first needs to obtain a high level of trust through long-term interaction with the target user, and second, the querier needs to query the friends through the (t, n) threshold function. On the one hand, the anticollusion attack strategy sets a fully trusted querier to help with the query when there are fewer than n queriers. On the other hand, it avoids the situation where all of the malicious queriers have a high trust value.

In the worst case, the number of collusion attackers needed for unpopular nodes is only 2, which requires at least two queries, while the number of collusion attackers needed for popular nodes is 3, which requires at least three queries. When querying by the (t, n) threshold function, the worst case of the class A has $n_A - 1$ malicious attackers among the queriers. The subsequent security analysis will verify and analyze the security of the friendships privacy protection strategy with the worst-case number of malicious attackers against the collusion attack.

5.3. Performance Analysis. In this section, we analyze the rationality of the trust calculations and the security of the anticollusion attack strategy using (t, n) threshold function access control.

5.3.1. Credibility Calculation Rationality. In this article, we propose a trust measure based on the interaction behaviors between users. Considering the number of interactions between two users in a period of time, interaction evaluation, interaction event weight, and other factors, the direct trust degree is calculated by regulating the function. Based on the direct trust degree, the calculation methods of the recommended trust degree and comprehensive trust degree are derived. In this section, the rationality of the trust calculation method is verified by relevant experiments.

As Figure 4 shows, when the time period spanned by user interactions is 2, the time decay coefficient is approximately 0.135; while when the time period spanned by user interactions is 3, the time decay coefficient has dropped to less than 0.1. When the number of user interactions was 9, the ratio of the interaction number conditioning function (INCN) to the number of interactions (IN) was 0.105, while when the number of interactions was 10, the ratio of the interaction number conditioning function to the number of

interactions was 0.095. The number of time decays and the ratio of the interaction number conditioning function to the number of interactions were too low to show the more obvious experimental data results. Therefore, the number of interactions between users selected for the experiment ranged from 1 to 9, and the time period spanned by user interactions was selected as 1 or 2.

Based on random numbers, the values of direct trust and recommended trust are calculated by equations (1) and (2), respectively, and the value of the user's comprehensive trust is calculated by equation (3). We selected 1000 sets of data to prove the correctness of the trust calculations. The results are shown in Figure 5.

Figures 5(a)–5(c) show that the results of the direct trust, recommend trust, and comprehensive trust calculations, respectively, are normally distributed. In addition, they are in line with realistic expectations.

5.3.2. Security Analysis. To improve the security and usability of the friend search engine, we assume that OSN administrators can be fully trusted in regard to the friend search engine. When the number of requestors is less than the number of query requests, the administrators can help the requestors complete the query.

In this work, we compare the proposed anticollusion strategy with the original collusion attack. The security of the proposed strategy is verified and analyzed in four aspects, such as limit rate, the number of victims, the number of collusion attackers, and the success rate of the collusion attack. It is assumed that the original conspiracy attacker uses a minimum number of malicious attackers and can compromise the privacy of the target user with a minimum number of queries, and the probability of success of its attack is 1; i.e., the conspiracy attacker can successfully compromise the privacy of the target's friendships in each query.

Limit Rate (LR). The LR of the system is defined as the ratio of the number of users in category B to the number of all users, that is, the proportion of users who cannot successfully query in the friend search engine among all requestors. Based on equation (3) $OT_{ij} = u \cdot DT_{ij} + v \cdot RT_{ij}$ ($u + v = 1, u > v$), where $DT_{ij}, RT_{ij} \in [0, 1]$. The direct trust weight coefficient u is set to 0.6, and the trust threshold is set to 0.5, 0.6, 0.7, 0.8, and 0.9. A total of 1000 experiments are conducted to verify the LR of the proposed strategy.

Figure 6 shows the LR and trust threshold results of the strategy. The value of the direct trust weight coefficient u is 0.6. When the trust threshold is 0.5, the LR of the strategy is approximately 40%. When the trust threshold is 0.6, the LR increases to 80%. At 0.7, the LR increases to almost 100%.

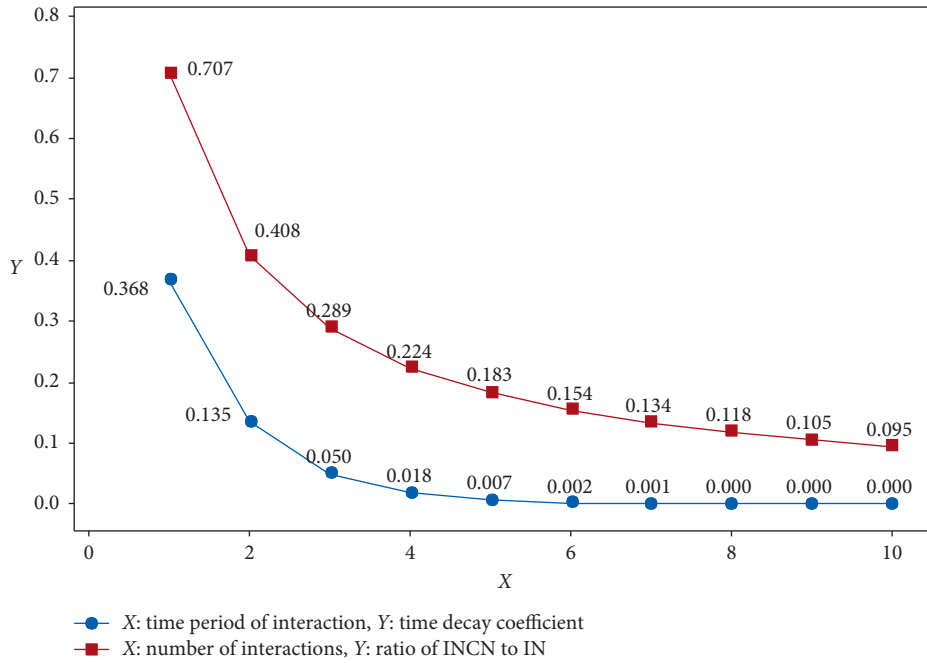


FIGURE 4: Schematic diagram of data selection.

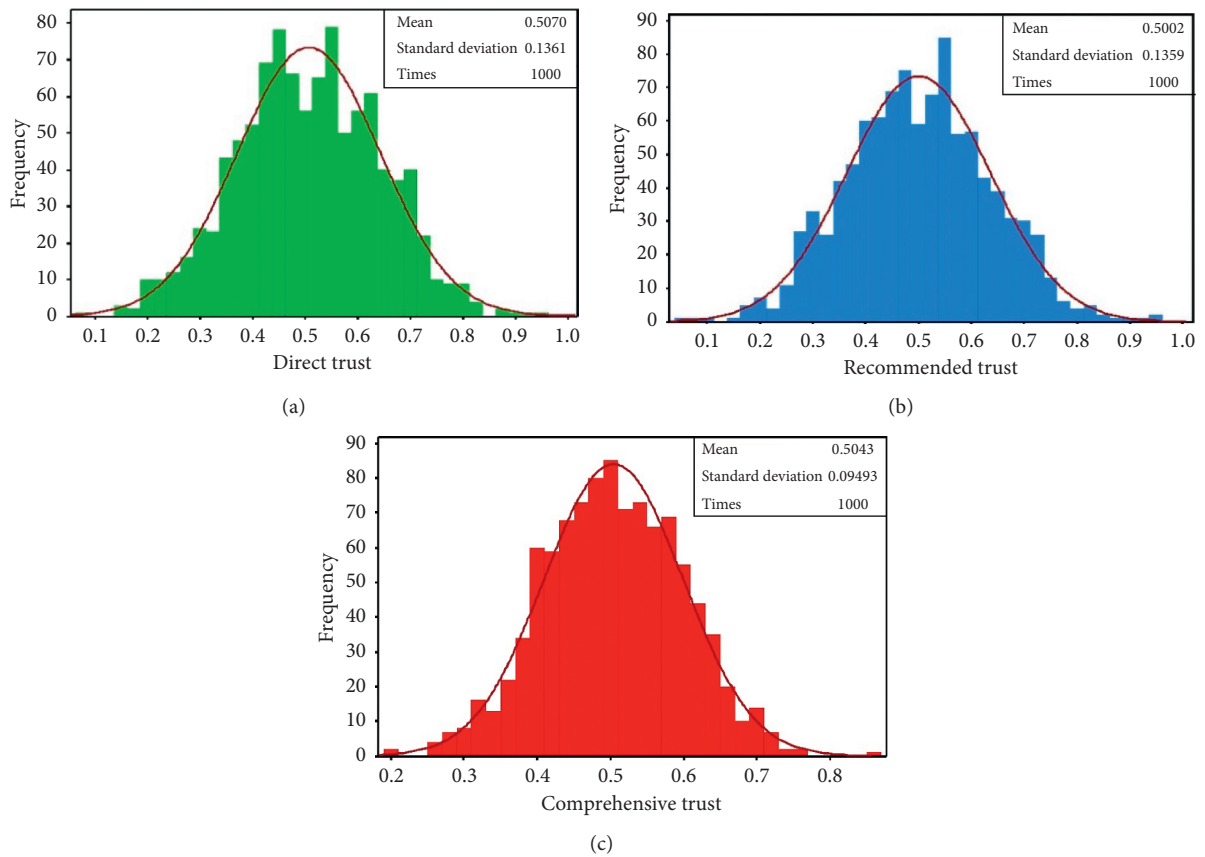


FIGURE 5: Trust values. (a) Direct trust. (b) Recommend trust. (c) Comprehensive trust.

Therefore, when the trust threshold is 0.7, almost no user reaches the trust threshold, and the friend search engine will not allow any querying. When the trust threshold is 0.6, 80%

of users in the OSN cannot reach the threshold. Thus, the number of requestors in the friend search engine is limited, and the safety of the friend search engine is increased.

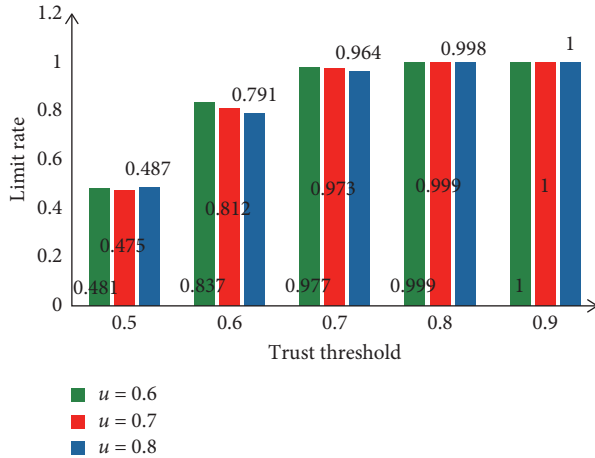


FIGURE 6: Limit rate under the trust threshold with $u = 0.6$.

Number of Victims. Consider the trust threshold of 0.5 as an example. Sixty percent of users can make normal queries. In the worst case of the friend search engine query, the number of malicious requestors is not limited, and malicious requestors can destroy the privacy of the target user via a one-time collusion attack at the first layer. The probability of successfully destroying the user's privacy is equation (10). The attack can destroy the privacy of 80% of the nodes in OSNs [3].

In a one-time collusion attack, the maximum number of malicious requestors is $n_A - 1$, and the collusion attack performs at least two queries. Thus, the probability of one collusion attack that destroys the target user's privacy at the first layer is

$$\left(\frac{n_A - 1}{n_A}\right)^2 \cdot p(\text{Attack at layer}_1). \quad (13)$$

When the trust threshold is set to 0.5 (lowest threshold), 40% of users' queries will be restricted. In this case, the anticollusion attack strategy can reduce the number of users whose privacy is breached by at least 47.9%. Accordingly, the number of users whose privacy is violated decreases. By comparing the Facebook, Gowalla, and Slashdot datasets, we obtain the results shown in Figure 7.

Due to the limitation of the trust threshold, the number of users whose privacy is breached is significantly reduced. The number of users whose privacy is breached in the Facebook and Slashdot datasets is reduced by approximately 20,000, while the number of users whose privacy is breached in the Gowalla dataset is reduced by approximately 60,000. In the three datasets, the number of users whose privacy has been violated will be reduced by at least 40%. The proposed strategy greatly reduces the number of users whose privacy is violated, which improves the privacy security of users in OSNs.

Number of Collusion Attackers. Based on the (t, n) threshold function, in the query process of the friend search engine, n inquirers are required to participate in

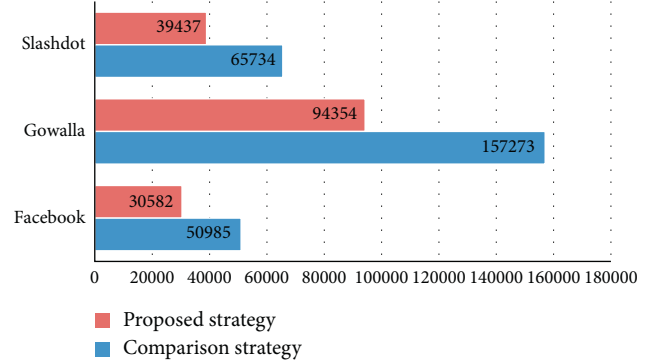


FIGURE 7: Comparison of the number of victims with a compromised strategy.

the query, and at least t requestors are required to perform secret reconstruction. Therefore, in a single query process, to ensure that malicious requestors can successfully query, it is necessary to ensure that t requestors are malicious requestors and that the comprehensive trust is higher than the trust threshold. In the best situation, two malicious requestors can destroy the privacy of the target user by making two queries. The total number of attackers required is $2n$, while in the comparison strategy, the number of inquirers required is only 2. Therefore, when the value of n set by the system is larger, more malicious attackers will be needed.

Figure 8 shows that the number of colluding attackers varies with the number of queries n . The number of attackers in the proposed strategy is twice that of the comparison strategy. Under the same conditions, the colluding attackers will need more entities or accounts to make queries with the proposed strategy.

Probability of a Successful Collusion Attack. Assume that malicious requestors who have not interacted with the target user in the OSN want to query the target's friendships. First, excellent long-term interactions with the target are needed to obtain the trust of the target. A successful collusion attack requires multiple malicious requestors to cooperate to coordinate their query order and target, and each malicious requestor can successfully query the friend list of the query target. Therefore, multiple malicious requestors need to maintain excellent interactions with users in the OSN, which will require colluding malicious requestors to spend a substantial amount of time disguising their intentions to obtain the trust of the target user.

Consider the successful collusion attack process in Table 2 as an example. The collusion attack was coordinated by four malicious requestors. MR_1 makes the first requests, and MR_2 determines the target to be queried based on the query results of MR_1 . MR_3 queries based on the query result of MR_2 . Thus, user $N_{0.2.2}$ will be "occupied," and the new friend $N_{0.2.3}$ of user $N_{0.2}$ can be queried. MR_4 makes a query based on the query result of MR_3 and obtains the $k + 1$ th friend of

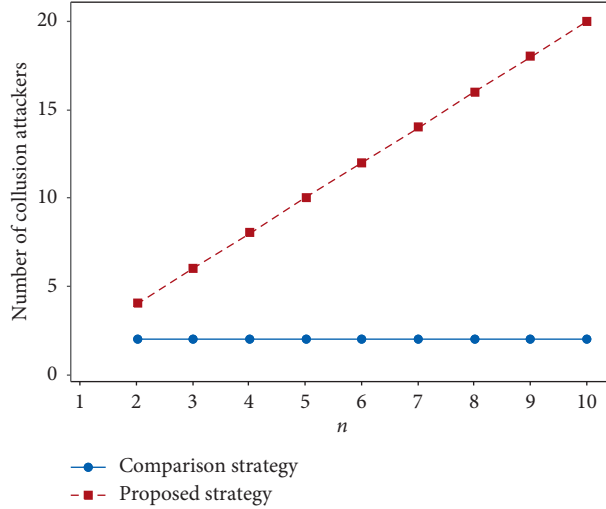


FIGURE 8: Number of collusion attackers.

N_0 , i.e., fourth friend $N_{0,4}$. The privacy of the friendships of user N_0 is destroyed.

Under (t, n) threshold function access control, four malicious requestors, i.e., $MR_1, MR_2, MR_3,$ and MR_4 , want to complete this query. First, they need to obtain the high trust of the target nodes, i.e., $N_0, N_{0,2}, N_{0,2,1}, N_{0,2,2},$ and $N_{0,2,3}$, and all four malicious requestors must have excellent long-term interactions with the target. If a malicious requestor cannot obtain the trust of the target, then $T_{ij} < T_t$, and the previously described attack cannot be successfully carried out. Therefore, a successful malicious attack by colluding attackers requires that all malicious requestors reach the trust threshold.

If malicious requestors already exist in the OSN and have interacted with the target user, this strategy restricts requestors whose trust level is below the trust threshold. A requestor cannot query the target user's friend list under (t, n) threshold function access control. Therefore, when the trust threshold is 0.5, 40% of users who do not reach the trust threshold will not be able to query. As described in the second part of this section, for the collusion attack strategy in [3], if (t, n) threshold function access control is not adopted, the probability that colluding attackers will successfully destroy a user's privacy is 1 for each query. In the (t, n) threshold secret sharing anticollusion attack strategy combined with trust, the comprehensive trust of the requestors who can successfully query the friendships of the target user must be higher than the trust threshold; that is, malicious requestors need to be in category A. Next, we take the trust threshold of 0.5 as an example to discuss the probability that colluding attackers will successfully destroy the privacy of a user's friendships under (t, n) threshold function access control.

If there is a collusion attack, the worst case is that there are enough colluding attackers, and the privacy of the target user is destroyed by just two queries. During a single query, the maximum number of malicious requestors is $n_A - 1$.

For unpopular nodes, the maximum probability of malicious requestors who make two requests is

$$\left[0.6^{(n_A-1)} \cdot \left(\frac{n_A-1}{n_A} \right) \right]^2. \quad (14)$$

For popular nodes, the maximum probability of malicious requestors who make three requests is

$$\left[0.6^{(n_A-1)} \cdot \left(\frac{n_A-1}{n_A} \right) \right]^3. \quad (15)$$

In Facebook, Gowalla, and Slashdot, we observe that regardless of whether a popular node or an unpopular node is considered, the number of malicious requestors required to conduct a successful collusion attack can reach 10,000, which is the best case of a successful collusion attack in the three datasets. Therefore, as Figures 9(a) and 9(b) show, in the case of $n_A \geq 2$, when there are at most $n_A - 1$ malicious requestors, the probabilities of successful collusion attacks for unpopular nodes and popular nodes are $p \leq 0.09$ and $p \leq 0.027$, respectively.

Figure 9 shows that when the number of malicious requestors is 2, the anticollusion attack strategy based on (t, n) threshold secret sharing can reduce the probability of a successful collusion attack from 1 to 0.09 and the probability of a successful conspiracy attack on popular nodes from 1 to 0.027. When the number of malicious requestors increases to 18, the anticollusion attack strategy reduces the probability of a successful collusion attack to 0. When the system trust threshold is higher, it is more difficult for malicious requestors to conduct collusion attacks.

Therefore, the trust-based SSS anticollusion attack strategy proposed in this work can substantially reduce the number of users whose privacy is compromised by means of credibility calculations, the trust threshold and the (t, n) threshold function. This strategy restricts user queries based on trust and uses the (t, n) threshold function of the SSS for access control. This strategy can also reduce the probability of successful collusion attacks, which has a significant effect on resisting collusion attacks and can protect the friendship privacy of users in OSNs.

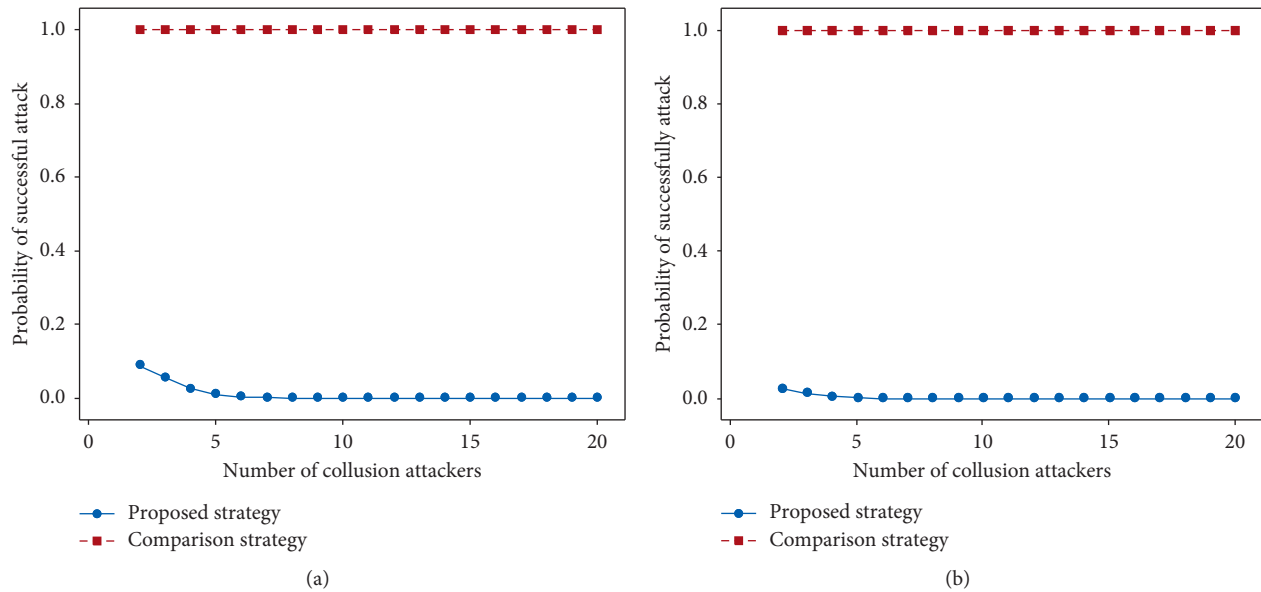


FIGURE 9: Comparison of the probabilities of a successful collusion attack based on the number of collusion attackers. (a) Unpopular nodes. (b) Popular nodes.

6. Conclusion

To address the problem of collusion attacks that compromise users' friendship privacy, we propose an anticollusion attack strategy that combines the trust metric and (t, n) threshold function. The trust metric is based on the interaction behaviors between users, and the calculation methods of direct trust, recommendation trust, and comprehensive trust are determined by considering the number of interactions, interaction time, interaction evaluation, and event weight. Meanwhile, by converting friendships into secrets and using the (t, n) threshold function to share and reconstruct the secrets, the conspiracy queries of malicious attackers are effectively restricted. The experimental results show that the proposed strategy can significantly reduce the probability of successful conspiracy attacks, reduce the number of victims, and protect the privacy of users' friendships while ensuring normal user queries.

Theoretically, this work simplifies the complex privacy protection of a user's friendships to the user's access control strategy in the friend search engine. This research starts by theoretically analyzing the calculation of trust between two users and applies the (t, n) threshold function to control querying in the friend search engine to protect the privacy of the user's friendships.

Overall, the proposed strategy can successfully decrease the probability of collusion attacks in friend search engines. Specifically, attacking the same number of users requires more attackers, and the number of users who violate the same number of attackers is greatly reduced.

Data Availability

The datasets 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 work was supported by the National Natural Science Foundation of China (61802106) and the Natural Science Foundation of Hebei Province (F2016201244). The authors of the article would like to express their gratitude to AJE for providing language assistance for this work.

References

- [1] finderman: <http://www.findermind.com/free-people-search-engines/>, 2021..
- [2] L. Na, "Privacy-aware display strategy in friend search," in *Proceedings of the 2014 IEEE International Conference on Communications ICC*, pp. 945–950, Sydney, Australia, June 2014.
- [3] L. Yuhong and L. Na, "Retrieving hidden friends: a collusion privacy attack against online friend search engine," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 833–847, 2019.
- [4] J. Morris, D. Lin, and A. Squicciarini, "Friendguard: a friend search engine with guaranteed friend exposure degree," in *Proceedings of the 24th ACM Symposium on Access Control Models and Technologies*, pp. 37–48, Toronto, Canada, June 2019.
- [5] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2019.
- [6] T. Junfeng, D. Ruizhong, and C. Hongyun, *Trusted Computing and Trust Management*, Science Press, Beijing, China, 2014.
- [7] Q. Weidong, H. Zheng, and L. Xiangxue, *Basics of Cryptographic Protocol*, Higher Education Press, Beijing, China, 2009.

- [8] O. Amusan, A. Thompson, T. Aderinola, and B. Alese, "Modelling malicious attack in social networks," *Network and Communication Technologies*, vol. 5, no. 1, Article ID 37, 2020.
- [9] A. Elyashar, S. Uziel, A. Paradise, and R. Puzis, "The chameleon attack: manipulating content display in online social media," in *Proceedings of the Web Conference 2020*, vol. 2, pp. 848–859, New York, NY, USA, April 2020.
- [10] B. DasGupta, N. Mobasher, and I. G. Yero, "On analyzing and evaluating privacy measures for social networks under active attack," *Information Sciences*, vol. 473, pp. 87–100, 2019.
- [11] B. Mei, Y. Xiao, R. Li, H. Li, X. Cheng, and Y. Sun, "Image and attribute based convolutional neural network inference attacks in social networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 869–879, 2020.
- [12] Y. Fu, W. Wang, H. Fu, W. Yang, and D. Yin, "Privacy preserving social network against dopv attacks," in *Proceedings of the International Conference on Web Information Systems Engineering*, pp. 178–188, Dubai, UAE, November 2018.
- [13] C. Liu, D. Yin, H. Li, W. Wang, and W. Yang, "Preserving privacy in social networks against label pair attacks," in *Proceedings of the 12th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2017)*, pp. 381–392, June 2017, <https://researchr.org/publication/wasa-2017>.
- [14] C. Sun, S. Y. Philip, X. Kong et al., "Privacy preserving social network publication against mutual friend attacks," in *Proceedings of the 2013 IEEE 13th International Conference on Data Mining Workshops*, pp. 883–890, Los Alamitos, CA, USA, December 2013.
- [15] K. S. Min, K. Y. Lee, J. B. Shin et al., "A privacy protection method for social network data against content/degree attacks," *IEICE - Transactions on Info and Systems*, vol. 95, no. 1, pp. 152–160, 2012.
- [16] B. Wang, J. Jia, L. Zhang et al., "Structure-based Sybil detection in social networks via local rule-based propagation," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 3, pp. 523–537, 2018.
- [17] Q. Zhou and G. Chen, "An efficient victim prediction for Sybil detection in online social network," *IEEE Access*, vol. 8, pp. 123228–123237, 2020.
- [18] Z. Xingwen and L. Hui, "Privacy preserving data-sharing scheme in content-centric networks against collusion name guessing attacks," *IEEE Access*, vol. 5, pp. 23182–23189, 2017.
- [19] K. Figl and C. Lehrer, "Privacy nudging: how the design of privacy settings affects disclosure in social networks," in *Proceedings of the 28th European Conference on Information Systems (ECIS): A Virtual AIS Conference*, Marrakech, Morocco, June 2020.
- [20] Z. Chen, Y. Tian, and C. Peng, "An incentive-compatible rational secret sharing scheme using blockchain and smart contract," *Science China Information Sciences*, vol. 64, no. 10, Article ID 202301, 2021.
- [21] J. Xiong, M. Zhao, M. Z. A. Bhuiyan et al., "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2019.
- [22] A. De Salve, R. Di Pietro, P. Mori et al., "A logical key hierarchy based approach to preserve content privacy in decentralized online social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 1, pp. 2–21, 2017.
- [23] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
- [24] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
- [25] V. Kumar and P. Pradhan, "Trust management," *International Journal of Service Science, Management, Engineering, and Technology*, vol. 11, no. 4, pp. 26–44, 2020.
- [26] L. Guo, C. Zhang, and Y. Fang, "A trust-based privacy-preserving friend recommendation scheme for online social networks," *Ieee Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 413–427, 2014.
- [27] L. Xu, C. Jiang, N. He et al., "Trust-based collaborative privacy management in online social networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 48–60, 2018.
- [28] K. Akilal, H. Slimani, and M. Omar, "A robust trust inference algorithm in weighted signed social networks based on collaborative filtering and agreement as a similarity metric," *Journal of Network and Computer Applications*, vol. 126, pp. 123–132, 2019.
- [29] K. Akilal, H. Slimani, and M. Omar, "A very fast and robust trust inference algorithm in weighted signed social networks using controversy, eclecticism, and reciprocity," *Computers & Security*, vol. 83, pp. 68–78, 2019.
- [30] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2018.
- [31] F. Facebook, "13 million US Facebook users don't change privacy settings," <http://www.zdnet.com/article/13-million-us-facebook-users-dontchange-privacy-settings/>.
- [32] J. Bonneau, J. Anderson, R. Anderson, and F. Stajano, "Eight friends are enough: social graph approximation via public listings," in *Proceedings of the 2nd ACM EuroSys Workshop on Social Network Systems*, pp. 13–18, SNS '09, New York, NY, USA, March 2009.
- [33] S. S. Malka, N. Li, and V. M. Doddapaneni, "A web application for studying collusion attacks through friend search engine," in *Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference*, pp. 388–393, Atlanta, GA, USA, 2016.
- [34] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [35] E. Dawson and D. Donovan, "The breadth of shamir's secret-sharing scheme," *Computers & Security*, vol. 13, no. 1, pp. 69–78, 1994.
- [36] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proceedings of the SIGCOMM 2009-Proc 2009 SIGCOMM Conf Co-Located Work Proc 2nd ACM Work Online Soc Networks*, pp. 37–42, WOSN 2009, Spain, Barcelona, August 2009.
- [37] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney, "Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.
- [38] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the The 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1w082–1090, ACM, San Diego, CA, USA, August 2011.

Research Article

A Secure Truth Discovery for Data Aggregation in Mobile Crowd Sensing

Taochun Wang ¹, Chengmei Lv,¹ Chengtian Wang,¹ Fulong Chen ²
and Yonglong Luo ¹

¹School of Computer and Information, Anhui Normal University, Wuhu, Anhui 241003, China

²Anhui Provincial Key Laboratory of Network and Information Security, Wuhu, Anhui 241003, China

Correspondence should be addressed to Taochun Wang; wangtc@nuaa.edu.cn

Received 12 April 2021; Revised 15 May 2021; Accepted 10 June 2021; Published 25 June 2021

Academic Editor: Jinbo Xiong

Copyright © 2021 Taochun Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of portable mobile devices, mobile crowd sensing systems (MCS) have been widely studied. However, the sensing data provided by participants in MCS applications is always unreliable, which affects the service quality of the system, and the truth discovery technology can effectively obtain true values from the data provided by multiple users. At the same time, privacy leaks also restrict users' enthusiasm for participating in the MCS. Based on this, our paper proposes a secure truth discovery for data aggregation in crowd sensing systems, STDDA, which iteratively calculates user weights and true values to obtain real object data. In order to protect the privacy of data, STDDA divides users into several clusters, and users in the clusters ensure the privacy of data by adding secret random numbers to the perceived data. At the same time, the cluster head node uses the secure sum protocol to obtain the aggregation result of the sense data and uploads it to the server so that the server cannot obtain the sense data and weight of individual users, further ensuring the privacy of the user's sense data and weight. In addition, using the truth discovery method, STDDA provides corresponding processing mechanisms for users' dynamic joining and exiting, which enhances the robustness of the system. Experimental results show that STDDA has the characteristics of high accuracy, low communication, and high security.

1. Introduction

With the rapid popularization of portable mobile sensing devices (such as smart phones and smart watches), which carry many sensors (gravity sensors, GPS, acceleration sensors, fingerprint, etc.), MCS has been extensively studied [1–4]. Participants with mobile sensing devices are encouraged to upload, analyze, and process their sensing data. After receiving the sensing data, the system is applied to all walks of life in society, such as transportation planning [5], environmental monitoring [6], and medical health [7]. For example, in MCS, participants upload the specific geographic location data of an object (such as supermarkets and schools) to the server, which analyzes and processes the data. And the obtained results are fed back to the corresponding application platforms. Then the platform utilizes these data to satisfy the needs of other participants, while enabling

participants to quickly and accurately locate the specific location of the required objects, and to facilitate the activities of participants.

Due to the unprofessionalism and mobility of participants, the sensing data uploaded by participants is often unreliable or even conflicting data. Moreover, malicious participants may upload outdated or wrong data, which possibly have serious consequences for decision-making. For example, getting misleading geographic location information on the application platform, ordinary participants miss the best viewing time for tourist attractions. In addition, in many applications, data needs to be obtained from multiple data sources, and multiple data sources may also provide conflicting information. For example, a natural event that may be observed and recorded by multiple laboratories, or a patient record composed of multiple different hospitals,

makes these pieces of data or information conflict with each other. Therefore, the service quality of MCS can be guaranteed by filtering out the incorrect sensing data and identifying the real information. Elimination of above-mentioned classification data conflict can be resolved by majority voting; that is, the most frequent information is considered to be the correct answer. For continuous data (e.g., height and weight), the mean/median value can be taken as the answer. The problem with voting or averaging method is that it assumes that the reliability of data from all sources is the same. Because normal participants continuously provide real and meaningful data, while malicious participants may generate biased or even false data, such traditional aggregation methods (such as voting and average) will not be able to get accurate aggregation results. In this case, in order to solve this problem, the truth discovery [7] approach, which is discovering truthful facts from unreliable or conflict information, has received extensive attention. The common principle of truth discovery is that the weight of the participant will be higher if the data provided by a participant is close to the aggregated result, and the reliability of the participant is higher and the data of participant will be counted more during the aggregation process if the participant's weight is higher. Based on this principle, the researchers have proposed multiple truth discovery methods to update the participant's weight and estimate the ground truth of each object.

However, the existing MCS faces serious privacy leakage issues which reduce the enthusiasm of participants. If the scheme based on truth discovery in MCS does not consider privacy, the server will obtain various types of information of participants, which may contain personal identity information and sensitive information such as phone number, home address, and health status. Attackers may take advantage of this sensitive information to conduct malicious deals. Based on this, our paper proposes a secure truth discovery for data aggregation in mobile crowd sensing (STDDA) in MCS. STDDA obtains final result by iteratively updating participant's weights and evaluating ground truth of each object. In order to protect data privacy, STDDA divides participant nodes into several clusters according to the location and number of participants. There are several participant nodes in each cluster which compute the corresponding secret random number according to the common parameters shared by the predecessor and successor nodes, while adding the secret random number to the sensing data to ensure data privacy. At the same time, the cluster head node uses secure sum protocol to fuse the sensing data in the cluster and sends it to the server which does corresponding storage and processing, so that the sensing data and weight of individual will not be known by the server, further ensuring the privacy of the participant's sensing data and weight. Using the truth discovery technology, STDDA gives the corresponding processing mechanism to the participant's failure exit and dynamic join, while enhancing the robustness of the system.

In summary, the contribution of our paper is summarized as follows:

- (1) STDDA not only accurately compute the final aggregation result and estimated ground truth but also protects the data and weight information of the participants. In addition, it greatly improves the calculation speed and reduces the communication overhead of the participants.
- (2) STDDA meets requests that participants fail to exit and join dynamically through cluster management and at the same time protects their data.
- (3) Finally, extensive experiments were conducted in the MCS, and the results verified that STDDA can generate accurate aggregate results while protecting the privacy of participant data and weights.

The rest of this article is arranged as follows. In Section 2, we discuss the related work of this article. Then, we describe the preliminaries and give the details of our proposed algorithm in Sections 3 and 4. In Section 5, we conduct a series of experiments and performance evaluation to demonstrate the claims given in this article. Finally, we make a conclusion in this article in Section 6.

2. Related Work

Recently, truth discovery is an effective method to obtain truth values of each object from many sensing data, which has received more and more attention [8–17]. TruthFind [8] first proposed the problem of truth discovery, which provides a probabilistic approach based on the following assumptions: different data sources are independent, so the unreliable pieces of information that appear on different data sources should be different from each other. Then, AcuSim [9] is suitable for Bayesian analysis, and CRH [12] is suitable for processing heterogeneous data. However, all the abovementioned truth discovery methods ignore important privacy issues and may lead to the disclosure of personal sensitive information. For example, in order to deal with heterogeneous data, a CRH [12] way with high precision and accuracy is proposed, but this method only takes into account the problem of work efficiency, and the protection of data privacy of participants is not within the scope of its research.

Once the user's privacy is leaked, such as home address and office address, malicious attackers may use this information to attack users, which will directly threaten users' property and life safety. Xiong et al. [18] proposed an edge-assisted privacy-preserving raw data sharing framework. The framework uses additional secret sharing technology to encrypt the original data into two ciphertexts and constructs two types of security functions. Tian et al. [19] proposed a secure key management based on blockchain solution (BC-EKM). They use secure cluster formation algorithm and secure node movement algorithm to realize key management.

At the same time, this damages the interests of users and restricts users' enthusiasm for participating in MCS. Privacy protection is a key factor in expanding and motivating MCS applications. Representative ways for solving various privacy issues include (1) anonymization [20, 21], i.e., removing

participant's identifying information during communication, (2) data disturbing [22], i.e., adding noise during communication to interfere with the identification of participant data, (3) cryptography or secure multiparty computation [23–25], which uses various encryption algorithms to protect participants' sensitive data or denoting multiple participants collaborating and cooperating under the condition of mutual distrust and outputting the calculation results.

In order to ensure the security of the truth discovery technology, researchers have recently proposed various privacy-oriented truth discovery schemes. For example, Miao et al. [26] first proposed a secure truth discovery scheme PPTD using the threshold Paillier cryptosystem [24] to protect the privacy of the sensing data and weights of participants. However, due to the complexity of the threshold Paillier cryptosystem, the participants undertake huge communication and computational overheads. To reduce the communication overhead of participants and improve system efficiency, Miao et al. [27] used homomorphic encryption to further propose a lightweight truth discovery privacy protection scheme, while designing dual noncollusive servers to achieve a lightweight privacy protection truth discovery system L2-PPTD. However, the premise assumption of the system is that the server does not have any collusion with other participants. Once collusion occurs, the privacy of the participants will be revealed. Zheng et al. [28] proposed a new system architecture that enables an encrypted truth discovery method to be implemented in MCS. In this system, participants send encrypted sensing data to the cloud, while performing CATD (Confidence-Aware Truth Discovery) in the encrypted domain, and the final encrypted inference truth value is sent to the requester for decryption. Xu et al. [29] proposed an EPTD framework to solve the problem that all participants must be online. However, this framework does not solve the problem of dynamic participation of participants, and the practicality is lacking. Therefore, it is a challenge to propose a practical privacy protection solution based on truth discovery. This scheme can solve the failure and join of participants and reduce the communication overhead and cost of participants.

3. Preliminaries

3.1. Network Model. MCS mainly includes three parts: server S , participants, and cluster head nodes CH. Among them, S is responsible for managing all participants and storing and processing the sensing data uploaded by participants. Participants accept the sensing tasks issued by the platform, collect the sensing data, and process it accordingly. CH manages the participant nodes in the cluster and processes related data. At the same time it has the role of ordinary participants. In STDDA, according to the location and number of participants, the network is divided into multiple clusters by the server S . Each cluster is composed of a CH and multiple participants. The CH forms a ring of all nodes in the cluster; that is, each node has a unique predecessor and successor node. The network topology is shown in

Figure 1. In each cluster, participants collect, process, and upload sensing data to CH. Then, CH aggregates all sensing data in the cluster and uploads them to S . Finally, S takes advantage of these data for various applications.

3.2. Truth Discovery. Truth discovery can effectively solve the problem of heterogeneous data information conflicts while extracting reliable information in MCS, where the object represents the description of the sensing task in the MCS, and the sensing data denotes the answers to the observations or questions collected by the participants. There are n participants, and a total of m objects require participants to collect data. x_j^i denotes the sensing data provided by the i th participant for the j th object. x_j^* represents the ground truth of j th object. w_i denotes the weight of i th participant, that is, the reliability of the i th participant. In addition, the goal of our article is to enable the server S to aggregate the sensing data of each participant $\{x_j^i\}_{i,j=1}^{m,n}$ and then accurately estimate ground truth of each object $\{x_j^*\}_{j=1}^m$, at the same time guaranteeing sensing data (i.e., $\{x_j^i\}_{i,j=1}^{m,n}$) and weights (i.e., $\{w_i\}_{i=1}^n$) are not known by other parties.

At present, existing truth discovery algorithms can generally be summarized in two procedures: weight update and truth evaluation. Before the weight is updated, the estimated ground truth of each object is first randomly initialized by the server S , and the weight and the estimated ground truth are updated iteratively until the convergence conditions are satisfied.

Weight update: it is assumed that the estimated ground truth of each object is fixed. Usually, the weight of each participant can be obtained as follows:

$$w_i = f\left(\sum_{j=1}^m d_{\text{ist}}(x_j^i, x_j^*)\right), \quad (1)$$

where f represents a monotonically decreasing function, and $d_{\text{ist}}(\cdot)$ represents the distance function between the sensing data and the estimated ground truth of participant. Since the CRH algorithm proposed has good practical performance, our paper uses the CRH algorithm to update the weight:

$$w_i = \log\left(\frac{\sum_{i=1}^n \sum_{j=1}^m d_{\text{ist}}(x_j^i, x_j^*)}{\sum_{j=1}^m d_{\text{ist}}(x_j^i, x_j^*)}\right), \quad (2)$$

where the distance function $d_{\text{ist}}(\cdot)$ is selected according to the application environment. This article considers the two most common data types (continuous data and categorical data) in the actual application of MCS.

In the continuous data (such as height and weight), the distance function $d_{\text{ist}}(\cdot)$ can be described as

$$d_{\text{ist}}(x_j^i, x_j^*) = \frac{(x_j^i - x_j^*)^2}{\text{std}_j}, \quad (3)$$

where std_j represents the standard deviation of the sensing data based on object j .

In the categorical data (such as gender and weather), this paper uses the vector $x_j^i = (0, \dots, 1(q\text{th}), \dots, 0)^T$ to represent

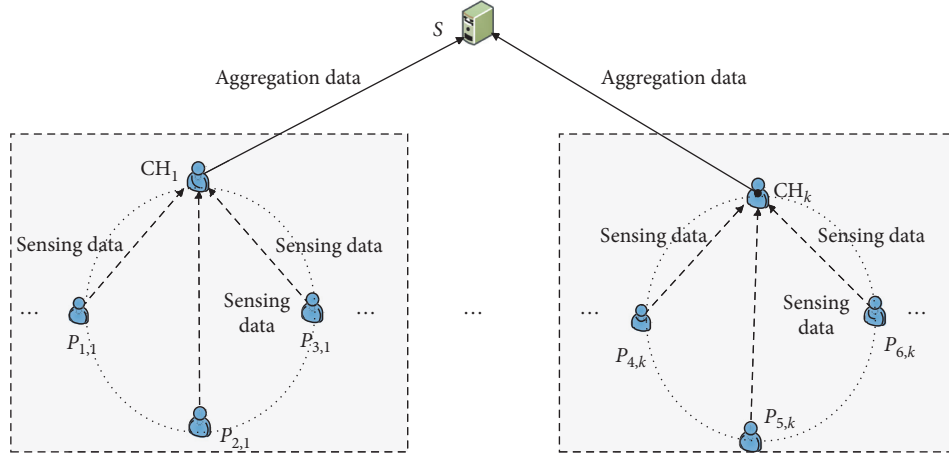


FIGURE 1: Topology of networks.

the q th choice of the i th participant based on the object j , and the calculation of $d_{ist}(\cdot)$ is

$$d_{ist}(x_j^i, x_j^*) = (x_j^i - x_j^*)^T. \quad (4)$$

Truth estimate: it is assumed that the weight of each participant is fixed. The ground truth of the j th object is estimated as

$$x_j^* \leftarrow \frac{\sum_{i=1}^n w_i x_j^i}{\sum_{i=1}^n w_i}. \quad (5)$$

Finally, the estimated ground truth of each object is obtained by iterating the above two procedures until the convergence condition is satisfied. The general truth discovery procedure can be described by Algorithm 1.

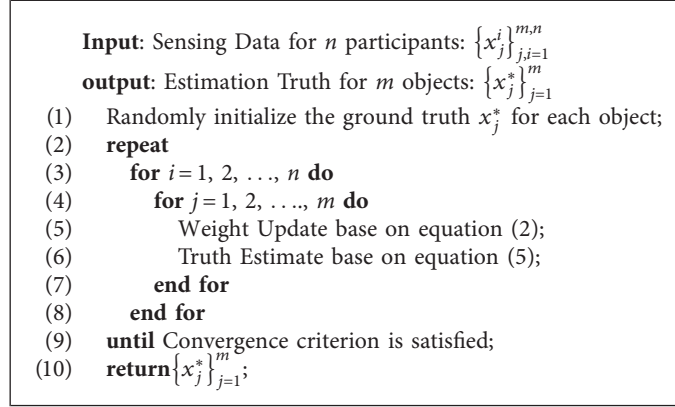
3.3. Attack Type. Attacks in MCS mainly include external attacks and internal attacks. (1) External attacks: since the information in MCS is transmitted wirelessly, the most common attack method is network eavesdropping to destroy data confidentiality. Our article assumes that the attacker can eavesdrop the entire network. (2) Internal attack: internal nodes or server S tries to obtain information to deduce the privacy information of other participants in MCS under the premise of completing the agreement. For example, the participant/server S tries to deduce the privacy information (such as location) of other participants on account of curiosity or interest. Our article adopts a semihonest model; that is, all parties of the MCS strictly implement the agreement, but the members retain the data obtained during the execution of the agreement and try to derive the privacy information of other members. Finally, our article, which can prevent collusion attacks (e.g., participants collude with S), uses data encryption to resist external attacks, so this article focuses on preventing internal attacks.

4. Security Truth Discovery

STDDA can accurately estimate the ground truth of each object based on the sensing data transmitted by participants.

At the same time, in order to ensure the security of sensitive information, the sensing data and weight of participants are not obtained by other participants and server S . We first introduce the idea of STDDA algorithm, second describe the process of STDDA algorithm, and finally discuss and analyze the dynamics and security of the network.

4.1. STDDA Framework. In STDDA, participants are divided into several clusters by server S according to the location and number of participants. All processing is in units of clusters, and the process of each cluster is divided into three steps. (1) Initialization: S provides initial estimated ground truth of each object for each participant node. Then participant nodes compute the corresponding secret random numbers based on the common parameters shared by the predecessor and successor nodes. (2) Secure weight update: based on the sensing data and the initial ground truth provided by S , each participant calculates D_i , which is the sum of object distance function, while encrypting and transmitting it to CH. After obtaining all the ciphertext data in the cluster, CH uses the secure sum protocol to fuse ciphertext data to get D_C , which is the sum of object distance function of the cluster, and uploads it to S . Finally S aggregates all cluster data to obtain D , which is the sum of object distance function of all participants in the entire system, and then broadcasts D to all participants to update the weight. (3) Secure truth evaluation: participant P_i encrypts the weight W_i and $W_i O_i$, the product of weight and sensing data, and transmits them to CH. Then CH takes advantage of the secure sum protocol to get W_C , which is the sum of weight of cluster, and $W_C O_C$, which is the product of weight and sensing data of cluster. Next, CH encrypts and uploads them to S . At the same time, S aggregates W_C and $W_C O_C$ to obtain W , the sum of weight of all participants, and $W O$, the sum of product of the weight and the sensing data of all participants in the entire system. Finally, the ground truth evaluation is performed until the convergence condition is satisfied; otherwise steps (2) and (3) are repeated. The procedure can be shown in Figure 2.



ALGORITHM 1: Truth discovery process.

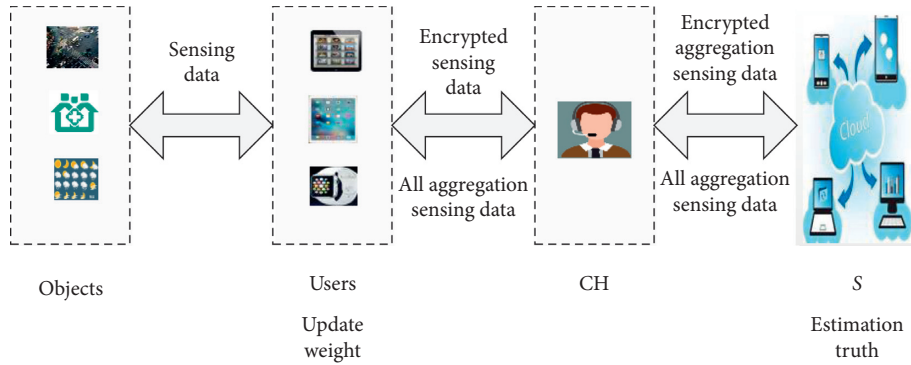


FIGURE 2: Flowchart of secure weight update and secure truth estimation.

4.2. STDDA Mechanism. In STDDA, it is assumed that n ($\{P_1, P_2, \dots, P_n\}$) participants participate in MCS and collect sensing data of m objects. Participants are divided into t clusters by server S . There are k ($k = n/t$ and $k \geq 3$) participants in each cluster, and some participant is randomly selected as the cluster head node (CH), and each cluster head node CH_i is assigned a secret key k_i . All participant nodes are formed into a ring; that is, each node has a unique precursor and successor node. For example, CH is P_1 ; that is, its precursor and successor nodes are P_k and P_2 . P_i node precursor and successor nodes are P_{i-1} and P_{i+1} , respectively. On this basis, the following specifically explains the initialization of the algorithm, the secure weight update, and truth evaluation.

4.2.1. Initialization. The server S generates initialization ground truth of all objects $\{x_j^*\}_{j=1}^m$ and broadcasts them to each participant P_i , at the same time, generating two q -order multiplication groups G_1, G . p, q are large prime numbers with the same number of digits, and q is divided by $p - 1$. At the same time $g_1 = h^{(p-1/q)} \bmod p$ is the generator of G_1 , where h is a random number. Moreover, $g_2 = g_1^p \bmod p^2$ is the generator of G .

Within each cluster, the node P_i randomly generates an integer $u_i \in Z$ and computes the common parameter $\beta_i = g_2^{u_i} \bmod p^2$. Then, β_i is shared with its predecessor and

successor nodes P_{i-1} and P_{i+1} . After a round of exchanges, P_i calculates the secret random number $R_i = (g_2^{u_{i+1}} / g_2^{u_{i-1}})^{u_i} \bmod p^2$, as shown in Figure 3.

4.2.2. Secure Weight Update. The main process of secure weight update is divided into four parts. (1) Participants compute $D_i = \sum_{j=1}^m d_{ist}(x_j^i, x_j^*)$, which is the sum of object distance function. It is encrypted and transmitted to the cluster head node CH. (2) CH fuses the ciphertext data to get the sum of object distance function of the cluster D_C . It is encrypted and transmitted to the server S . (3) S gets D and broadcasts it to the participants. (4) All participants complete the weight update. When the participant P_i calculates the sum of object distance function $D_i = \sum_{j=1}^m d_{ist}(x_j^i, x_j^*)$ between the sensing data and the evaluation ground truth, the distance function $d_{ist}(\cdot)$ calculation methods of continuous data and categorical data are different. So, they need to be considered separately in the calculation. For categorical data, $d_{ist}(\cdot)$ is simply computed according to equation (4). For continuous data, the $d_{ist}(\cdot)$ is calculated according to equation (3), which needs to first compute the std of the sensing data, which is standard deviation. Since the std calculation is performed only once in the entire algorithm, it is not included in the iterative process. Therefore, this section first introduces the general steps (Step 1–Step 4) of all data types in the weight update and then introduces the

calculation process of the std_j in continuous data, which is the standard deviation of object j . See Step 5 for details.

Step 1 (each participant P_i encryption): P_i receives the evaluation ground truth sent by the server S (the first round is a random value generated by the S or a specific value). Then, P_i computes and encrypts D_i to form a ciphertext $E(D_i)$ as follows. At the same time, $E(D_i)$ is transmitted to the corresponding CH:

$$E(D_i) = (1 + p \times D_i) \times R_i \bmod p^2. \quad (6)$$

Step 2 (CH fusion): we can derive equation (7) from literature [30], where p represents a large prime number:

$$\begin{aligned} \prod_{i=1}^n (1 + p)^{D_i} &= \prod_{i=1}^n (1 + p \times D_i) \\ &= \left(1 + p \sum_{i=1}^n D_i \right) \bmod p^2. \end{aligned} \quad (7)$$

After receiving $E(D_i)$ in the cluster (including its own ciphertext), CH performs the calculation as shown in equation (8), according to equation (7):

$$\begin{aligned} E_C^D &= \prod_{i=1}^k E(D_i) \bmod p^2 \\ &= \prod_{i=1}^k (1 + p \times D_i) \times R_i \bmod p^2 \\ &= \prod_{i=1}^k (1 + p \times D_i) \times \left(\frac{g_2^{u_{i+1}}}{g_2^{u_{i-1}}} \right)^{u_i} \bmod p^2 \\ &= \prod_{i=1}^k (1 + p \times D_i) \times g_2^{u_{i+1} \times u_i - u_{i-1} \times u_i} \bmod p^2 \\ &= \left(1 + p \sum_{i=1}^k D_i \right) \times g_2^{\sum_{i=1}^k u_{i+1} \times u_i - u_{i-1} \times u_i} \bmod p^2 \\ &= \left(1 + p \sum_{i=1}^k D_i \right) \bmod p^2, \end{aligned} \quad (8)$$

where $u_{k+1} = u_1$ and $u_0 = u_k$. In order to ensuring accurate results, p needs to be large enough. CH gets $D^{C_y} = \sum_{P_i \in C_y} D_i$, which is the sum of object distance function of k participants in the cluster, based on $(E_C^D - 1)/p = \sum_{i=1}^k D_i \bmod p$, while using the secret key k_i to form ciphertext $E_{k_i}(D^{C_y})$. Finally, the ciphertext is uploaded to the server S .

Step 3 (the server S aggregation): after receiving all the data uploaded by CH, S decrypts and aggregates the cluster data to obtain $D = \sum_{i=1}^n D_i = \sum_{y=1}^t D^{C_y}$, which is

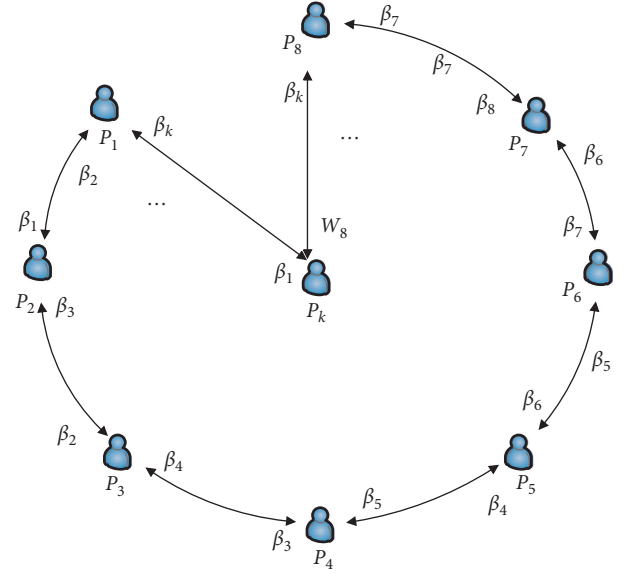


FIGURE 3: Secret random number in the setup.

the sum of object distance function of n participants in the entire system, while broadcasting D to all participants for weight update.

Step 4 (weight update): after P_i receives the D sent by S , the weight W_i is updated according to (2) as

$$w_i = \log\left(\frac{D}{D_i}\right). \quad (9)$$

Step 5: the standard deviation std_j computing

- ① The ciphertext of P_i 's sensing data based on the j th object is $E(x_j^i) = (1 + p \times x_j^i) \times R_i \bmod p^2$ and is transmitted to the CH of the cluster where P_i is located.
- ② After receiving $E(x_j^i)$ of all nodes in the cluster (including its own ciphertext), according to (7), the CH computes $\sum_{i=1}^k x_j^i$, which is the sum of the sensing data of k participants in the cluster based on the object j , and adopts the secret key k_i to form $E_{k_i}(\sum_{i=1}^k x_j^i)$, while uploading it to server S .
- ③ After receiving the data uploaded by CH, the server S decrypts and aggregates all cluster data to obtain $\sum_{i=1}^n x_j^i = \sum_{t=1}^t \sum_{i=1}^k x_j^i$, which is the sum of sensing data of n participants in the system based on object j . Then S calculates the average value $\bar{x}_j = \sum_{i=1}^n x_j^i / n$ based on the sensing data of object j and sends it to all participants.
- ④ After receiving \bar{x}_j , the participant P_i calculates $(x_j^i - \bar{x}_j)^2$. It is encrypted to $E((x_j^i - \bar{x}_j)^2) = (1 + p \times (x_j^i - \bar{x}_j)^2) \times R_i \bmod p^2$ and transmitted to CH.
- ⑤ The CH calculates $\sum_{i=1}^k (x_j^i - \bar{x}_j)^2$ of the k participants in the cluster and encrypts and uploads it to S . After receiving all the data $\text{SUM} = \sum_{i=1}^n (x_j^i - \bar{x}_j)^2 = \sum_{t=1}^t \sum_{i=1}^k (x_j^i - \bar{x}_j)^2$ uploaded by CH, S can obtain

and calculate the standard deviation $\text{std}_j = \sqrt{\text{SUM}/n}$ of participant's sensing data based on object j according to SUM.

4.2.3. Secure Truth Evaluation. The secure truth evaluation phase can be divided into three parts: (1) Participants compute WO_i , which is the product of weight and sensing data, and the weight W_i . They are transmitted to CH. (2) The ciphertexts of product and weight are fused by CH separately, while being encrypted and uploaded to the server S . (3) S obtains the sum of weight and product of all participants, respectively, and finally completes the truth evaluation. The specific process is show as follows.

Step 1 (each participant P_i encryption): P_i computes the WO_i , which is the product of weight and sensing data according to the obtained weight W_i , encrypts W_i and WO_i to form ciphertext $E(W_i) = (1 + p \times W_i) \times R_i \bmod p^2$ and $E(WO_i) = (1 + p \times WD_i) \cdot R_i \bmod p^2$, and then transmits them to the CH.

Step 2 (CH fusion): after receiving the ciphertext of all nodes in the cluster (including its own ciphertext), the CH performs calculations such as (10) and (11) in combination with (7):

$$\begin{aligned} E_C^W &= \prod_{i=1}^k E(W_i) \bmod p^2 \\ &= \prod_{i=1}^k (1 + p \times W_i) \times \left(\frac{g_2^{u_{i+1}}}{g_2^{u_{i-1}}} \right)^{u_i} \bmod p^2 \quad (10) \\ &= \left(1 + p \sum_{i=1}^k W_i \right) \bmod p^2, \end{aligned}$$

$$\begin{aligned} E_C^O &= \prod_{i=1}^k E(WO_i) \bmod p^2 \\ &= \left(1 + p \sum_{i=1}^k WO_i \right) \cdot \left(\frac{g_2^{u_{i+1}}}{g_2^{u_{i-1}}} \right)^{u_i} \bmod p^2 \quad (11) \\ &= \left(1 + p \sum_{i=1}^k WO_i \right) \bmod p^2. \end{aligned}$$

CH computes $E_C^W - 1/p$ and $E_C^O - 1/p$ to obtain $W^{C_y} = \sum_{P_i \in C_y} W_i$ and $WO^{C_y} = \sum_{P_i \in C_y} WO_i$, which are the sum of weight and product of the k participants in the cluster, and then uses the secret key k_i to form ciphertexts $E_{k_i}(W^{C_y})$ and $E_{k_i}(WO^{C_y})$, uploading them to the server S .

Step 3. Truth Evaluation. After receiving all the data uploaded by the CH, S decrypts and aggregates all the cluster data to obtain $W = \sum_{i=1}^n W_i = \sum_{y=1}^t W^{C_y}$, which is the sum of weight of n participants in the entire system, and $WO = \sum_{i=1}^n WO_i = \sum_{y=1}^t WO^{C_y}$, which is the sum of the product of the weight and the sensing

data in the entire system. Finally the ground truth of each object is estimated based on (3) as

$$x_j^* \leftarrow \frac{WO}{W}. \quad (12)$$

The algorithm iteratively and securely updates participants' weight and estimates ground truth of object until the convergence condition is satisfied. The server S finally obtains the estimated ground truth of each object j as Algorithm 2, where steps 1–3 are the initialization procedure. Step7–10 are secure weight update process, and steps 11–13 are secure truth evaluation procedure.

4.3. Participant Dynamics. Because of the unprofessional nature of MCS participants and the characteristics of wireless transmission, it is often the case that participants are often (temporarily) invalid or newly join. In order to increase the robustness of the system, STDDA gives the corresponding processing mechanism which solves the failure exit or dynamic join of participant nodes.

4.3.1. Node Join. In order to encourage users to participate in MCS, STDDA allows new nodes to participate in the system and enhances the usability of the system. When the node P_j wants to join the MCS system, the node P_j first sends a join request message to the server S and S verifies its identity and determines whether the number of cluster nodes is less than the upper limit k . If it exists, select the cluster C_y according to the number of nodes in the cluster and the position of P_j and then forward the request message to the cluster head node CH_y . After CH_y receives the message, CH_y randomly informs two consecutive nodes in the cluster C_y (without loss of generality, such as nodes P_i , P_{i+1}) as the predecessor and successor nodes of P_j . At the same time, the nodes P_i , P_{i+1} and P_j update the public parameters ($\beta_i^{C_y}$, $\beta_{i+1}^{C_y}$, $\beta_j^{C_y}$) and secret random numbers ($R_i^{C_y}$, $R_{i+1}^{C_y}$, $R_j^{C_y}$). After the above work is completed, P_j will participate in the next truth discovery process. If the number of nodes in the existing cluster reaches the upper limit ($=k$), the server randomly selects the cluster C_y and randomly selects a ($2 \leq a < k$) nodes in the cluster to establish a new cluster N_y with the newly added node. Updating the public parameters and secret random numbers are added to the next truth discovery process. The procedure can be described by Algorithm 3.

4.3.2. Node Invalid. When the node P_j fails to transmit data normally due to its own aspiration or software and hardware problems, STDDA needs to perform invalidation processing on the node P_j . This section considers two situations of node failure:

- ① Active failure: the node sends a leave request message to the server S before the node fails and applies to leave the cluster C_y . If the number of nodes of the

Input: n participants, m objects, sensing data for n participants base on m objects: $\{x_j^i\}_{j=1}^{m,n}$

output: Estimation ground truths for m objects: $\{x_j^*\}_{j=1}^m$

- (1) Server S randomly initializes the estimated ground truth x_j^* for each object and sends to n participants;
- (2) P_i randomly produces a integer $u_i \in Z_p$ and calculates the public parameters W_i , while sharing W_i with the precursor and successor nodes;
- (3) After a round of swapping, P_i computes secret random number R_i ;
- (4) **repeat**
- (5) **for** $i = 1, 2, \dots, n$ **do**
- (6) **for** $j = 1, 2, \dots, m$ **do**
- (7) P_i calculates $D_i = \sum_{j=1}^m d_{ist}(x_j^{P_i}, x_j^*)$, then encrypting them forms ciphertext $E(D_i) = (1 + p \times D_i) \times R_i \bmod p^2$ and sending $E(D_i)$ to CH;
- (8) CH fuses $E(D_i)$, which is transmitted by the P_i in the cluster based on the secure sum protocol, to obtain $D^{C_y} = \sum_{P_i \in C_y} D_i$, and uploads it as ciphertext to S by using the secret key k_i ;
- (9) S decrypts and aggregates all the cluster data to obtain $D = \sum_{i=1}^n D_i = \sum_{y=1}^t D^{C_y}$ and sends them to P_i ;
- (10) After receiving D sent by S , P_i update the W_i according to equation (9);
- (11) P_i calculates ciphertext $E(W_i) = (1 + p \times W_i) \times R_i \bmod p^2$ with $E(WO_i) = (1 + p \times WD_i) \cdot R_i \bmod p^2$ respectively and sends them to CH;
- (12) CH fuses $E(W_i)$ and $E(WO_i)$ based on the secure sum protocol to obtain $W^{C_y} = \sum_{P_i \in C_y} W_i$ with $WO^{C_y} = \sum_{P_i \in C_y} WO_i$, while uploading them as ciphertext to S by using the secret key k_i ;
- (13) S decrypts and aggregates all the cluster data to obtain $W = \sum_{i=1}^n W_i = \sum_{y=1}^t W^{C_y}$ with $WO = \sum_{i=1}^n WO_i = \sum_{y=1}^t WO^{C_y}$, and estimates the ground truths for m objects according to equation (12);
- (14) **end for**
- (15) **end for**
- (16) **until** Convergence criterion is satisfied;
- (17) **return** $\{x_j^*\}_{j=1}^m$;

ALGORITHM 2: Truth discovery process.

- (1) Denoting k_{C_y} is the number of nodes in the cluster C_y ;
- (2) $P_j \rightarrow S$; // P_j sends a request to join message to server S
- (3) if $(\exists k_{C_y} < k)$
- (4) S selects C_y ;
- (5) $S \rightarrow CH_y$; // S forwards the join request to the cluster head node CH_y
- (6) $CH_y \rightarrow P_j$;
- (7) $CH_y \rightarrow P_{i+1}$;
- (8) Denoting k_{C_y} is the number of nodes in the cluster C_y ;
- (9) $P_j \rightarrow S$; // P_j sends a request to join message to server S
- (10) if $(\exists k_{C_y} < k)$
- (11) S selects C_y ;
- (12) $S \rightarrow CH_y$; // S forwards the join request to the cluster head node CH_y
- (13) $CH_y \rightarrow P_j$;
- (14) $CH_y \rightarrow P_{i+1}$;
- (15) $u_j = \text{random}()$, $u_j \in Z_p$; // P_j randomly generates an integer
- (16) $\beta_i^{C_y}, \beta_j^{C_y}, \beta_{i+1}^{C_y}$; // updating the public parameters
- (17) $R_i^{C_y}, R_j^{C_y}, R_{i+1}^{C_y}$; // updating secret random numbers
- (18) else
- (19) establish a new cluster N_y ;
- (20) $N_y: \beta_{N_y}^{C_y}$;
- (21) $N_y: R_j^{C_y}$;
- (22) end if

ALGORITHM 3: Node join.

cluster C_y after P_j leaves is less than 3, the cluster is disbanded. And the remaining nodes are added to other clusters according to Algorithm 3. If the number of nodes in the cluster C_y after P_j leaves is

greater than 3, the cluster head node CH_y notifies P_j 's predecessor node P_{j-1} and successor node P_{j+1} to update the public parameters and secret random numbers, while processing to the next iteration.

- ② Passive failure: node P_j has sent relevant data, but the phenomenon of data loss occurs during the transmission. That is, the receiver has not received the message sent by P_j within the specified time. STDDA adopts a fast retransmission mechanism to solve this type of passive failure problem. Its main idea is that when the receiver receives every piece of data, it needs to reply with an acknowledgement ACK (value 1). When the receiver does not receive the data within the specified time, it sends a redundant ACK (value 0) to the node. STDDA selects 3 redundant ACKs as the threshold. Specifically, after the node P_j continuously receives 3 redundant ACKs, it immediately retransmits the data that has not been received by the other party. When the receiver has not received the sender's data within the specified time after sending 3 redundant ACKs, it is determined that the sender is passively invalid. The server can determine the number of remaining nodes in the cluster according to the node failure situation ①, while updating the public parameters and secret random numbers of the relevant nodes, so that the next iteration can be performed normally.

4.4. Security Analysis. We will conduct a theoretical analysis of the security of the STDDA algorithm in this section. Since attacks can be divided into external attacks and internal attacks according to the source in MCS, this chapter will conduct a theoretical analysis of security from both external and internal attacks.

4.4.1. External Attack. External attacks are attacks initiated by malicious nodes outside the network. The most common attack method is network eavesdropping. This article assumes that the attacker can conduct network-wide eavesdropping.

Theorem 1 (under honest but curious setting). *During the execution of the STDDA algorithm, the sensing data and weight of participant can resist theft attacks.*

Proof. In this article, we prove the participants' sensing data and weight against eavesdropping attacks from both the participants and the server. (1) Participants: In the secure weight update procedure, since the transmitted sensing data is encrypted by participants, the external attacker eavesdrops to obtain the encrypted ciphertext $E(D_i) = (1 + p \times D_i) \times R_i \pmod{p_2}$, so the attacker must infer the large prime number p and the secret random number R_i to get the plaintext D_i . However, the secret random number R_i is only known by the participant, so the attacker cannot eavesdrop on the ciphertext (D_i) to infer the plaintext D_i . Similarly, in the secure truth evaluation procedure, the transmitted weight is encrypted by participants, and the attacker cannot get the plaintext of weight. In addition, in order to further increase data privacy, participants update the secret random number R_i after N rounds of transmission. (2) Server: In the secure weight update procedure, the attacker eavesdrops on the

sum of the object distance D ($D = \sum_{i=1}^n D_i = \sum_{t=1}^t \sum_{i=1}^k D_i$) of n participants transmitted by the server. Because D is aggregated data, the attacker cannot determine D is obtained by fusion of which nodes; that is, the sensing data of any node cannot be derived. In summary, the participant's sensing data and weight can prevent external eavesdropping attacks. \square

4.4.2. Internal Attack. Internal attack refers to internal participants/server S or participants and S colluding to derive the sensing data and weight of other nodes.

Theorem 2 (under honest but curious setting). *During the execution of the STDDA algorithm, the sensing data and weight of participant can resist internal attacks.*

Proof. Internal attacks that derive the sensing data and weight of participants can be attributed to three types: participants, servers, and participants and servers colluding. (1) When an internal attacker is a participant: Because the transmitted sensing data and weight are encrypted by the target node in the cluster which uses the secret random number $R_i = (g_2^{u_{i+1}} / g_2^{u_{i-1}})^{u_i} \pmod{p^2}$, the attacker must obtain the secret random number R_i to obtain the plaintext of the target node. But the integer u_i is only known by the target node. Therefore, the attacker cannot obtain the plaintext of sensing data and weight. (2) When the internal attacker is a server: the attacker can only get the aggregated plaintext data but cannot derive the plaintext data of a single node. (3) A collusion attack between participants and the server: When the server colludes with $(k - 1)$ nodes in the cluster, the data of the target node will be leaked. Assuming that the probability of malicious nodes in the cluster is p , the probability of the target node leaking is related to the number of member nodes in the cluster, and its specific probability is $p^{k-1} \times (1 - p) \times k$. So, when k is large, its probability is negligible. In summary, the participant's sensing data and weight can prevent internal attacks. \square

5. Experiment and Performance Evaluation

5.1. Performance Evaluation. The performance evaluation of the truth discovery algorithm with privacy protection capability mainly includes the following: (1) whether the correct truth discovery results can be obtained; (2) whether the privacy of users can be guaranteed; (3) whether to rely on a trusted third party; (4) whether the user and the server (user) are required to not collude with each other; (5) whether to consider the dynamics of users in mobile crowd sensing. From Table 1, we can see that STDDA has advantages in the above five aspects.

5.2. Experiment Verification. In order to more realistically estimate the performance of STDDA, we design and develop a privacy protection truth discovery APP and background processing system. The front-end experimental environment is a smartphone (Huawei, iPhone, etc.), the operating system is Android 9.0 and above, the running memory is 4 GB and

TABLE 1: Performance comparison with existing approaches.

Properties	CRH [12]	PPTD [27]	EPTD [29]	STDDA
Correct truth discovery results	Yes	Yes	Yes	Yes
Ensured privacy	No	Yes	Yes	Yes
Trusted third party	No	No	Yes	No
Anticollusion attack	No	No	Yes	Yes
Dynamic join and quit of participants	No	No	No	Yes

above, and the back-end environment is operating system Win7, CPU Intel Core i5, 16 GB RAM. In our experiment, 100 mobile smart devices are used to target objects (latitude, longitude, etc.) in 10 buildings (such as schools, supermarkets, and hotels) for data collection. The truth discovery processing result of the object in the building and the corresponding map location are displayed as red dots in Figure 4, where the red mark indicates the building collection result and the corresponding display location.

In addition, we also analyze the accuracy, convergence, computational overhead, and communication overhead of the algorithm. In order to more truly reflect the experimental results, each experiment below is repeated 10 times, and the experiment shows that the result is the average value of the experiment.

5.2.1. Accuracy. In this experiment, the accuracy of CRH [12], PPTD [27], and STDDA algorithm is measured by the mean of absolute error (MAE) and the root of mean squared error (RMSE). Since PPTD requires sensing data to be calculated in integers, it is necessary to introduce the parameter L to approximate the data by rounding method [27] when computing the MAE and RMSE of PPTD. Therefore we set $L = 106$. Figures 5(a) and 5(b) show the changes in the MAE and RMSE of the corresponding three algorithm longitudes as the number of participants increases. Figures 5(c) and 5(d), respectively, show the changes of MAE and RMSE of the latitude. From Figure 5, we can see that the accuracy of the STDDA is consistent with CRH, because the parameter L is introduced by PPTD, so the accuracy is lower.

5.2.2. Convergence. By setting 5 different initial estimated ground truth values x_j^* to verify the convergence of the STDDA algorithm, it can be seen from Figure 6 that, under different estimated ground truth, basically two iterations can achieve the convergence requirements and higher efficiency.

5.2.3. Computational Overhead. Under the same hardware environment, by experimenting with a different number of objects, we obtain the communication overhead (run time) of the weight update and truth evaluation. We will explain the running time of the weight update, truth evaluation, and the entire process. As the number of objects increases, the running time of STDDA's weight update and truth evaluation is shown in Figure 7. At the same time, Figure 8 shows the running time of STDDA, PPTD, and EPTD for different numbers of users. In the secure weight update



FIGURE 4: The map display of the building.

procedure, the participant P_i needs to encrypt and decrypt the data twice, respectively, in PPTD. In EPTD, the user needs to perform the Diffie-Hellman key exchange protocol to obtain the public key, and the user needs to perform two encryption operations and one decryption operation, but in STDDA, P_i only needs to encrypt D_i , which is the sum of object distance function, to get $E(D_i)$, while CH only performs simple multiplication. In the secure truth evaluation procedure, the P_i needs to perform two encryption operations and one data decryption in PPTD. In EPTD, the user needs to negotiate a public key, and the user needs to perform two encryption operations and one decryption operation, which is the same as the weight update stage, but in STDDA, the participant P_i needs to perform two encryption operations on W_i and WO_i , and CH only performs multiplication operations. In summary, STDDA has the shortest running time, EPTD is the second, and PPTD is

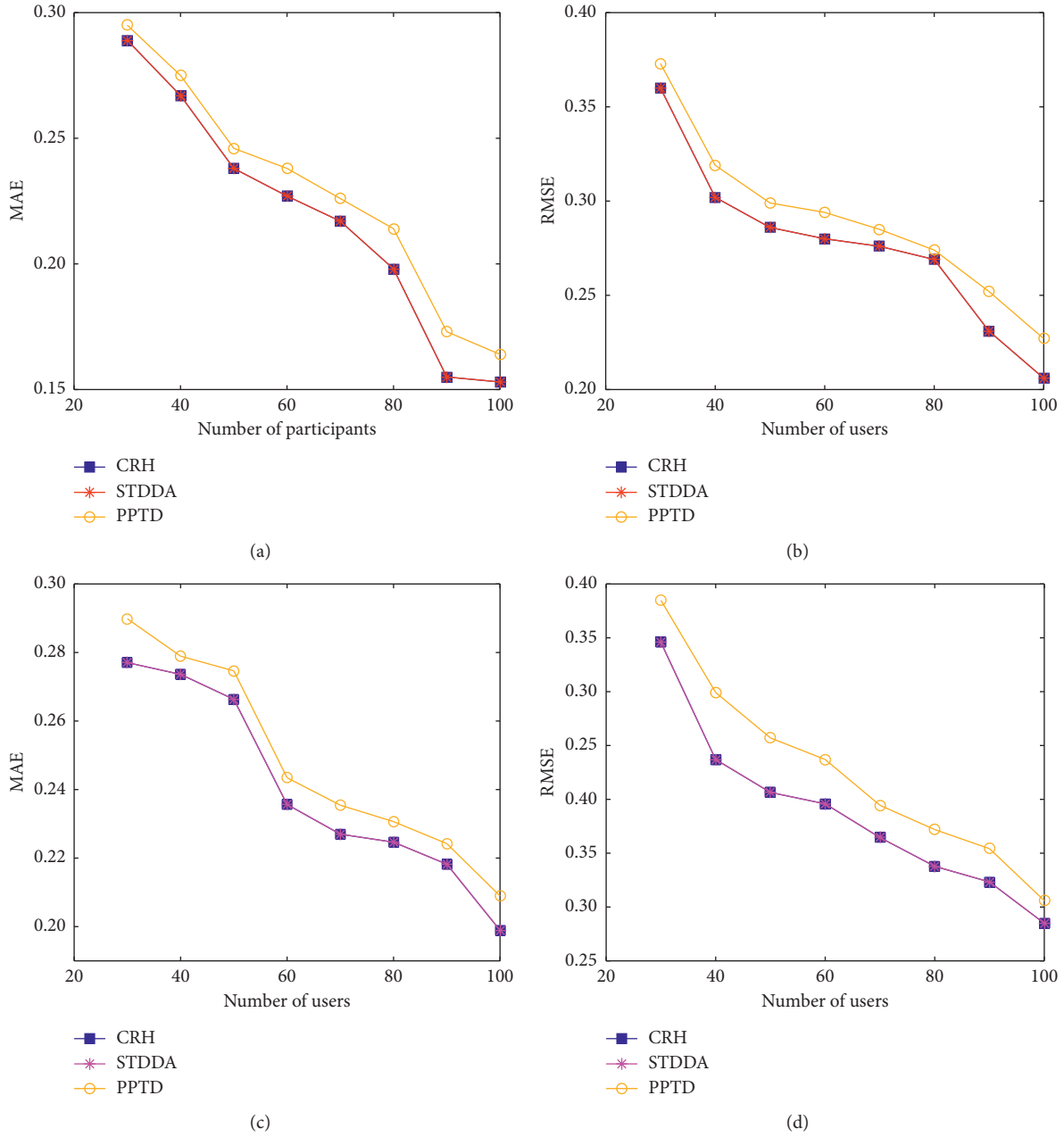


FIGURE 5: MAE and RMSE of object under different number of participants. (a) MAE. (b) RMSE. (c) MAE. (d) RMSE.

the longest. Figure 9 shows the comparison of total running time of the three algorithms.

5.2.4. Communication Overhead. The truth discovery algorithm mainly includes two procedures: weight update and truth evaluation. In this section, the communication overhead of the algorithm is obtained by analyzing the resource consumption of the participant nodes and the traffic between participant nodes and the CH in the two phases. Our article assumes that the length of all sent ciphertext data is u bits, and the number of iterations is a . (1) Secure weight update procedure: Participant node calculates the sum of object distance function D_i based on the sensing data and the

initial ground truth provided by the server S , while encrypting and transmitting it to CH. So the time and space complexity are $O(1)$ and $O(|u|)$ ($|u|$ represents the length of the ciphertext) of a single participant node. And the total time and space complexity of this phase are $O(n)$ and $O(|u|)$. When each node P_i sends $E(D_i)$ to CH, the communication overhead is u . CH receives the ciphertext of all participants in the cluster, while fusing and sending it to the server S . And its traffic is $(k-1) \times u + u$ (each cluster has $(k-1)$ nodes and 1 cluster head node on average). (2) Secure truth evaluation procedure: The participant node encrypts the weight and the product of the weight and the sensing data and transmits it to CH. The time complexity of a single node is $O(1)$ and the space complexity is $O(|u|)$, so the time and space complexity

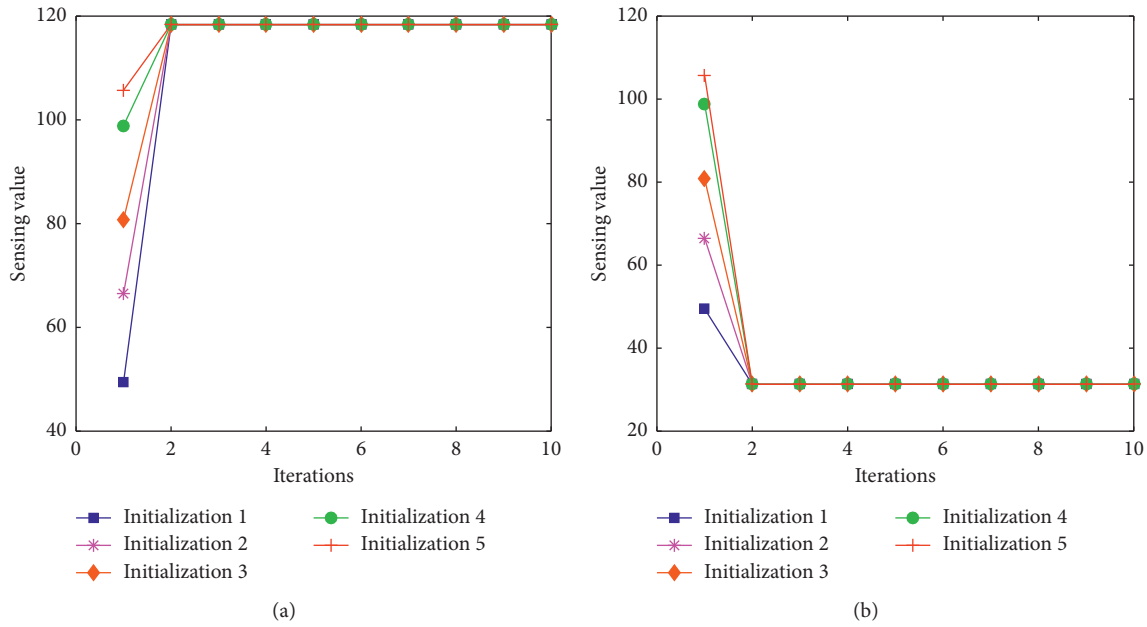


FIGURE 6: Comparison of convergence. (a) Longitude. (b) Latitude.

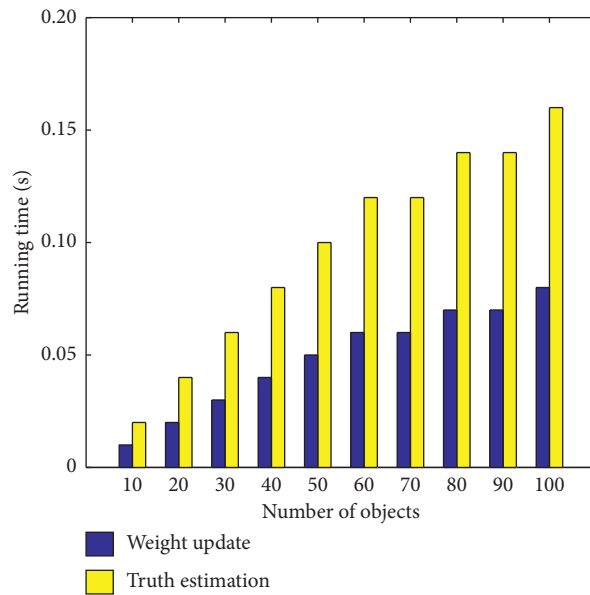


FIGURE 7: Running time of weight update and truth estimation under different number of objects.

of the STDDA algorithm in the secure truth evaluation phase are $O(n)$ and $O(n|u|)$, respectively. Each node P_i sends $E(W_i)$ and $E(WO_i)$ to CH, whose traffic is $2u$. CH receives $E(W_i)$ and $E(WO_i)$ from all participants in the cluster and fuses and uploads them to S, whose traffic is $(k-1) \times 2u + 2u$. Since the algorithm iterates a times on average, the algorithm traffic is shown in Table 2.

In PPTD, a single user needs to send ciphertext data three times and receive ciphertext data once. $(t' - 1)$ users receive three times ciphertext and send three times plaintext data to the server. Therefore, the communication overhead of PPTD is $4 \times n \times u \times a + 6 \times u \times a \times (t' - 1)$ in the whole process, where t' represents the number of users at the time of decryption. In EPTD, a single user needs to use Shamir's

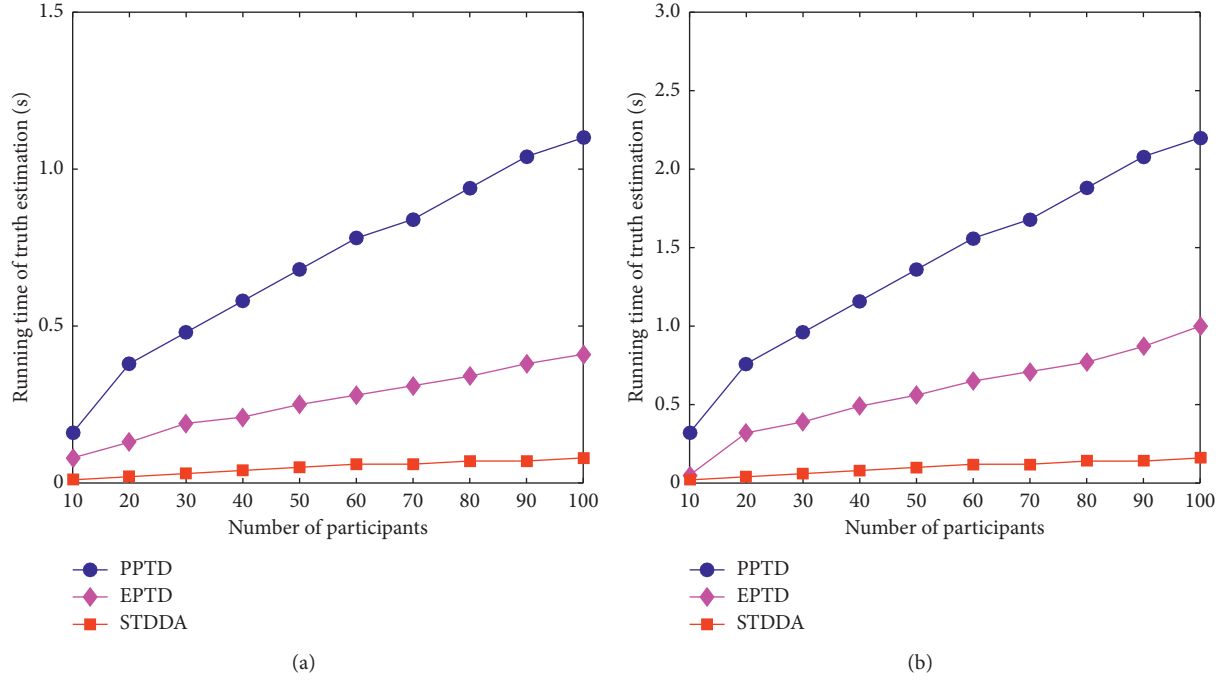


FIGURE 8: Comparison of running time. (a) The running time of weight update under different number of objects. (b) The running time of truth estimation under different number of objects.

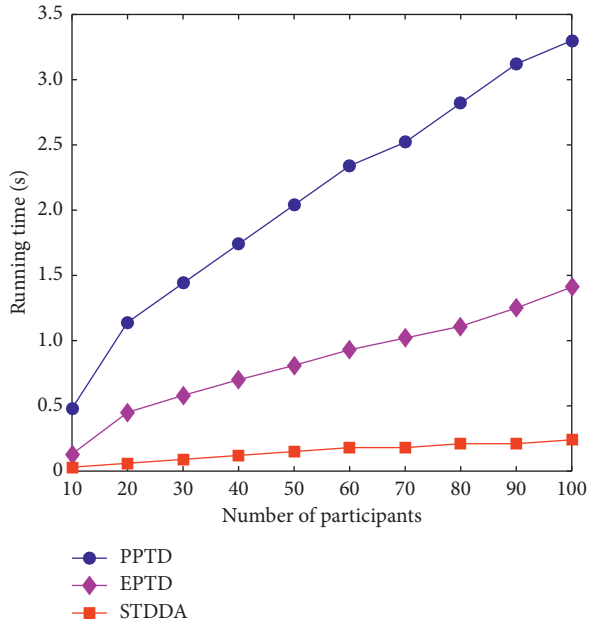


FIGURE 9: Comparison of total running time.

TABLE 2: Traffic overhead.

	Each P_i (b)	CH(b)	Total traffic
Secure weight update	$u \times a$	$k \times u \times a$	$(n-t) \times u \times a + t \times u \times a$
Secure truth estimation	$2u \times a$	$k \times 2u \times a$	$(n-t) \times 2u \times a + t \times 2u \times a$
Entire process	$3u \times a$	$k \times 3u \times a$	$3 \times n \times u \times a$

TABLE 3: Comparison of communication overhead.

Methods	Communication overhead
STDDA	$3 \times n \times u \times a$
PPTD [27]	$4 \times n \times u \times a + 6 \times u \times a \times (t' - 1)$
EPTD [29]	$4 \times n \times u \times a \times t'' + 7 \times u \times a \times t''$

(k, n) threshold key sharing protocol to distribute the private key four times to t'' users. A single user sends four ciphertexts to the server. At the same time, t'' users also need to send three times decryption key to the server again. Therefore, the communication overhead of EPTD in the whole process is $4 \times n \times u \times a \times t'' + 7 \times u \times a \times t''$, where t'' represents the number of users when uploading data or decrypting. Table 3 shows three comparisons of the total communication overhead, where $t' > 0$ and $t'' > 0$.

6. Conclusion

The STDDA algorithm proposed in this paper is used to solve the problem of truth discovery for privacy protection data fusion in MCS. Participants are divided into several clusters based on the number and position of participants, and the cluster head node is randomly assigned in each cluster. Then participants inside compute the corresponding secret random number according to the common parameters shared by the predecessor and successor nodes, ensuring the privacy of the data by adding secret random number to the sensing data. At the same time, the cluster head node uses the secure sum protocol to fuse the sensing data in the cluster, while encrypting and uploading it to the server, which decrypts and aggregates all cluster data to

obtain the sum of the sensing data of all participants in the entire system, and finally we iterate weight update and truth evaluation until convergence. So the server cannot obtain the sensing data and weight of a single participant, which further ensures the privacy of participants' sensing data and weight. In addition, using the truth discovery technology, the STDDA algorithm provides corresponding processing mechanisms for the dynamic join and invalid exit of participant nodes, enhancing the system robustness. Theoretical analysis shows that the STDDA algorithm can both defend against external attacks and resist internal attacks. A large number of experimental results prove that the STDDA algorithm has the characteristics of high security, high accuracy, and low communication. Besides, STDDA algorithm has great advantages over existing methods.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Science Foundation of China (61972439, 61972438, and 61871412), Key Research and Development Projects in Anhui Province (202004a05020002), 2019 Key Project of Natural Science Research in Colleges and Universities of Anhui Provincial Department of Education (KJ2019A1164), the Anhui Normal University PhD Startup Fund (2018XJJ66), and the Anhui Normal University Innovation Fund (2018XJJ114).

References

- [1] A. El, F. El, F. Ennaji, and M. Sadgal, "A mobile crowd sensing framework for suspect investigation: an objectivity analysis and de-identification approach," *Computer Science and Information Systems*, vol. 17, no. 1, pp. 253–269, 2020.
- [2] J. Nan, X. Dong, Z. Jie et al., "Toward optimal participant decisions with voting-based incentive model for crowd sensing," *Journal of Information Science*, vol. 512, pp. 1–17, 2020.
- [3] D. Wu, J. Liu, and Z. Yang, "Bilateral satisfaction aware participant selection with MEC for mobile crowd sensing," *IEEE Access*, vol. 8, Article ID 48110, 2020.
- [4] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, "A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2347–2360, 2020.
- [5] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8549–8560, 2018.
- [6] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels, "Noisetube: measuring and mapping noise pollution with mobile phones," in *Proceedings of the Information Technologies in Environmental Engineering*, pp. 215–228, Springer, Berlin, Germany, April 2009.
- [7] S. Vigneshwaran, K. Amit, N. Vikrant et al., "ConferenceSense: a case study of sensing public gatherings using participatory smartphones," in *Proceedings of the International Workshop on Pervasive Urban Crowdsensing Architecture and Applications*, Zürich, Switzerland, September 2013.
- [8] X. Xiaoxin Yin, J. Jiawei Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, 2008.
- [9] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, "Truth finding on the deep web," *Proceedings of the VLDB Endowment*, vol. 6, no. 2, pp. 97–108, 2012.
- [10] H. Jin, L. Su, and K. Nahrstedt, "Theseus: incentivizing truth discovery in mobile crowd sensing systems," in *Proceedings of the Mobihoc*, pp. 1–10, Chennai, India, July 2017.
- [11] Z. Daniel Yue, B. Jose, Z. Yang et al., "Towards reliable missing truth discovery in online social media sensing applications," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 143–150, Vancouver Canada, August 2018.
- [12] L. Qi, L. Yaliang, G. Jing et al., "Resolving Conflicts in Heterogeneous data by truth discovery and source reliability estimation," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pp. 1187–1198, Snowbird, UT, USA, June 2014.
- [13] L. Qi, L. Yaliang, G. Jing et al., "A confidence-aware approach for truth discovery on long-tail data," *Proceedings of the VLDB Endowment*, vol. 8, no. 4, pp. 425–436, 2014.
- [14] Y. Yi, B. Quan, and L. Qing, "A probabilistic model for truth discovery with object correlations," *Knowledge-Based Systems*, vol. 165, pp. 360–373, 2019.
- [15] J. Yang, J. Wang, and W. P. Tay, "Using social network information in community-based Bayesian truth discovery," *IEEE Transactions on Signal and Information Processing Over Networks*, vol. 5, no. 3, pp. 525–537, 2019.
- [16] H. Xiao, J. Gao, Q. Li et al., "Towards confidence interval estimation in truth discovery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 575–588, 2019.
- [17] Z. Daniel, W. Dong, N. Vance et al., "On scalable and robust truth discovery in big data social media sensing applications," *IEEE Transactions on Big Data*, vol. 5, no. 2, pp. 195–208, 2019.
- [18] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
- [19] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
- [20] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.
- [21] C. Chi-Yin, F. M. Mohamed, H. Tian et al., "A privacy-preserving location monitoring system for wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 94–107, 2010.
- [22] H. Kargupta, S. Datta, W. Qi et al., "On the privacy preserving properties of random data perturbation techniques," in *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*, Melbourne, FL, USA, November 2003.
- [23] Y. Shen, T. Zhang, Y. Wang, H. Wang, and X. Jiang, "Microthings: a generic iot architecture for flexible data

- aggregation and scalable service cooperation,” *IEEE Communications Magazine*, vol. 55, no. 9, pp. 86–93, 2017.
- [24] I. Damgård and M. Jurik, “A generalisation, a simplification and some applications of Paillier’s probabilistic public-key system,” in *Public Key Cryptography* Springer, Berlin, Germany, 2001.
- [25] J. Liu, Y. Tian, Y. Zhou, Y. Xiao, and N. Ansari, “Privacy preserving distributed data mining based on secure multi-party computation,” *Computer Communications*, vol. 153, pp. 208–216, 2020.
- [26] C. Miao, W. Jiang, L. Su et al., “Cloud-enabled privacy-preserving truth discovery in crowd sensing systems,” in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pp. 183–196, Seoul Republic of Korea, November 2015.
- [27] C. Miao, W. Jiang, L. Su et al., “Privacy-preserving truth discovery in crowd sensing systems,” *ACM Transactions on Sensor Networks*, vol. 15, no. 1, pp. 9–32, 2019.
- [28] Y. Zheng, H. Duan, and C. Wang, “Learning the truth privately and confidently: encrypted confidence-aware truth discovery in mobile crowdsensing,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2475–2489, 2018.
- [29] G. Xu, H. Li, S. Liu, M. Wen, and R. Lu, “Efficient and privacy-preserving truth discovery in mobile crowd sensing systems,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3854–3865, 2019.
- [30] T. Jung, X.-Y. Li, and M. Wan, “Collusion-tolerable privacy-preserving sum and product calculation without secure channel,” *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 45–57, 2015.

Research Article

A Lightweight Three-Factor Authentication and Key Agreement Scheme for Multigateway WSNs in IoT

Lingyan Xue,¹ Qinglong Huang ,¹ Shuaiqing Zhang ,¹ Haiping Huang ,^{1,2}
and Wenming Wang ^{1,3}

¹School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, Jiangsu, China

²Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, Jiangsu, China

³School of Computer and Information, Anqing Normal University, Anqing 246011, Anhui, China

Correspondence should be addressed to Haiping Huang; hhp@njupt.edu.cn

Received 10 April 2021; Revised 23 May 2021; Accepted 5 June 2021; Published 22 June 2021

Academic Editor: Jinbo Xiong

Copyright © 2021 Lingyan Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet of Things (IoT) has built an information bridge between people and the objective world, wherein wireless sensor networks (WSNs) are an important driving force. For applications based on WSN, such as environment monitoring, smart healthcare, user legitimacy authentication, and data security, are always worth exploring. In recent years, many multifactor user authentication schemes for WSNs have been proposed using smart cards, passwords, as well as biometric features. Unfortunately, these schemes are revealed to various vulnerabilities (e.g., password guessing attack, impersonation attack, and replay attack) due to nonuniform security evaluation criteria. Wang et al. put forward 12 pieces of widely accepted evaluation criteria by investigating quantities of relevant literature. In this paper, we first propose a lightweight multifactor authentication protocol for multigateway WSNs using hash functions and XOR operations. Further, BAN logic and BPR model are employed to formally prove the correctness and security of the proposed scheme, and the informal analysis with Wang et al.'s criteria also indicates that it can resist well-known attacks. Finally, performance analysis of the compared schemes is given, and the evaluation results show that only the proposed scheme can satisfy all 12 evaluation criteria and keep efficient among these schemes.

1. Introduction

As the third revolution of the information technology industry, Internet of Things (IoT) has been developing for over 20 years. During this period, more and more physical objects embedded with sensors and terminal devices are constantly connected to IoT to exchange information. For an instance, in wireless sensor networks (WSNs), tens of thousands of different sensors are deployed everywhere (e.g., architectures, bridges, and intelligent terminals). These devices collect the real-time data from surrounding environment or target objects and, at fixed periods, forward the collected data directly to nearby gateway nodes for further analysis. Then, application systems access the data through the network, to further provide various personalized services. In heterogeneous WSNs, any insecure terminal nodes possibly threaten the whole network's security as the flexible access

mode; potential vulnerabilities continually come forth due to the complexity of heterogeneous networks [1]. Thus, it is necessary to design an authentication protocol to ensure that only legitimate users have access to the network [2]. Generally, as far as sensor nodes are resource-constrained in some aspects such as low energy, insufficient computing capabilities, and lack of memory space, many expensive cryptographic primitives are not suitable. As a whole, the designed proposal for WSNs should be balanced well in both security and efficiency.

When it was 1981, Lamport [3] proposed the password-based authentication scheme, and in 1991, Chang and Wu [4] pioneered the smart card-based authentication scheme. Henceforth, achievements on single-factor identity authentication protocols for WSNs emerge in an endless stream. Until 2009, combining the smart card with password, Das [5] put forward a pioneering work on multifactor

authentication protocols for WSNs. However, it was revealed to many weaknesses, i.e., destitution of mutual authentication, and vulnerabilities to password guessing attack, sensor node capture attack, and denial-of-service attack (DoS) [6–8]. Later, many multifactor authentication schemes that asserted high security and efficiency were proposed yet they were prone to various attacks [9, 10]. Xue et al. [11] presented a temporal-credential-based mutual authentication and key agreement scheme for WSNs. Soon afterwards, loopholes were pointed out in their scheme, i.e., vulnerabilities to offline password guessing attack, user tracing, impersonation attack, and stolen-verifier attack, as well as the lack of user anonymity [12–14]. In recent years, biological information of human bodies, such as fingerprint and iris, has been excavated for authentication. With its unforgeability, uniqueness, and stability, biometric authentication technology is inherently convenient, reliable, and promising [15]. Yuan [16] took human’s fingerprint as a third factor to achieve user authentication for WSNs, which was lightweight. Nevertheless, their scheme was pointed out that it did not withstand offline password guessing attack, privileged insider attack, and gateway impersonation attack. Then, Li et al. [17] introduced a three-factor authentication scheme for WSNs using biometric features. Subsequently, their scheme was illustrated that it could not resist to stolen smart card attack and support forward secrecy [18]. Additionally, in the practical applications of WSNs, multiple gateways are usually deployed to jointly manage multiple areas. As such, the user can access any sensor node for the real-time data in any area. Research on multigateway-based authentication protocols is also a deserving discussion. Amin et al. [19] proposed a two-factor multiple gateways’ authentication protocol using hash functions. Later, Wu et al. [20] believed that their scheme did not realize mutual authentication and resist impersonation attack; then, they put forward a new scheme. And, Srinivas et al. [21] also found many flaws in [19], i.e., stolen smart card attack and sensor node spoofing attack, and then, they presented a three-factor authentication scheme using hash functions. However, their scheme was also revealed to vulnerability to sensor node capture attack and nonsupport for user anonymity. In 2019, Guo et al. [22] found that the scheme designed by Wu et al. [20] could not resist to stolen smart card attack and session key reveal attack. In order to address these drawbacks, Guo et al. [22] presented a new scheme based on biometric features. Recently, Vinoth et al. [23] proposed a secure multifactor authentication key agreement scheme for industrial IoT, which was insecure as they claimed. It actually could not deal with such attacks such as sensor node capture attack, DoS attack, and replay attack.

As all mentioned above, these schemes are exposed to various vulnerabilities constantly, which in fact are trapped into a “break-propose-break” cycle. Security properties of one scheme is determined by an evaluation standard system, thereby researchers always find new flaws under different systems. In 2018, on the basis of the previous research studies, Wang and Wang [24] summarized and put forward security criteria for two-factor authentication protocols, which are recognized by the industry at present. In these

criteria, 12 pieces of independent and fundamental rules are contained that multifactor authentication protocols shall satisfy. Specific content of the criteria can be referred to [24]; we call it “12-Criteria” here for the sake of convenience.

In terms of 12-Criteria, most existing multifactor authentication protocols cannot satisfy all. This paper will put forward a new lightweight three-factor authentication and key agreement scheme for multigateway WSNs, and main contributions are summed up as below:

- (1) We first reanalyse Guo et al.’s protocol [22]. And, in accordance with 12-Criteria, we further point out some vulnerabilities and drawbacks that still exist in their scheme, including no repairability, improper treatment of biological factors, offline password guessing attack, and lack of forward secrecy.
- (2) In the light of the 12-Criteria, we put forward a new lightweight three-factor authentication and key agreement scheme for the multigateway environment. In our scheme, biometric features, as an important factor, are extracted and validated by fuzzy extractor [25]. And, honey_list [24] is introduced to assist the effective smart card logout.
- (3) Formal and informal security analyses are given amply to prove the correctness and security of the proposed scheme, and comparisons with similar research studies show that this new scheme achieves a superior balance between security and efficiency.

The reminder of this paper is organized as follows. The relevant background is introduced in Section 2. In Section 3, discussions of some security flaws in Guo et al.’s work [22] are given. The proposed protocol and the corresponding security analysis are presented in Sections 4 and 5, respectively. The performance of the proposed protocol is evaluated in Section 6, and finally, the whole paper is concluded in Section 7.

2. Preliminaries

This section briefly introduces some necessary notations, system model, and adversary model, as well as preknowledge about formal proofs.

2.1. Notations. The related notations used in this paper are described in Table 1.

2.2. System Model. A multigateway system model is illustrated in Figure 1, wherein three roles, i.e., users, gateway nodes (GWNs), and sensor nodes, are included. Considering the distance measure, the relatively close node is referred to the home gateway node (HGWN), while the opposite is the foreign gateway node (FGWN). The communication processes are summarized as follows.

While a legitimate user attempts to communicate with the sensor node, first he needs to login successfully and send a message to inform HGWN. After the reception of the message, HGWN first checks its database with the key

TABLE 1: Notations.

Notation	Description
$U_i, ID_i, PW_i,$ and BIO_i	The identity ID_i , password PW_i , and biological factor BIO_i of the user U_i
S_j and SID_j	The identity SID_j of the sensor node S_j
HGWN, ID_{hg} , and x_{hg}	The identity ID_{hg} and the private key x_{hg} of home gateway node HGWN
FGWN, ID_{fg} , and x_{fg}	The identity ID_{fg} and the private key x_{fg} of home gateway node FGWN
SA	The system administrator
SC	The smart card
ΔT	The maximum permitted transmission delay
$SK_u, SK_s, SK_{fg},$ and SK_{hg}	The negotiated session key
$h(\cdot)$ and $H(\cdot)$	The hash function
$Gen(\cdot)$ and $Rep(\cdot)$	The biometric feature extraction function and verification function
\oplus	The XOR operator
\parallel	The concatenation operator
$A \rightarrow B$	A sends messages to B over a public channel
$A \Rightarrow B$	A sends messages to B over a private channel

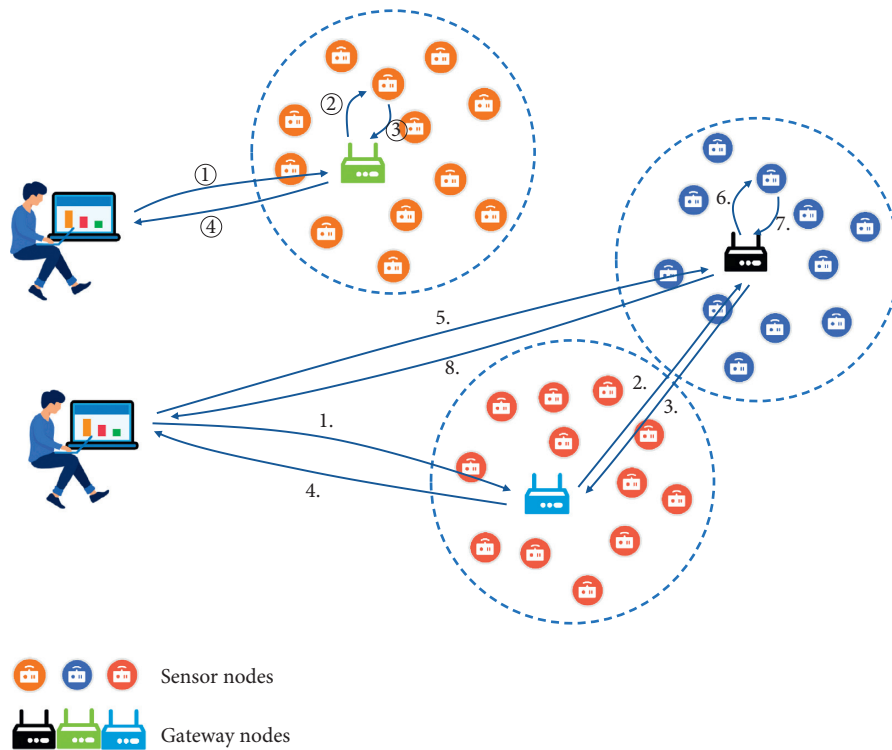


FIGURE 1: System model.

information of the target sensor node as an index. Here, two cases would be taken into an account. Case 1 is presented in steps ①–④, wherein if the target sensor node exists in the database, HGWN authenticates the user and sends a message to the sensor node. Then, the sensor node authenticates HGWN and returns a message. After the complete verification of the returned message, HGWN returns a message to the user. Similarly, once the message is verified correctly by the user, the three parties can derive a common session key for further communication. While Case 2 is shown in steps 1–8, that is, the target sensor node does not exist in the database, HGWN broadcasts the request message to other nodes. When FGWN receives that and finds that the wanted sensor node exists in its database, it sends a message to

HGWN. Then, HGWN returns a message to the user. After a complete authentication process, the user, FGWN, and the sensor node can negotiate the very session key.

2.3. *Notations and Formulas of Ban Logic.* The Burrows-Abadi-Needham logic [26], BAN logic for short, plays a positive and effective role when proving that one scheme can support authentication and key agreement among communicating participants. Formally, it needs three steps including idealization of interaction messages in the protocol, initial assumptions according to specific situations, and achievements of expected goals by inference rules. We first present the basic notations of BAN logic in Table 2.

TABLE 2: Notations of BAN logic.

Notations	Descriptions
$P \equiv X$	P believes X is true
$P \triangleleft X$	P sees X and is capable of reading and repeating it
$P \sim X$	P once said X ; at some time, P has sent the message containing X
$P \Longrightarrow X$	P has control or jurisdiction over X
$\#(X)$	X is fresh which means it was never sent before the current execution of the protocol
$P \xleftrightarrow{K} Q$	Both P and Q can use the shared key K to communicate with each other, and K is an intact key
$P \xleftrightarrow{X} Q$	X is a secret only known to P and Q and possibly to principals trusted by them
$\langle X \rangle_Y$	X combined with Y

The basic formulas of BAN logic are described as follows.

- (i) (R1) Message-meaning rule: if P concludes that the secret K or Y is shared with Q and sees $\langle X \rangle_Y$ or $(X)_K$, then P believes Q once said X :

$$\frac{P \equiv P \xleftrightarrow{Y} Q, P \triangleleft \langle X \rangle_Y}{P \equiv Q | \sim X} \quad (1)$$

- (ii) (R2) Freshness rule: if P believes X is fresh, then P believes (X, Y) is also fresh:

$$\frac{P \equiv \#(X)}{P \equiv \#(X, Y)} \quad (2)$$

- (iii) (R3) Belief rule: if P believes X and Y , then P believes the combination of X and Y :

$$\frac{P \equiv X, P \equiv Y}{P \equiv (X, Y)} \quad (3)$$

- (iv) (R4) Nonce-verification rule: if P believes that X is fresh and Q once said X , then P believes that Q believes X :

$$\frac{P \equiv \#(X), P \equiv Q | \sim X}{P \equiv Q | \equiv X} \quad (4)$$

- (v) (R5) Jurisdiction rule: if P believes Q has jurisdiction over X and Q believes X , then P believes X :

$$\frac{P \equiv Q | \Longrightarrow X, P \equiv Q | \equiv X}{P \equiv X} \quad (5)$$

- (vi) (R6) Seeing rule: if P once received a formula and knew the associated key, then P once saw the components of the formula:

$$\frac{P \triangleleft (X, Y)}{P \triangleleft X}, \quad (6)$$

$$\frac{P \triangleleft \langle X \rangle_Y}{P \triangleleft X}$$

- (vii) (R7) Session key rule: if P believes X is fresh and Q believes X , then P believes he shares the key K with Q :

$$\frac{P \equiv \#(X), P \equiv Q | \equiv X}{P \equiv P \xleftrightarrow{K} Q} \quad (7)$$

2.4. Adversary Model. Combing with the 12-Criteria, we list pieces of widely accepted valid assumptions to show the capabilities of an adversary \mathcal{A} , accordingly to analyse the security of the authentication and key agreement protocols.

- (i) When entities in WSN communicate with each other over an insecure wireless channel, \mathcal{A} can eavesdrop and intercept all messages transmitted over a public channel and is capable of tempering with and deleting the intercepted messages. In addition, \mathcal{A} can participate in running the protocol as a legitimate entity.

- (ii) In reality, users' devices and sensors are usually equipped with the hardware to prevent reading and tempering with data illegally [27], but to adhere to the extreme-adversary principle [28], it is reasonable to assume that when the user's device or the sensor is captured by \mathcal{A} , \mathcal{A} has the ability to obtain the data stored in the memory of the captured sensors through side channel attack [24].

- (iii) \mathcal{A} is capable of enumerating the Cartesian products of the user's identity and password. Besides, in the n -factor authentication protocol, \mathcal{A} can obtain $(n - 1)$ factors at most.

- (iv) Only when evaluating the forward secrecy of the protocol, \mathcal{A} can obtain the long-term private key of a gateway node or a sensor node.

2.5. Security Model. To formalize our proposed proposal later, the BPR model [29] can be introduced in this section, i.e., depictions of the random oracle model and definition of authentication and key-exchange (AKE) security.

Participants. The authentication protocol \mathcal{P} involves three communication participants, i.e., the user, HGWN/FGWN, and sensor node. Each participant has many diverse instances which are called oracles. For a specific session, the three entities are instantiated into Π_U^i , $\Pi_{\text{HGWN}}^k / \Pi_{\text{FGWN}}^k$, and Π_S^j , respectively. Here, let Π_I^* denote any instance.

Queries. \mathcal{A} can only interact with honest participants through oracle queries and attempt to collect the returned messages to break the protocol. Thus, the following queries simulate \mathcal{A} 's abilities in practice.

- (i) Execute $(\Pi_U^i, \Pi_{\text{HGWN}}^k, \Pi_S^j)$: it simulates the passive attack, through which \mathcal{A} can obtain all messages

among the three communicators during a normal interaction.

- (ii) Send (Π_I^*, m) : it represents the active attack, which allows \mathcal{A} intercepts, forges the message, further sends it to Π_I^* , and obtains the corresponding response.
- (iii) Reveal (Π_I^*) : it models abuse of the session key. Once Π_I^* accepts the current session and generates a session key SK , it will return SK to \mathcal{A} ; otherwise, return \perp .
- (iv) Corrupt (Π_U^i, a) : it simulates that \mathcal{A} can corrupt any two of the three factors of a legal user U_i , but not at the same time. (1) If $a = 1$, \mathcal{A} can obtain PW_i and all parameters stored in SC ; (2) if $a = 2$, \mathcal{A} can receive BIO_i and all parameters stored in SC ; (3) if $a = 3$, \mathcal{A} can get PW_i and BIO_i .
- (v) Test (Π_I^*) : it represents the semantic security of the session key. Flip a coin b at random; if $b = 1$, it returns \mathcal{A} the session key of Π_I^* ; if $b = 0$, returns a random number equal in length to the session key to \mathcal{A} . If the session key of Π_I^* does not exist, it returns \perp . It is noted that it can only be invoked once at any time for fresh sessions.

Partners. Let sid denote the session identifier; pid is the session identifier of partners. Π_U^i and Π_S^j are partners if and only if (1) they are both authenticated successfully; (2) they both have the same sid ; (3) pid of Π_U^i is Π_S^j , while pid of Π_S^j is Π_U^i .

Freshness. A fresh Π_I^* satisfies that (1) Π_I^* is accepted and owns its session key; (2) \mathcal{A} does not query Reveal $(*)$ to Π_I^* or its partner; (3) since \mathcal{P} runs, \mathcal{A} queries Corrupt $(*)$ to Π_I^* or its partner once at most.

Definition 1. (AKE security) Given $\text{Succ}(\mathcal{A})$ denotes an event, that is, \mathcal{A} makes Test $(*)$ queries to several new accepted instances and can guess the right b' satisfying $b = b'$. Then, the advantage of \mathcal{A} breaking the AKE security of \mathcal{P} can be defined as $\text{Adv}_{\mathcal{P}}^{\text{AKE}}(\mathcal{A}) = |\text{Pr}[\text{Succ}(\mathcal{A})] - 1/2| = |\text{Pr}[b' = b] - 1/2|$. For any adversary capable of breaking \mathcal{P} in probability polynomial time (PPT), $\text{Adv}_{\mathcal{P}}^{\text{AKE}}(\mathcal{A})$ is negligible; then, we say \mathcal{P} achieves AKE security.

3. Cryptanalysis of Guo et al.'s Scheme

The scheme designed by Guo et al. [22] is composed of five parts, including system setup, registration, login, authentication, and password change. Here, we have to leave out the review of their scheme due to space constraints, and readers can refer to [22]. Thus, on the basis of the aforementioned assumptions, security flaws in their scheme are analysed in this section later.

No Sound Repairability. As a usual case, those discarded smart cards are not in the safe keeping of users. If unfortunate, his smart card is captured by an attacker \mathcal{A} . \mathcal{A} possibly launches the offline password guessing attack. Therefore, it is essential to provide a method to cancel the smart card of the user in multifactor authentication protocols.

Improper Treatment of Biometric Factors. As described in this protocol, after the user enters his biometric factor BIO_i , SC calculates $O_i = H(BIO_i)$ which is a key parameter to verify the true identity of the user. In practice, however, a certain error bit always occurs in the extraction of biometric features (e.g., fingerprint and iris) by reading devices, that is, biometric features extracted each time are not always identical. Therefore, O_i calculated by SC may not equal to that obtained during the user's registration phase, which may result in the failed authentication even if the user has input the right password.

Offline Password Guessing Attack. In the login phase, \mathcal{A} is assumed to have the ability to obtain two of the three authentication factors. Given that \mathcal{A} has accessed the user's identity ID_i and biometric factor BIO_i , then he can launch offline password guessing attack as the following process.

\mathcal{A} guesses a possible password PW_i^* , calculates $O_i = H(BIO_i)$, $r_i^* = B_1 \oplus h(O_i \| ID_i \| PW_i^*)$, and $MP_i^* = h(r_i^* \| PW_i^*)$, and checks whether the equation $B_2 = h(MP_i^* \| ID_i \| O_i \| r_i^*)$ holds. \mathcal{A} can repeat these operations until the calculated B_2 equals to $h(MP_i^* \| ID_i \| O_i \| r_i^*)$. Finally, \mathcal{A} can succeed in obtaining the user's correct PW_i .

Lack of Forward Secrecy. Given that the long-term secret key of the GWN is revealed, \mathcal{A} can grab the private key of the sensor and further restore previous session keys.

(i) Case 1:

- (1) \mathcal{A} obtains x_{hg} of HGWN and eavesdrops the message M_1 to gain the identity SID_j of the user-pointed communication object S_j . Then, \mathcal{A} computes $f_j = h(SID_j \| x_{hg})$.
- (2) \mathcal{A} eavesdrops messages M_2 and M_3 and then calculates $Y_j = h(f_j \| T_2)$, $r_{hg} = D_3 \oplus Y_j$, $r_u = D_4 \oplus h(r_{hg} \| f_j \| T_2)$, and $r_s = D_6 \oplus h(r_{hg} \| f_j \| T_3)$. In this way, the session key can be derived by \mathcal{A} as $SK = h(r_s \| r_{hg} \| r_u)$.

(ii) Case 2:

- (1) \mathcal{A} obtains x_{fg} of FGWN and computes $f_j = h(SID_j \| x_{fg})$ after eavesdropping the message M_1 .
- (2) \mathcal{A} eavesdrops messages M_6 and M_7 and then calculates $Y_j = h(f_j \| T_2)$, $r_{fg} = D_{10} \oplus Y_j$, $r_{uu} = D_{11} \oplus h(r_{fg} \| f_j \| T_2)$, and $r_s = D_{13} \oplus h(r_{fg} \| f_j \| T_3)$. Thus, \mathcal{A} can figure out $SK = h(r_s \| r_{fg} \| r_u)$ with ease.

4. The Proposed Scheme

In this section, we present a lightweight three-factor authentication and key agreement scheme for multigateway

WSNs in IoT, which involves users, sensor nodes, HGWNs, and FGWNs. Our scheme includes 6 phases: system initialization, registration, login, authentication and key agreement, password update, and smart card logout.

4.1. System Initialization. SA assigns the identity ID_{hg} and private key x_{hg} to HGWN, similarly, ID_{fg} and x_{fg} to FGWN, and SID_j to the sensor S_j . Then, SA sets up a shared key K_{hf} for the communication between HGWN and FGWN. Beyond that, HGWN and FGWN need to select three random numbers R_h , R_f , and R_{fh} , respectively.

4.2. Registration. As shown in Figure 2, this phase involves two parts, sensor registration and user registration. Both sensor nodes and users need to register their essential information with the closest gateway, namely, HGWN.

4.2.1. Sensor Registration

Step 1: $S_j \Rightarrow$ HGWN: SID_j . S_j sends its identity SID_j to HGWN over a private channel, and HGWN stores SID_j to its database for checking whether or not S_j is registered.

Step 2: HGWN \Rightarrow S_j : $x_j = h(SID_j \| x_{hg}) \oplus R_h$. HGWN calculates $x_j = h(SID_j \| x_{hg}) \oplus R_h$ and sends x_j to S_j via a private channel. After the reception of x_j , S_j saves it secretly.

4.2.2. User Registration

Step 1: $U_i \Rightarrow$ HGWN: $\{ID_i, HPW_i, \beta_i\}$.

U_i inputs his username ID_i , the password PW_i , and his biometric information BIO_i . Next, he chooses a number $r_i \in Z_p^*$ at random and then computes $(\alpha_i, \beta_i) = \text{Gen}(BIO_i)$ and $HPW_i = h(PW_i \| \alpha_i \| r_i)$.

Step 2: HGWN \Rightarrow U_i : $SC \{TID_i, \beta_i, e_i, ID_{hg}\}$.

HGWN selects a pseudoidentity TID_i for U_i and calculates $x_i = h(TID_i \| x_{hg}) \oplus R_h$, $K_i = h(ID_i \| \beta_i)$, and $e_i = HPW_i \oplus K_i \oplus x_i$. Then, HGWN stores $\{ID_i, K_i, \text{honey_list} = 0\}$ into its database and $\{TID_i, \beta_i, e_i, ID_{hg}\}$ to SC, where honey_list records the number of the user logon failures.

Step 3: U_i computes $B_1 = h(\alpha_i \| ID_i \| PW_i) \oplus r_i$ and $B_2 = h(HPW_i \| \alpha_i \| ID_i \| r_i) \bmod n_0$, where $n_0 \in [2^4, 2^8]$. Next, U_i stores $\{B_1, B_2\}$ into his SC.

4.3. Login

Step 1: U_i first inputs ID_i , PW_i , and BIO_i ; then, SC computes $\alpha_i = \text{Rep}(BIO_i, \beta_i)$, $r_i = B_1 \oplus h(\alpha_i \| ID_i \| PW_i)$, and $HPW_i = h(PW_i \| \alpha_i \| r_i)$ and checks whether $B_2 = h(HPW_i \| \alpha_i \| ID_i \| r_i) \bmod n_0$ holds. If so, turn to the next step; otherwise, return a logon failure message and terminate this session.

Step 2: $U_i \rightarrow$ HGWN: $M_1 = \{TID_i, ID_{hg}, SID_j, D_0, D_1, D_2, D_3, T_1\}$. SC chooses a timestamp T_1 and a random number $r_u \in Z_p^*$ and then calculates $K_i = h(ID_i \| \beta_i)$, $x_i = e_i \oplus K_i \oplus HPW_i$, $D_0 = \beta_i \oplus h(x_i \| r_u)$, $D_1 = r_u \oplus x_i$, $D_2 = ID_i \oplus h(r_u \| x_i)$, and $D_3 = h(TID_i \| ID_i \| SID_j \| r_u \| x_i \| K_i \| T_1)$.

4.4. Authentication and Key Agreement. After the reception of U_i 's request to communicate with SID_j , HGWN first confirms whether the specified sensor S_j is located within its communication range. Specifically, if HGWN can query its local database for SID_j , then the authentication can be conducted as described in Case 1 (see Figure 3); otherwise, run as shown in Case 2 (see Figure 4).

(i) Case 1:

Step 1: after receiving M_1 , HGWN records the current timestamp T_2 . If $|T_2 - T_1| \leq \Delta T$ is true, then M_1 is valid; otherwise, this session would be closed up. Next, HGWN computes $x_i = h(TID_i \| x_{hg}) \oplus R_h$, $r_u = D_1 \oplus x_i$, $\beta_i = D_0 \oplus h(x_i \| r_u)$, $ID_i = D_2 \oplus h(r_u \| x_i)$, and $K_i = h(ID_i \| \beta_i)$ and verifies whether the equation $D_3 = h(TID_i \| ID_i \| SID_j \| r_u \| x_i \| K_i \| T_1)$ is true; if so, it turns into the next step; otherwise, it sets $\text{honey_list} = \text{honey_list} + 1$ and returns a logon failure message to U_i . Note that once $\text{honey_list} \geq 10$, U_i 's account would be frozen, and the session is also terminated.

Step 2: HGWN \rightarrow S_j : $M_2 = \{D_4, D_5, D_6, T_2\}$. HGWN selects $r_{hg} \in Z_p^*$ randomly and then computes $x_j = h(SID_j \| x_{hg}) \oplus R_h$, $D_4 = r_{hg} \oplus h(x_j \| T_2)$, $D_5 = r_u \oplus h(r_{hg} \| x_j \| T_2)$, and $D_6 = h(SID_j \| ID_{hg} \| r_u \| r_{hg} \| x_j \| T_2)$.

Step 3: After the reception of M_2 , S_j records the timestamp T_3 and checks the freshness of T_2 . Next, S_j calculates $r_{hg} = D_4 \oplus h(x_j \| T_2)$ and $r_u = D_5 \oplus h(r_{hg} \| x_j \| T_2)$ and checks whether the equation $D_6 = h(SID_j \| ID_{hg} \| r_u \| r_{hg} \| x_j \| T_2)$; if so, it turns to the next step; otherwise, it terminates the current session.

Step 4: $S_j \rightarrow$ HGWN: $M_3 = \{D_7, D_8, T_3\}$. S_j chooses a random number $r_s \in Z_p^*$ and computes $SK_s = h(r_u \| r_{hg} \| r_s \| ID_{hg})$, $D_7 = r_s \oplus h(x_j \| r_{hg} \| T_4)$, and $D_8 = h(ID_{hg} \| SID_j \| x_j \| SK_s \| r_s \| T_3)$.

Step 5: when receiving M_3 from S_j , HGWN records the present timestamp T_4 and verifies the freshness of T_3 . Next, HGWN calculates $r_s = D_7 \oplus h(x_j \| r_{hg} \| T_4)$ and $SK_{hg} = h(r_u \| r_{hg} \| r_s \| ID_{hg})$ and checks whether $D_8 = h(ID_{hg} \| SID_j \| x_j \| SK_s \| r_s \| T_3)$ holds; if so, it turns to the next step; otherwise, it aborts this session.

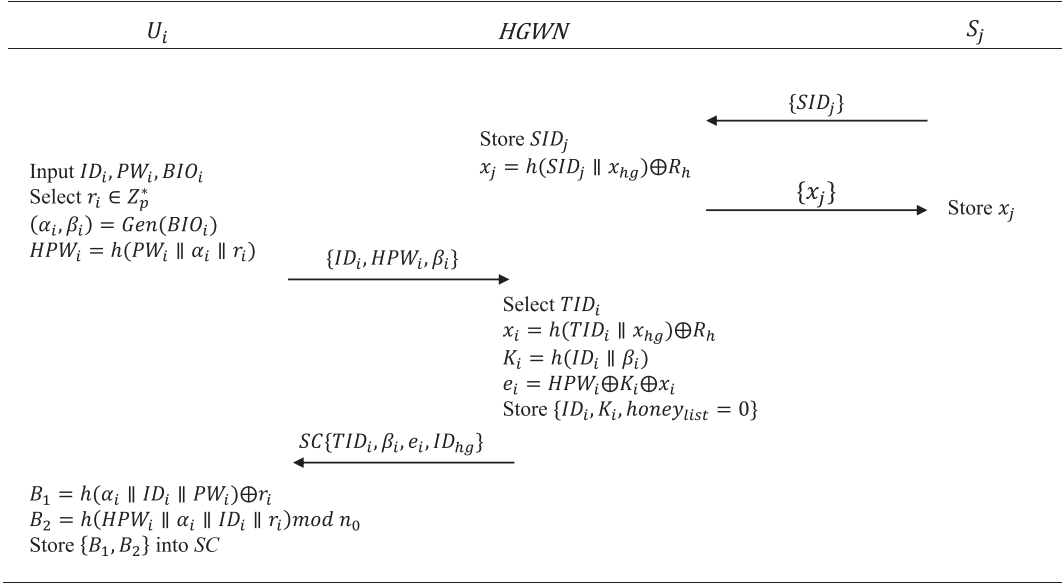


FIGURE 2: Registration phase.

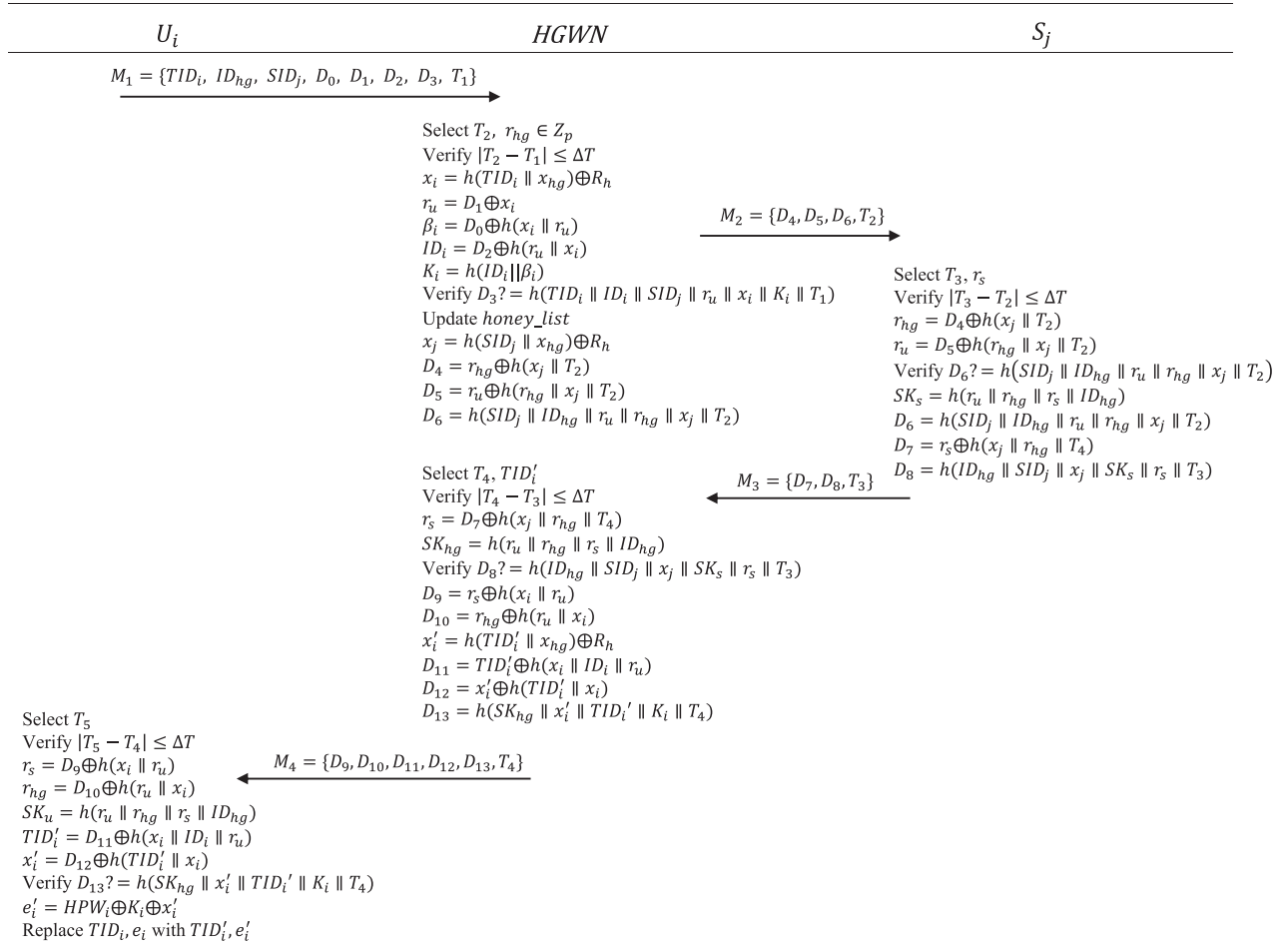


FIGURE 3: Case 1 of the authentication and key agreement phase.

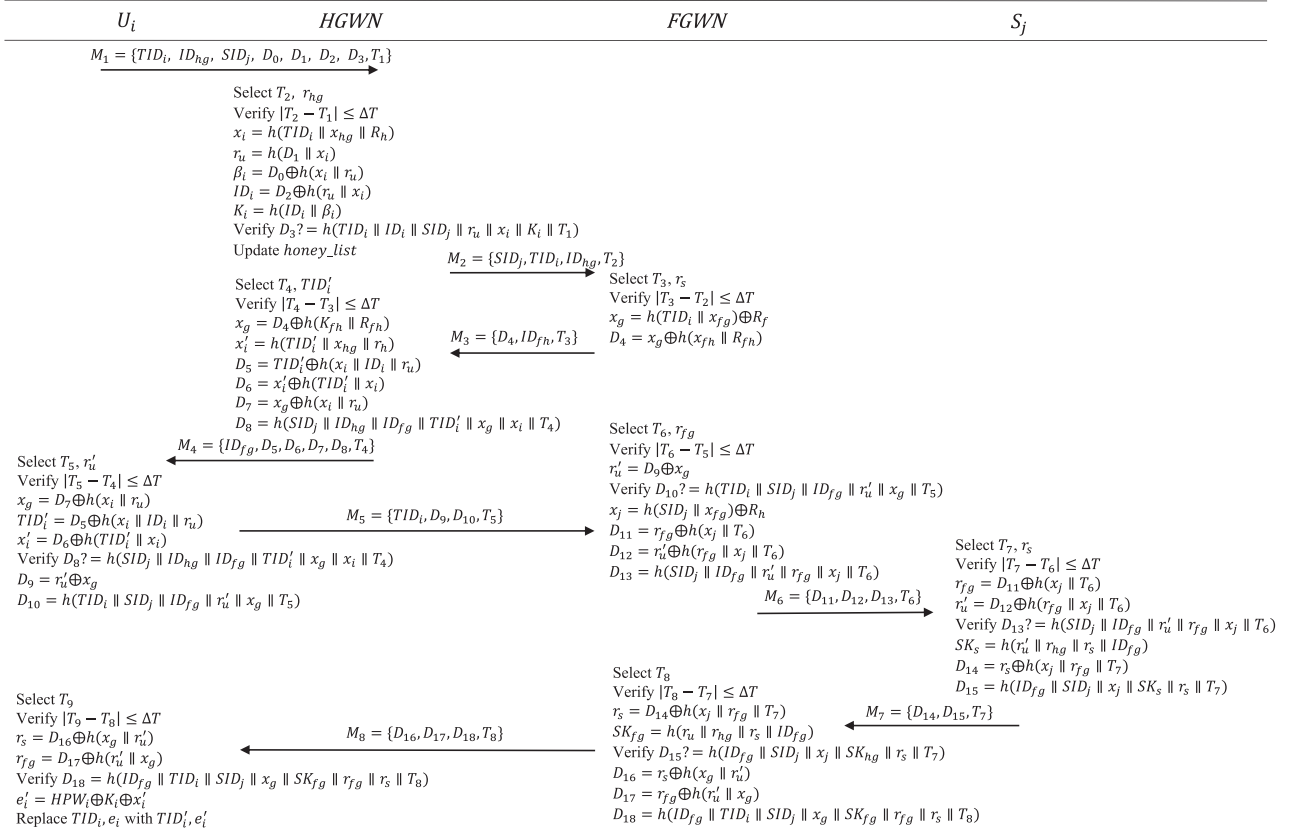


FIGURE 4: Case 2 of the authentication and key agreement phase.

Step 6: $HGWN \rightarrow U_i$: $M_4 = \{D_9, D_{10}, D_{11}, D_{12}, D_{13}, T_4\}$. $HGWN$ chooses a new pseudonym TID'_i for U_i and continues to compute $D_9 = r_s \oplus h(x_i \| r_u)$, $D_{10} = r_{hg} \oplus h(r_u \| x_i)$, $x'_i = h(TID'_i \| x_{hg}) \oplus R_h$, $D_{11} = TID'_i \oplus h(x_i \| ID_i \| r_u)$, $D_{12} = x'_i \oplus h(TID'_i \| x_i)$, and $D_{13} = h(SK_{hg} \| x'_i \| TID'_i \| K_i \| T_4)$.

Step 7: after the reception of M_4 , U_i takes down the current timestamp T_5 and checks the validity of T_4 . Next, U_i computes $r_s = D_9 \oplus h(x_i \| r_u)$, $r_{hg} = D_{10} \oplus h(r_u \| x_i)$, $SK_u = h(r_u \| r_{hg} \| r_s \| ID_{hg})$, $TID'_i = D_{11} \oplus h(x_i \| ID_i \| r_u)$, and $x'_i = D_{12} \oplus h(TID'_i \| x_i)$ and verifies whether the equation $D_{13} = h(SK_{hg} \| x'_i \| TID'_i \| K_i \| T_4)$ matches; if so, then it turns to the next step; otherwise, it discontinues the session.

Step 8: SC calculates $e'_i = HPW_i \oplus K_i \oplus x'_i$ and substitutes $\{TID'_i, e'_i\}$ for $\{TID_i, e_i\}$.

(ii) Case 2:

Step 1: similarly, after the reception of M_1 , $HGWN$ takes down the current timestamp T_2 . If $|T_2 - T_1| \leq \Delta T$, then M_1 is valid; otherwise, the session is discontinued. Next, $HGWN$ computes $x_i = h(TID_i \| x_{hg}) \oplus R_h$, $r_u = D_1 \oplus x_i$, $\beta_i = D_0 \oplus h(x_i \| r_u)$, $ID_i = D_2 \oplus h(r_u \| x_i)$, and $K_i = h(ID_i \| \beta_i)$ and verifies

$D_3 = h(TID_i \| ID_i \| SID_j \| r_u \| x_i \| K_i \| T_1)$. If the equation holds, $HGWN$ runs the next step; otherwise, it sets *honey_list* = *honey_list* + 1, returns a logon failure message to U_i , and aborts the session.

Step 2: $HGWN$ broadcasts $M_2 = \{SID_j, TID_i, ID_{hg}, T_2\}$ to other gateway nodes.

Step 3: $FGWN \rightarrow HGWN$: $M_3 = \{D_4, ID_{fh}, T_3\}$. $FGWN$ finds SID_j in its database, then records the present timestamp T_3 , and computes $x_g = h(TID_i \| x_{fg}) \oplus R_f$ and $D_4 = x_g \oplus h(x_{fh} \| R_{fh})$.

Step 4: $HGWN \rightarrow U_i$: $M_4 = \{ID_{fg}, D_5, D_6, D_7, D_8, T_4\}$. When receiving M_3 from $FGWN$, $HGWN$ takes down the timestamp T_4 and verifies the freshness of T_3 . $HGWN$ selects a new pseudonym TID'_i and calculates $x_g = D_4 \oplus h(K_{fh} \| R_{fh})$, $x'_i = h(TID'_i \| x_{hg} \| r_h)$, $D_5 = TID'_i \oplus h(x_i \| ID_i \| r_u)$, $D_6 = x'_i \oplus h(TID'_i \| x_i)$, $D_7 = x_g \oplus h(x_i \| r_u)$, and $D_8 = h(SID_j \| ID_{hg} \| ID_{fg} \| TID'_i \| x_g \| x_i \| T_4)$.

Step 5: after receiving M_4 , U_i records the time stamp T_5 and checks the validity of T_4 . Then, U_i computes $x_g = D_7 \oplus h(x_i \| r_u)$, $TID'_i = D_5 \oplus h(x_i \| ID_i \| r_u)$, and $x'_i = D_6 \oplus h(TID'_i \| x_i)$ and checks $D_8 = h(SID_j \| ID_{hg} \| ID_{fg} \| TID'_i \| x_g \| x_i \| T_4)$. If the

equation holds, U_i continues the next step; otherwise, it terminates the session.

Step 6: $U_i \rightarrow$ FGWN: $M_5 = \{TID_i, D_9, D_{10}, T_5\}$. U_i selects a random number $r'_u \in Z_p^*$ and computes

$$D_{10} = h(TID_i \| SID_j \| ID_{fg} \| r'_u \| x_g \| T_5).$$

Step 7: after the reception of M_5 , FGWN records T_6 and verifies the freshness of T_5 . Next, FGWN computes $r'_u = D_9 \oplus x_g$ and further checks whether $D_{10} = h(TID_i \| SID_j \| ID_{fg} \| r'_u \| x_g \| T_5)$ matches. If so, FGWN continues the next step; otherwise, it discontinues the session.

Step 8: FGWN \rightarrow S_j : $M_6 = \{D_{11}, D_{12}, D_{13}, T_6\}$. FGWN selects r_{fg} at random and computes $x_j = h(SID_j \| x_{fg}) \oplus R_h$, $D_{11} = r_{fg} \oplus h(x_j \| T_6)$, $D_{12} = r'_u \oplus h(r_{fg} \| x_j \| T_6)$, and $D_{13} = h(SID_j \| ID_{fg} \| r'_u \| r_{fg} \| x_j \| T_6)$.

Step 9: after the reception of M_6 , S_j takes down the timestamp T_7 and verifies the freshness of T_6 . Next, S_j calculates $r_{fg} = D_{11} \oplus h(x_j \| T_6)$ and $r'_u = D_{12} \oplus h(r_{fg} \| x_j \| T_6)$ and checks the equation $D_{13} = h(SID_j \| ID_{fg} \| r'_u \| r_{fg} \| x_j \| T_6)$. If the equation holds, S_j turns to the next step; otherwise, it terminates the session.

Step 10: $S_j \rightarrow$ FGWN: $M_7 = \{D_{14}, D_{15}, T_7\}$. S_j selects r_s at random and computes $SK_s = h(r'_u \| r_{hg} \| r_s \| ID_{fg})$, $D_{14} = r_s \oplus h(x_j \| r_{fg} \| T_7)$, and $D_{15} = h(ID_{fg} \| SID_j \| x_j \| SK_s \| r_s \| T_7)$.

Step 11: once receiving M_7 , FGWN takes down T_8 and verifies the freshness of T_7 . Further, FGWN computes $r_s = D_{14} \oplus h(x_j \| r_{fg} \| T_7)$ and $SK_{fg} = h(r'_u \| r_{fg} \| r_s \| ID_{fg})$ and checks whether the equation $D_{15} = h(ID_{fg} \| SID_j \| x_j \| SK_{fg} \| r_s \| T_7)$ is true; if so, it continues the next step; otherwise, it terminates the session.

Step 12: FGWN \rightarrow U_i : $M_8 = \{D_{16}, D_{17}, D_{18}, T_8\}$. FGWN computes $D_{16} = r_s \oplus h(x_g \| r'_u)$, $D_{17} = r_{fg} \oplus h(r'_u \| x_g)$, and $D_{18} = h(ID_{fg} \| TID_i \| SID_j \| x_g \| SK_{fg} \| r_{fg} \| r_s \| T_8)$.

Step 13: after receiving M_8 , U_i thereupon records the timestamp T_9 and checks the validity of T_8 . Further, U_i computes $r_s = D_{16} \oplus h(x_g \| r'_u)$, $r_{fg} = D_{17} \oplus h(r'_u \| x_g)$, and $SK_u = h(r'_u \| r_{fg} \| r_s \| ID_{fg})$ and checks whether the equation $D_{18} = h(ID_{fg} \| TID_i \| SID_j \| x_g \| SK_u \| r_{fg} \| r_s \| T_8)$ holds; if so, it continues the next step; otherwise, it discontinues the session.

Step 14: SC computes $e'_i = HPW_i \oplus K_i \oplus x'_i$ and replaces $\{TID_i, e_i\}$ with $\{TID'_i, e'_i\}$.

4.5. Password Update

Step 1: U_i first inputs his ID_i , PW_i , and BIO_i . SC computes $\alpha_i = \text{Rep}(BIO_i, \beta_i)$, $r_i = B_1 \oplus h(\alpha_i \| ID_i \| PW_i)$, and $HPW_i = h(PW_i \| \alpha_i \| r_i)$ and checks the equation $B_2 = h(HPW_i \| \alpha_i \| ID_i \| r_i) \bmod n_0$. If the equation holds, the next step can be run; otherwise, a logon failure message would be returned and the logon request also would be terminated.

Step 2: U_i inputs a new password PW'_i , and SC computes $K_i = h(ID_i \| \beta_i)$, $HPW'_i = h(PW'_i \| \alpha_i \| r_i)$, $e'_i = HPW'_i \oplus e_i \oplus HPW_i$, $B'_1 = h(\alpha_i \| ID_i \| PW'_i) \oplus r_i$, and $B'_2 = h(HPW'_i \| \alpha_i \| ID_i \| r_i) \bmod n_0$ and then replaces $\{B_1, B_2, e_i\}$ with $\{B'_1, B'_2, e'_i\}$.

4.6. Smart Card Logout

Step 1: U_i inserts his smart card SC and inputs ID_i , PW_i as well as BIO_i . Further, SC computes $\alpha_i = \text{Rep}(BIO_i, \beta_i)$, $r_i = B_1 \oplus h(\alpha_i \| ID_i \| PW_i)$, and $HPW_i = h(PW_i \| \alpha_i \| r_i)$ and checks whether $B_2 = h(HPW_i \| \alpha_i \| ID_i \| r_i) \bmod n_0$ matches; if so, it turns to Step 2; otherwise, it returns a logon failure message and terminates this session.

Step 2: $U_i \rightarrow$ HGWN: $M_0 = \{TID_i, \beta_i, R_0, T_1\}$. U_i selects the current timestamp T_1 , thereupon computes $K_i = h(ID_i \| \beta_i)$, $x_i = e_i \oplus K_i \oplus HPW_i$, and $R_0 = K_i \oplus (x_i \| T_1)$.

Step 3: after the reception of M_0 , HGWN records the timestamp T_2 . If $|T_2 - T_1| \leq \Delta T$ is true, then M_0 is fresh. Then, HGWN computes $x_i = h(TID_i \| x_{hg}) \oplus R_h$ and $K'_i = R_0 \oplus (x_i \| T_1)$ and continues to check whether $K'_i = K_i = h(ID_i \| \beta_i)$. If the equation holds, it runs the next step; otherwise, it aborts the session.

Step 4: HGWN deletes all local records $\{ID_i, K_i, \text{honey_list}\}$ of U_i .

5. Security Analysis

This section provides a rigorous security analysis for the proposed authentication scheme. On the basis of 12-Criteria, informal analysis first discusses how the proposed scheme resists against some well-known attacks. Second, the well-popular BAN logic is utilized to validate the correctness of the proposed scheme as well as the feasibility for authentication and key negotiation. Finally, the BPR model-based formal security proof demonstrates the security of the proposed scheme well.

5.1. Informal Analysis

Resistance to Insider Attack. In multifactor authentication schemes, the user's password, as a second factor, is of vital for the server/gateway to authenticate the user. The server/gateway in its usual sense is worth

trusting, while it is facing a real possibility that insiders may disclose users' sensitive information. At the registration phase, U_i 's password PW_i is masked by $HPW_i = h(PW_i \| \alpha_i \| r_i)$ to transmit to HGWN. Though \mathcal{A} has the ability to obtain HPW_i , he cannot guess the correct PW_i . That is because r_i is a random number, only known to U_i , and α_i and derived information from U_i 's biometric factors are also secret. Additionally, the two parameters never appear in any communication channel, and \mathcal{A} does not possess the ability to crack hash functions. As a consequence, the proposed scheme can resist insider attack.

Resistance to Password Guessing Attack. Assuming that \mathcal{A} has generated the Cartesian products $\{(ID_i, PW_i)\}$ of U_i and maliciously obtained the biometric factors BIO_i and SC through the reading device, then \mathcal{A} can calculate $\alpha_i = \text{Rep}(BIO_i, \beta_i)$, $r_i = B_1 \oplus h(\alpha_i \| ID_i \| PW_i)$, and $HPW_i = h(PW_i \| \alpha_i \| r_i)$ and further check whether the equation $B_2 = h(HPW_i \| \alpha_i \| ID_i \| r_i) \bmod n_0$ holds to find out a correct password. It is noted that there are 2^{32} [24] passwords satisfying the equation, the attempts of which are enormous, thus the offline password guessing attack bounds to fail. Furthermore, honey_list records the number of user logon failure when HGWN verifies the identity of U_i , which makes it extremely unlikely that \mathcal{A} can guess the right password through online password guessing within finite attempts. Clearly, the proposed scheme can resist diverse password guessing attacks.

Resistance to Replay Attack. It is known that \mathcal{A} has the ability to eavesdrop and intercept messages over the public channel. So, \mathcal{A} may retransmit the eavesdropped or intercepted messages in a new round of the protocol implementation, to make the other party believe that "he" is legitimate to communicate with him. In the proposed protocol, however, the timestamp is employed to demonstrate the freshness of each message, so as to filter out old messages intercepted by \mathcal{A} . For an instance, \mathcal{A} has intercepted $M_1 = \{TID_i, ID_{hg}, SID_j, D_0, D_1, D_2, D_3, K_i, T_1\}$, where $D_3 = h(TID_i \| ID_i \| SID_j \| r_u \| x_i \| K_i \| T_1)$, and at time T'_1 , he attempts to resend $M'_1 = \{TID_i, ID_{hg}, SID_j, D_0, D_1, D_2, D_3, K_i, T'_1\}$ to HGWN for login. However, \mathcal{A} can only change the timestamp in the message but not that in D_3 , thus the launched replay attack bounds to fail. This instance illustrates that the proposed scheme can withstand replay attack.

User Anonymity. In terms of user anonymity, it is required that \mathcal{A} cannot find out the true identities of users or trace their communication trajectories. In this scheme, each user U_i is assigned a pseudonym TID_i , and after a round of key negotiation, his pseudonym will be updated with a new pseudonym TID'_i . Moreover, the calculation of TID'_i depends on U_i 's private key x_i and identity ID_i , neither of which is exposed to the open channel. Therefore, \mathcal{A} cannot trace the communication trajectory of the user via the pseudonym. As analysed above, user anonymity is effective.

Forward Secrecy. According to the proposed protocol, U_i 's and S_j 's private keys are both calculated by a random number and the gateway node's long-term key. It helps that even if the long-term key of the gateway node is leaked for some reason, \mathcal{A} cannot figure out U_i 's or S_j 's private key due to no idea of the random number. As the session key $SK = SK_u = SK_s = SK_{hg} = h(r_u \| r_{hg} \| r_s \| ID_{hg})$ depends on r_u , r_{hg} , as well as r_s , three of which are severally masked by private keys of three parties, \mathcal{A} cannot compute the right SK at all. Consequently, the presented scheme supports forward secrecy.

Effective Smart Card Logout. For those smart cards not used any more, improper handling may pose a huge safety hazard. On the basis of the smart card logout method described in this protocol, U_i must enter his right ID_i , PW_i , and BIO_i simultaneously while cancelling his SC, so as to prevent \mathcal{A} from launching malicious cancellation after the smart card is lost. In addition, \mathcal{A} cannot achieve password guessing attack and obtain three authentication factors at the same time, so there is no way for \mathcal{A} to masquerade as a legitimate user to cancel the smart card. Hence, the smart card logout method presented in this protocol is effective and secure.

5.2. Formal Analysis Based on BAN Logic. In the light of BAN logic, a detailed analysis in this section will illustrate that the interacting parties (U_i , HGWN, and S_j) can achieve mutual authentication and negotiate a common session key properly and securely. The analytic procedures for two cases in the proposed scheme are described as follows.

5.2.1. Security Analysis for Case 1

(i) Goals:

$$\begin{aligned}
G1: U_i | \equiv HGWN \xleftrightarrow{SK} U_i \\
G2: U_i | \equiv HGWN | \equiv HGWN \xleftrightarrow{SK} U_i \\
G3: HGWN | \equiv U_i \xleftrightarrow{SK} HGWN \\
G4: HGWN | \equiv U_i | \equiv U_i \xleftrightarrow{SK} HGWN \\
G5: HGWN | \equiv S_j \xleftrightarrow{SK} HGWN \\
G6: HGWN | \equiv S_j | \equiv S_j \xleftrightarrow{SK} HGWN \\
G7: S_j | \equiv HGWN \xleftrightarrow{SK} S_j \\
G8: S_j | \equiv HGWN | \equiv HGWN \xleftrightarrow{SK} S_j
\end{aligned}$$

(ii) Idealized forms:

$$\begin{aligned}
M_1: U_i \xrightarrow{K_i} HGWN: TID_i, ID_{hg}, SID_j, D_0, \langle r_u \rangle_{x_i} \\
D_2, \langle U_i \xleftrightarrow{SK} HGWN, r_u \rangle_{x_i} \\
M_2: HGWN \longrightarrow S_j: D_4, D_5, \langle r_u, r_{hg} \rangle_{x_j} \\
M_3: S_j \longrightarrow HGWN: D_7, \langle S_j \xleftrightarrow{SK} HGWN, r_s \rangle_{x_j} \\
M_4: HGWN \xrightarrow{SK} U_i: D_9, D_{10}, D_{11}, D_{12}, \\
\langle HGWN \xleftrightarrow{SK} U_i, x'_i, TID'_i \rangle_{K_i}
\end{aligned}$$

(iii) Assumptions:

$$\begin{aligned}
A_1: U_i | \equiv \#(r_u, r_{hg}, r_s) \\
A_2: HGWN | \equiv \#(r_u, r_{hg}, r_s) \\
A_3: S_j | \equiv \#(r_u, r_{hg}, r_s)
\end{aligned}$$

$$\begin{aligned}
A_4: U_i | &\equiv U_i \xleftrightarrow{x_i} \text{HGWN}_{x_j} \\
A_5: \text{HGWN} | &\equiv \text{HGWN} \xleftrightarrow{x_j} S_j \\
A_6: S_j | &\equiv S_j \xleftrightarrow{x_i} \text{HGWN} \\
A_7: \text{HGWN} | &\equiv \text{HGWN} \xleftrightarrow{K_i} U_i \\
A_8: U_i | &\equiv U_i \xleftrightarrow{K_i} \text{HGWN} \\
A_9: \text{HGWN} | &\equiv \text{HGWN} \xleftrightarrow{K_i} U_i \\
A_{10}: \text{HGWN} | &\equiv U_i \xrightarrow{r_u} \\
A_{11}: S_j | &\equiv \text{HGWN} \xrightarrow{r_{hg}} \\
A_{12}: \text{HGWN} | &\equiv S_j \xrightarrow{r_s} \\
A_{13}: U_i | &\equiv \text{HGWN} \xrightarrow{r_{hg}} \\
A_{14}: U_i | &\equiv \text{HGWN} \xrightarrow{\text{SK}} \text{HGWN} \xleftrightarrow{U_i} \\
A_{15}: \text{HGWN} | &\equiv S_j \xrightarrow{\text{SK}} \text{HGWN}
\end{aligned}$$

(iv) Main proofs:

From M_1 and R_6 , we can know $S_1: \text{HGWN} \triangleleft \langle r_u \rangle_{x_i}$.
From S_1 , A_4 , and R_1 , we can get $S_2: \text{HGWN} | \equiv U_i | \sim r_u$.
From S_2 , A_2 , R_2 , and R_4 , we can get $S_3: \text{HGWN} | \equiv U_i | \equiv r_u$.
From S_3 , A_{10} , and R_5 , we can get $S_4: \text{HGWN} | \equiv r_u$.
From A_2 , R_2 , and $\text{SK} = h(r_u \| r_{hg} \| r_s \| \text{ID}_{hg})$, we can get $S_5: \text{HGWN} | \equiv \#(\text{SK})$.
From S_3 , S_5 , and R_7 , we can get $S_6: \text{HGWN} | \equiv (U_i \xleftrightarrow{\text{SK}} \text{HGWN})$.
Here, we have achieved G_3 .
From S_6 , A_2 , and R_4 , we can get $S_7: \text{HGWN} | \equiv U_i | \equiv U_i \xleftrightarrow{\text{SK}} \text{HGWN}$.
Then, G_4 has been also achieved.
From M_2 and R_6 , we can know $S_8: S_j \triangleleft \langle r_u, r_{hg} \rangle_{x_i}$.
From S_8 , A_6 , and R_1 , we can gain $S_9: S_j | \equiv \text{HGWN} | \sim (r_u, r_{hg})$.
From S_9 , A_3 , R_2 , and R_4 , we can gain $S_{10}: S_j | \equiv \text{HGWN} | \equiv (r_u, r_{hg})$.
From S_{10} and R_3 , we can gain $S_{11}: S_j | \equiv \text{HGWN} | \equiv r_{hg}$.
From S_{11} , A_{11} , and R_5 , we can gain $S_{12}: S_j | \equiv r_{hg}$.
From A_3 , R_2 , and $\text{SK} = h(r_u \| r_{hg} \| r_s \| \text{ID}_{hg})$, we can gain $S_{13}: S_j | \equiv \#(\text{SK})$.
From S_{11} , S_{13} , and R_7 , we can gain $S_{14}: S_j | \equiv \text{HGWN} \xleftrightarrow{\text{SK}} S_j$. Here, G_7 has been proved.
From S_{14} , A_3 , and R_4 , we can gain $S_{15}: S_j | \equiv \text{HGWN} | \equiv \text{HGWN} \xleftrightarrow{\text{SK}} S_j$.
So, G_8 has been also gained.
From M_3 and R_6 , we can get $S_{16}: \text{HGWN} \triangleleft \langle S_j \xleftrightarrow{\text{SK}} \text{HGWN}, r_s \rangle_{x_i}$.
From S_{16} , A_5 , and R_1 , we can get $S_{17}: \text{HGWN} | \equiv S_j | \sim (S_j \xleftrightarrow{\text{SK}} \text{HGWN}, r_s)$.
From S_{17} , A_2 , R_2 , and R_4 , we can get $S_{18}: \text{HGWN} | \equiv S_j | \equiv (S_j \xleftrightarrow{\text{SK}} \text{HGWN}, r_s)$.
From S_{18} and R_3 , we can get $S_{19}: \text{HGWN} | \equiv S_j | \equiv S_j \xleftrightarrow{\text{SK}} \text{HGWN}$.
Here, we have achieved G_6 .

From S_{19} , A_{15} , and R_5 , we can get $S_{20}: \text{HGWN} | \equiv S_j \xleftrightarrow{\text{SK}} \text{HGWN}$.
So, G_5 has been also gained.
From M_4 and R_6 , we can gain $S_{21}: U_i \triangleleft \langle \text{HGWN} \xleftrightarrow{\text{SK}} U_i, x'_i, \text{TID}'_{iK_i} \rangle_{K_i}$.
From S_{21} , A_8 , and R_1 , we can obtain $S_{22}: U_i | \equiv \text{HGWN} | \sim (\text{HGWN} \xleftrightarrow{\text{SK}} U_i, x'_i, \text{TID}'_i)$.
From S_{22} , A_1 , R_2 , and R_4 , we can obtain $S_{23}: U_i | \equiv \text{HGWN} | \equiv (\text{HGWN} \xleftrightarrow{\text{SK}} U_i, x'_i, \text{TID}'_i)$.
From S_{23} and R_3 , we can obtain $S_{24}: U_i | \equiv \text{HGWN} | \equiv \text{HGWN} \xleftrightarrow{\text{SK}} U_i$.
So, we have achieved G_2 .
From S_{24} , A_{14} , and R_5 , we can obtain $S_{25}: U_i | \equiv \text{HGWN} \xleftrightarrow{\text{SK}} U_i$.
Finally, we have gained G_1 .

5.2.2. Security Analysis for Case 2

(i) Goals:

$$\begin{aligned}
G_1: U_i | &\equiv \text{FGWN} \xleftrightarrow{\text{SK}} U_i \\
G_2: U_i | &\equiv \text{FGWN} | \equiv \text{FGWN} \xleftrightarrow{\text{SK}} U_i \\
G_3: \text{FGWN} | &\equiv U_i \xleftrightarrow{\text{SK}} \text{FGWN} \\
G_4: \text{FGWN} | &\equiv U_i | \equiv U_i \xleftrightarrow{\text{SK}} \text{FGWN} \\
G_5: \text{FGWN} | &\equiv S_j \xleftrightarrow{\text{SK}} \text{FGWN} \\
G_6: \text{FGWN} | &\equiv S_j | \equiv S_j \xleftrightarrow{\text{SK}} \text{FGWN} \\
G_7: S_j | &\equiv \text{FGWN} \xleftrightarrow{\text{SK}} S_j \\
G_8: S_j | &\equiv \text{FGWN} | \equiv \text{FGWN} \xleftrightarrow{\text{SK}} S_j
\end{aligned}$$

(ii) Idealized forms:

$$\begin{aligned}
M_5: U_i &\longrightarrow \text{FGWN}: \text{TID}_i, D_9, \langle r'_u \rangle_{x_g} \\
M_6: \text{FGWN} &\longrightarrow S_j: D_{11}, D_{12}, \langle r'_u, r'_{fg} \rangle_{x_j} \\
M_7: S_j &\longrightarrow \text{FGWN}: D_{14}, \langle \text{FGWN} \xleftrightarrow{\text{SK}} S_j, r_s \rangle_{x_j} \\
M_8: \text{FGWN} &\longrightarrow U_i: D_{16}, D_{17}, \langle \text{FGWN} \xleftrightarrow{\text{SK}} U_i, r'_{fg}, r_s \rangle_{x_g}
\end{aligned}$$

(iii) Assumptions:

$$\begin{aligned}
A_1: U_i | &\equiv \#(r'_u, r'_{fg}, r_s) \\
A_2: \text{FGWN} | &\equiv \#(r'_u, r'_{fg}, r_s) \\
A_3: S_j | &\equiv \#(r'_u, r'_{fg}, r_s) \\
A_4: U_i | &\equiv U_i \xleftrightarrow{x_g} \text{FGWN} \\
A_5: \text{FGWN} | &\equiv \text{FGWN} \xleftrightarrow{x_j} S_j \\
A_6: S_j | &\equiv S_j \xleftrightarrow{x_j} \text{FGWN} \\
A_7: \text{FGWN} | &\equiv \text{FGWN} \xleftrightarrow{x_g} U_i \\
A_8: \text{FGWN} | &\equiv U_i \xrightarrow{r'_u} \\
A_9: S_j | &\equiv \text{FGWN} \xrightarrow{r'_{fg}} \\
A_{10}: \text{FGWN} | &\equiv S_j \xrightarrow{r_s} \\
A_{11}: U_i | &\equiv \text{FGWN} \xrightarrow{r'_{fg}} \\
A_{12}: U_i | &\equiv \text{FGWN} \xrightarrow{\text{SK}} \text{FGWN} \xleftrightarrow{U_i} \\
A_{13}: \text{FGWN} | &\equiv S_j \xrightarrow{\text{SK}} \text{FGWN}
\end{aligned}$$

(iv) Main proofs:

From M_5 and R_6 , we obtain S26: $\text{FGWN} \triangleleft \langle r'_u \rangle_{x_g}$.

From S26, A_7 , and R_1 , we obtain S27: $\text{FGWN} | \equiv U_i | \sim r'_u$.

From S27, A_2 , R_2 , and R_4 , we obtain S28: $\text{FGWN} | \equiv U_i | \equiv r'_u$.

From S28, A_8 , and R_5 , we obtain S29: $\text{FGWN} | \equiv r'_u$.

From A_2 , R_2 , and $\text{SK} = h(r'_u \| r_{fg} \| r_s \| \text{ID}_{fg})$, we obtain S30: $\text{FGWN} | \equiv \#(\text{SK})$.

From S28, S30, and R_7 , we obtain S31: $\text{FGWN} | \equiv U_i \xleftrightarrow{\text{SK}} \text{FGWN}$.

So, G_3 has been achieved.

From S31, A_2 , and R_4 , we obtain S32: $\text{FGWN} | \equiv U_i | \equiv U_i \xleftrightarrow{\text{SK}} \text{FGWN}$.

Here, G_4 has been also obtained.

From M_6 and R_6 , we get S33: $S_j \triangleleft \langle r'_u, r_{fg} \rangle_{x_j}$.

From S33, A_6 , and R_1 , we get S34: $S_j | \equiv \text{FGWN} | \sim (r'_u, r_{fg})$.

From S34, A_3 , R_2 , and R_4 , we get S35: $S_j | \equiv \text{FGWN} | \equiv (r'_u, r_{fg})$.

From S35 and R_3 , we get S36: $S_j | \equiv \text{FGWN} | \equiv r_{fg}$.

From S36, A_9 , and R_5 , we get S37: $S_j | \equiv r_{fg}$.

From A_3 , R_2 , and $\text{SK} = h(r'_u \| r_{fg} \| r_s \| \text{ID}_{fg})$, we get S38: $S_j | \equiv \#(\text{SK})$.

From S36, S38, and R_7 , we get S39: $S_j | \equiv \text{FGWN} \xleftrightarrow{\text{SK}} S_j$. Here, we have proved G_7 .

From S39, A_3 , and R_4 , we get S40: $S_j | \equiv \text{FGWN} | \equiv \text{FGWN} \xleftrightarrow{\text{SK}} S_j$.

Here, we have achieved G_8 .

From M_7 and R_6 , we gain S41: $\text{FGWN} \triangleleft \langle \text{FGWN} \xleftrightarrow{\text{SK}} S_j, r_s \rangle_{x_j}$.

From S41, A_5 , and R_1 , we gain S42: $\text{FGWN} | \equiv S_j | \sim (\text{FGWN} \xleftrightarrow{\text{SK}} S_j, r_s)$.

From S42 and R_3 , we gain S43: $\text{FGWN} | \equiv S_j | \sim \text{FGWN} \xleftrightarrow{\text{SK}} S_j$.

From A_2 , R_2 , and $\text{SK} = h(r'_u \| r_{fg} \| r_s \| \text{ID}_{fg})$, we gain S44: $\text{FGWN} | \equiv \#(\text{SK})$.

From S43, S44, R_2 , and R_4 , we gain S45: $\text{FGWN} | \equiv S_j | \equiv \text{FGWN} \xleftrightarrow{\text{SK}} S_j$.

Here, we have achieved G_6 .

From A_{13} , S45, and R_5 , we gain S46: $\text{FGWN} | \equiv \text{FGWN} \xleftrightarrow{\text{SK}} S_j$.

So, we have also achieved G_5 .

From M_8 and R_6 , we know S47: $U_i \triangleleft \langle \text{FGWN} \xleftrightarrow{\text{SK}} U_i, r_{fg}, r_s \rangle_{x_g}$.

From S47, A_4 , and R_1 , we get S48: $U_i | \equiv \text{FGWN} | \sim (\text{FGWN} \xleftrightarrow{\text{SK}} U_i, r_{fg}, r_s)$.

From S48 and R_3 , we get S49: $U_i | \equiv \text{FGWN} | \sim \text{FGWN} \xleftrightarrow{\text{SK}} U_i$.

From A_1 , R_2 , and $\text{SK} = h(r'_u \| r_{fg} \| r_s \| \text{ID}_{fg})$, we get S50: $U_i | \equiv \#(\text{SK})$.

From S49, S50, R_2 , and R_4 , we get S51: $U_i | \equiv$

$\text{FGWN} | \equiv \text{FGWN} \xleftrightarrow{\text{SK}} U_i$.

So, G_2 has been gained.

From S51, A_{12} , and R_5 , we get S52: $U_i | \equiv \text{FGWN} \xleftrightarrow{\text{SK}} U_i$. So, G_1 has been also obtained.

Consequently, all security goals are amply demonstrated, both in Case 1 and in Case 2. In the meantime, it also confirms that the communication participants (U_i , HGWN/FGWN , and S_j), can authenticate mutually and negotiate a common key successfully.

5.3. Formal Analysis Based on BPR Model

Theorem 1. For the protocol \mathcal{P} , assuming that, in a polynomial time t , \mathcal{A} makes up to q_s $\text{Send}(\Pi_i^*, m)$ queries, q_e $\text{Excute}(\Pi_U^i, \Pi_{\text{HGWN}}^k, \Pi_S^j)$ queries, and q_h oracle queries. Let \mathcal{D} represent the password space subject to Zipf distribution, wherein C' and s' are Zipf parameters; let l denote the output length of hash functions. Now, we can get

$$\% \text{Adv}_{\mathcal{P}}^{\text{AKE}}(\mathcal{A}) \leq 2C'q_s^{s'} + \frac{q_s}{2^{l-1}} + \frac{q_h^2}{2^l} + \frac{(q_s + q_e)^2}{p-1}. \quad (8)$$

Proof. Five games \mathcal{E}_i ($i = 0, 1, 2, 3$, and 4) are considered to demonstrate Theorem 1, and simulation process of each game is analysed as below, wherein S_i indicates an event that \mathcal{A} outputs the right random bit b in \mathcal{E}_i , where $i = 0, 1, 2, 3$, and 4.

\mathcal{E}_0 : it simulates a true attack under the random oracle model. \mathcal{A} has the ability to access all oracles; so according Definition 1, we have

$$\text{Adv}_{\mathcal{P}}^{\text{AKE}}(\mathcal{A}) = 2\Pr[S_0] - 1. \quad (9)$$

\mathcal{E}_1 : it maintains two lists, L_H and L_M , respectively, recording oracle queries and communications during the execution of \mathcal{P} . Besides, all other queries are run as the actual protocol. In \mathcal{E}_1 , \mathcal{A} launches the passive attack to intercept all messages M_j ($j = 1, 2, 3, 4$) through $\text{Excute}(\ast)$ query and then guesses the output result of $\text{Test}(\Pi_i^*)$ query. Due to the impossibility of figuring out $\text{SK} = h(r_u \| r_{hg} \| r_s \| \text{ID}_{hg})$, the advantage of a successful attack does not increase for \mathcal{A} , so we can get

$$\Pr[S_1] = \Pr[S_0]. \quad (10)$$

\mathcal{E}_2 : here, \mathcal{A} can make $\text{Send}(\Pi_i^*, m)$ queries and \mathcal{H} queries to convince the true communicator of forged messages. Only when \mathcal{A} happens to find some collisions and succeeds in constructing credible messages, the simulation terminates. In \mathcal{E}_2 , two kinds of collisions may be contained: output collisions of hash functions and collisions of random numbers selected in \mathcal{P} . According to Birthday Paradox [30], the probabilities of their occurrence are $(q_h^2/2^{l+1})$ and $((q_s + q_e)^2/2(p-1))$, respectively. Therefore, we obtain

$$|\Pr[S_2] - \Pr[S_1]| \leq \frac{q_h^2}{2^{l+1}} + \frac{(q_s + q_e)^2}{2(p-1)}. \quad (11)$$

\mathcal{E}_3 : this game differs from the above games in the case that when \mathcal{A} can guess the correct authentication factors D_3 , D_6 , D_8 , and D_{13} without \mathcal{H} queries, the simulation terminates. It is indistinguishable from the previous games except that some instance refuses the right authentication. Thus, we have

$$|\Pr[S_3] - \Pr[S_2]| \leq \frac{q_s}{2}. \quad (12)$$

\mathcal{E}_4 : in this game, \mathcal{A} has abilities to reach more information through $\text{Corrupt}(\Pi_U^i, a)$ query.

- (i) \mathcal{A} queries $\text{Corrupt}(\Pi_U^i, 1)$, which means he has got the user's password and parameters stored in SC. Then, in q_s $\text{Send}(\Pi_U^*, m)$ queries, \mathcal{A} succeeds in guessing α_i with the length l_α , the possibility of which is $(q_s/2^{l_\alpha})$.
- (ii) \mathcal{A} queries $\text{Corrupt}(\Pi_U^i, 2)$, that is, \mathcal{A} has accessed the user's biometric factors and parameters stored in SC. Then, in q_s $\text{Send}(\Pi_U^*, m)$ queries, \mathcal{A} succeeds in guessing the victim's password, the possibility of which is $C'q_s^{s'}$.
- (iii) \mathcal{A} queries $\text{Corrupt}(\Pi_U^i, 3)$; similarly, \mathcal{A} has the user's password and biometric factors. Then, the possibility of \mathcal{A} guessing the right x_i is $(q_s/2^l)$.

\mathcal{E}_4 and \mathcal{E}_3 are indistinguishable unless the above attack is successful. So, we have

$$|\Pr[S_4] - \Pr[S_3]| \leq \max\left\{\frac{q_s}{2^{l_\alpha}}, C'q_s^{s'}, \frac{q_s}{2^l}\right\} = C'q_s^{s'}. \quad (13)$$

When \mathcal{A} has no efficient input to make queries to \mathcal{H} , there is no advantage to distinguish the real SK from a random number with the same size through $\text{Test}(\Pi_U^*)$. Therefore,

$$\Pr[S_4] = \frac{1}{2}. \quad (14)$$

From (2)–(7), we can draw conclusion (1) or (8); this is

$$\begin{aligned} \text{Adv}_{\mathcal{F}}^{\text{AKE}}(\mathcal{A}) &= 2|\Pr[S_4] - \Pr[S_0]| \leq 2C'q_s^{s'} \\ &+ \frac{q_s}{2^{l-1}} + \frac{q_h^2}{2^l} + \frac{(q_s + q_e)^2}{p-1}. \end{aligned} \quad (15)$$

□

6. Performance Comparison

In this section, the proposed protocol is compared with several existing multifactor authentication protocols in terms of performance, involving security features, computation overhead, and storage costs. Specific comparison results and analysis are described as follows.

6.1. Security Features. On the basis of the security 12-Criteria, Table 3 presents the comparison results of these diverse authentication protocols, i.e., Guo et al. [22], Wu et al. [20], Srinivas et al. [21], Amin [19], and our proposed protocol. Definitely, the proposed protocol can satisfy all 12 evaluation criteria whereas others can meet 8 pieces at most. In particular, the new protocol in this paper exclusively provides the repairability and forward security, as well as resistance against stolen smart card attack. The protocol presented by Guo et al. [22] has weaknesses in no repairability, improper treatment of biometric features, and offline password guessing attack; the protocol of Wu et al. [20] cannot resist insider attack, stolen smart card attack, and offline password guessing attack; the protocol proposed by Srinivas et al. [21] does not protect against insider attack and offline password guessing attack and ensure that the user will be not traced; Amin's protocol [19] does not provide resistance to insider attack and guarantee of untraceability of the user. Furthermore, none of these protocols, except the proposed one, implements forward secrecy.

It should be noted that, the 12 security evaluation criteria was proposed by Wang and Wang [24]: C1 for no password verifier-table; C2 for password-friendly; C3 for no password exposure; C4 for no smart card loss attack; C5 for resistance to known attacks; C6 for sound repairability; C7 for provision of key agreement; C8 for no clock synchronization; C9 for timely typo detection; C10 for mutual authentication; C11 for user anonymity; C12 for forward secrecy.

6.2. Computation Overhead. In this section, we compare the computation overhead among the above relevant schemes. In reality, login and authentication are much more frequent than registration, thus the performance of authentication and key-agreement protocols depends primarily on the computational costs of login and authentication phases. As depicted in Table 4, the proposed scheme is more computationally expensive than other schemes at the user side. This happens unsurprisingly because that fuzzy extractor is employed in this paper to extract and verify the biometric features, which is more applicable for high security systems. As for the gateways and resource-constrained sensor nodes, the computational costs are nearly the same. At any side, the schemes proposed by Wu et al. [20] and Amin [19] have the least computational overhead as they trade low safety features for high efficiency. In summary, despite other schemes outperforming in computational complexity, the proposed scheme can protect against all security threats faced by other schemes, which is more feasible in the real world.

6.3. Storage Costs. Comparison of storage costs among the proposed scheme and other relevant schemes is stated in this section, see Table 5 and Figure 5. Primarily, it is recommended that 32 bits for the (pseudo-) identity, 160 bits for the hash output, 128 bits for the fuzzy extractor public data, and 128 bits for a random number, as well as 32 bits for a timestamp are agreed, and these parameters are denoted separately as L_{ID} , L_h , L_{fe} , L_r , and L_T . As shown in Figure 5, storage overhead on the user and sensor nodes sides is nearly the same, but that on the gateway nodes is higher as in the proposed scheme; smart card logout is achieved with the assistance of honey_list saving in

TABLE 3: Comparison of security features.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Guo et al. [22]	✓	✓	✓	✓	×	×	✓	✓	×	✓	✓	×
Wu et al. [20]	✓	✓	×	×	×	×	✓	×	×	✓	✓	×
Srinivas et al. [21]	✓	✓	×	×	✓	×	✓	×	×	✓	×	×
Amin [19]	✓	✓	×	×	×	×	✓	×	×	✓	×	×
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

“✓” means the protocol satisfies this property; “×,” the opposite.

TABLE 4: Comparison of computation overhead.

		Guo et al. [22]	Wu et al. [20]	Srinivas et al. [21]	Amin [19]	Ours
U_i	Case 1	$13T_h$	$9T_h$	$10T_h$	$7T_h$	$13T_h + T_{fe}$
	Case 2	$18T_h$	$11T_h$	$14T_h$	$8T_h$	$15T_h + T_{fe}$
HGWN	Case 1	$17T_h$	$11T_h$	$14T_h$	$8T_h$	$18T_h$
	Case 2	$10T_h$	$7T_h$	$6T_h$	$1T_h$	$11T_h$
FGWN	Case 1	0	0	0	0	0
	Case 2	$14T_h$	$7T_h$	$17T_h$	$7T_h$	$12T_h$
S_j	Case 1	$6T_h$	$4T_h$	$7T_h$	$5T_h$	$6T_h$
	Case 2	$6T_h$	$4T_h$	$6T_h$	$5T_h$	$6T_h$
Total	Case 1	$36T_h$	$24T_h$	$31T_h$	$20T_h$	$37T_h + T_{fe}$
	Case 2	$48T_h$	$29T_h$	$43T_h$	$21T_h$	$44T_h + T_{fe}$

TABLE 5: Comparison of storage costs.

	Guo et al. [22]	Wu et al. [20]	Srinivas et al. [21]	Amin [19]	Ours
SC	$L_{ID} + 3L_h + L_r$	$L_{ID} + 3L_h + L_r$	$L_{ID} + 4L_h$	$2L_{ID} + 3L_h + L_r$	$2L_{ID} + 3L_h + L_{fe}$
HGWN/FGWN	$3L_{ID} + 2L_r$	$3L_{ID} + 2L_r$	$4L_{ID} + L_r + L_T$	$4L_{ID} + L_r + L_h$	$3L_{ID} + 4L_r + L_h$
S_j	$2L_{ID} + L_h$	$2L_{ID} + L_h$	$L_{ID} + L_h + L_T$	$L_{ID} + L_h$	$L_{ID} + L_h$

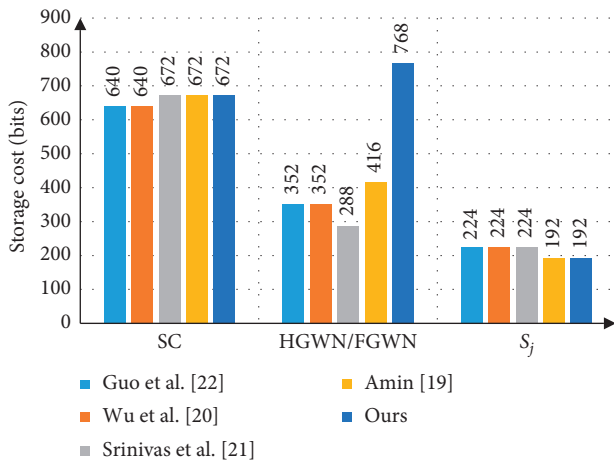


FIGURE 5: Comparison of storage costs.

gateway nodes’ memories. However, in terms of storage capacity, gateway nodes are much better than smart cards and sensor nodes, thus the overhead is acceptable.

7. Conclusion

WSNs are becoming increasingly vital in IoT applications. Inevitably, multifactor and multigateway authentication

protocols have become a focus. In this paper, through analysing weaknesses in the existing schemes, we introduced the widely accepted criteria for evaluating security protocols. In line with the criteria, we revisited Guo et al.’s scheme and found some security flaws, i.e., no repairability, improper treatment of biometric factors, offline password guessing, and no forward secrecy. Then, we proposed a new three-factor authentication protocol for multiple gateways using fuzzy extractor and honey_list technique. Following that, we proved the correctness and security of the proposed scheme by BAN logic and BPR model. As a whole, our proposed scheme outperformed other relevant schemes for keeping efficient in performance, meanwhile satisfying the security criteria.

Data Availability

No data were used to support this study.

Conflicts of Interest

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

Authors’ Contributions

L. Xue and Q. Huang contributed equally to this work.

Acknowledgments

This work was supported in part by the National Key Research and Development Program (2019YFB2101704 and 2018YFB0803403), National Natural Science Foundation of China (61872194 and 62072252), and Key Project on Anhui Provincial Natural Science Study by Colleges and Universities (KJ2019A0579, KJ2020A0513 and KJ2020A0497).

References

- [1] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
- [2] R. Hajian, S. ZakeriKia, S. H. Erfani, and M. Mirabi, "SHAPARAK: scalable healthcare authentication protocol with attack-resilience and anonymous key-agreement," *Computer Networks*, vol. 183, Article ID 107567, 2020.
- [3] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.
- [4] C.-C. Chang and T.-C. Wu, "Remote password authentication with smart cards," *IEE Proceedings E Computers and Digital Techniques*, vol. 138, no. 3, pp. 165–168, 1991.
- [5] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.
- [6] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.
- [7] T.-H. Chen and W.-K. Shih, "A robust mutual authentication protocol for wireless sensor networks," *ETRI Journal*, vol. 32, no. 5, pp. 704–712, 2010.
- [8] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor user authentication scheme in wireless sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 10, pp. 361–371, 2010.
- [9] S. G. Yoo, K. Y. Park, and J. Kim, "A security-performance-balanced user authentication scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 8, no. 3, Article ID 382810, 2012.
- [10] P. Kumar and H.-J. Lee, "Cryptanalysis on two user authentication protocols using smart card for wireless sensor networks," in *Proceedings of the 2011 Wireless Advanced*, pp. 241–245, London, UK, June 2011.
- [11] K. Xue, C. Ma, P. Hong, and R. Ding, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 316–323, 2013.
- [12] C.-T. Li, C.-Y. Weng, and C.-C. Lee, "An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks," *Sensors*, vol. 13, no. 8, 2013.
- [13] Q. Jiang, J. Ma, X. Lu, and Y. Tian, "An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 8, no. 6, pp. 1070–1081, 2015.
- [14] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Information Sciences*, vol. 321, pp. 263–277, 2015.
- [15] C.-T. Li and M.-S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010.
- [16] J.-J. Yuan, "An enhanced two-factor user authentication in wireless sensor networks," *Telecommunication Systems*, vol. 55, 2013.
- [17] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K.-K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network and Computer Applications*, vol. 103, pp. 194–204, 2018.
- [18] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for internet of things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.
- [19] R. Amin, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Networks*, vol. 36, 2015.
- [20] F. Wu, L. Xu, S. Kumari et al., "An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment," *Journal of Network and Computer Applications*, vol. 89, 2016.
- [21] J. Srinivas, S. Mukhopadhyay, and D. Mishra, "Secure and efficient user authentication scheme for multi-gateway wireless sensor networks," *Ad Hoc Networks*, vol. 54, pp. 147–169, 2017.
- [22] H. Guo, Y. Gao, T. Xu, X. Zhang, and J. Ye, "A secure and efficient three-factor multi-gateway authentication protocol for wireless sensor networks," *Ad Hoc Networks*, vol. 95, Article ID 101965, 2019.
- [23] R. Vinoth, L. J. Deborah, P. Vijayakumar, and N. Kumar, "Secure multi-factor authenticated key agreement scheme for industrial IoT," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3801–3811, 2020.
- [24] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2018.
- [25] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology—EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds., Springer, Berlin, Germany, pp. 523–540, 2004.
- [26] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," in *Proceedings of the Twelfth ACM Symposium on Operating Systems Principles*, Litchfield Park, AZ, USA, November 1989.
- [27] M. Tunstall, K. E. Mayes, and K. Markantonakis, "Smart card security. secure smart embedded devices, platforms and applications," 2014.
- [28] F. Hao, "On robust key agreement based on public key authentication," *Security & Communication Networks*, vol. 7, no. 1, pp. 77–87, 2014.
- [29] E. Bresson, O. Chevassut, and D. Pointcheval, "Security proofs for an efficient password-based key exchange," in *Proceedings of the 10th ACM Conference on Computer and Communication Security: CCS '03*, pp. 241–250, Washington, DC, USA, October 2003.
- [30] V. Boyko, P. MacKenzie, and S. Patel, "Provably secure password-authenticated key exchange using diffie-hellman," in *Advances in Cryptology—EUROCRYPT 2000*, B. Preneel, Ed., Springer, Berlin, Germany, pp. 156–171, 2000.

Research Article

A Privacy-Preserving Identity Authentication Scheme Based on the Blockchain

Sheng Gao ¹, Qianqian Su ², Rui Zhang ², Jianming Zhu,¹ Zhiyuan Sui,¹
and Junsheng Wang³

¹School of Information, Central University of Finance and Economics, Beijing 100086, China

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

³State Grid Blockchain Technology Laboratory, State Grid E-Commerce Co., Ltd., Beijing 100053, China

Correspondence should be addressed to Qianqian Su; suqianqian@iie.ac.cn

Received 1 April 2021; Accepted 23 May 2021; Published 4 June 2021

Academic Editor: James Ying

Copyright © 2021 Sheng Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traditional identity authentication solutions mostly rely on a trusted central entity, so they cannot handle single points of failure well. In addition, most of these traditional schemes need to store a large amount of identity authentication or public key information, which makes the schemes difficult to expand and use in distributed situations. In addition, the user prefers to protect the privacy of their information during the identity verification process. Due to the open and decentralized nature of the blockchain, the existing identity verification schemes are difficult to apply well in the blockchain. To solve this problem, in this article, we propose a privacy protection identity authentication scheme based on the blockchain. The user independently generates multiple-identity information, and these identities can be used to apply for an identity certificate. Authorities use the ECDSA signature algorithm and the RSA encryption algorithm to complete the distribution of the identity certificate based on the identity information and complete the registration of identity authentication through the smart contract on the blockchain. On the one hand, it can realize the protection of real identity information; on the other hand, it can avoid the storage overhead caused by the need to store a large number of certificates or key pairs. Due to the use of the blockchain, there is no single point of failure in the authentication process, and it can be applied to distributed scenarios. The security and performance analysis show that the proposed scheme can meet security requirements and is feasible.

1. Introduction

Nowadays, in the Internet of Things (IoT) environment, a massive quantity of devices and sensors can feel each other through the internet to share and process data [1–3]. Users have lost control of sensitive data, which has caused concerns about data security to become one of the main obstacles to data sharing between parties [4, 5]. Take e-health systems as an example; with the popularity of wearable medical equipment, the application of the e-health system has obtained widespread attention and is constantly changing our living habits [6–8]. Through the e-health system, doctors can analyze the patient's physique data obtained by sensors in real time, realize the research on the

effect of drugs, or provide patients with better medical care. In these scenarios, wearable medical sensors can obtain parameters related to the patient's health, such as blood pressure, heart rate, and body temperature. Through the internet, the collected health data are transmitted to the doctors. Internet-based medical treatment enables doctors to treat patients no longer limited to geographic locations, which not only reduces medical costs but also saves treatment time. Even if the patient is located in a remote area, doctors can monitor the patient's health in real time through the transmitted data and give targeted treatment plans.

In this scenario, the patient's medical data are an important information resource containing a large amount of sensitive information, which can be in the form of signals,

text, voice data, images, etc. This information needs to be effectively protected. However, since medical systems are vulnerable to cyberattacks, sharing sensitive patient information in an IoT environment may cause a series of serious security and privacy issues. For example, if the third party who obtains the information does not use the data as agreed, but instead sells or uses other forms of data abuse; this will pose a severe challenge to the privacy and safety of patients. In order to ensure that patients' data are not used by unauthorized people in the smart medical environment, an effective identity management solution must be used. Firstly, the amount of data generated by sensors in real-time medical treatment is very large, and the data formats are heterogeneous. Therefore, for terminals with limited processing capabilities, it is not feasible to encrypt data before transmitting the data. Secondly, since terminals often have limited storage capacity, it is not feasible to use existing identity management and verification methods that require storing a large number of key pairs. In addition, most of the existing solutions rely on a trusted third party to implement identity management and authentication, which not only leads to the potential danger of a single point of failure but also makes users lose control of their own identity information.

Recently, as a decentralized technology, blockchain [9–11] provides a feasible solution to ensure the data integrity. The advantage of blockchain technology is that, through the consensus mechanism, the distributed storage of medical data can be realized, and the modification or deletion of the data of a few participants will not affect other participants. It is an interesting idea to use blockchain to solve the problem of relying on trusted third parties in traditional identity authentication. For the key management [12] and user identity authentication [13], it is also necessary to resolve user anonymity, verifiability, and nonrepudiation [14–16].

In this paper, we propose a blockchain-based identity authentication scheme, which can realize anonymous user identity authentication and identity management without a lot of storage space. The main contributions of this paper can be summarized as follows:

- (i) We propose a blockchain-based identity authentication scheme. By introducing the blockchain, users will generate their own identities and generate publicly verifiable information for those identities. Users store public information on the blockchain, thus solving the problem of relying on third parties to manage identity information. Users do not need to maintain a database of publicly verified information and can realize identity authentication by querying the blockchain, which saves the time delay of waiting for block confirmation. Therefore, during the identity authentication process, there is no need to rely on a trusted third party, and there is no need for users to store the identity information of other users.
- (ii) The identity authentication scheme we proposed can support privacy preservation, including

communication privacy protection and user identity privacy protection. By using the ECDSA signature scheme, the verifiability and unforgeability of the identity verification process are ensured. The communication process is encrypted by the RSA encryption algorithm to ensure the security of communication. In addition, the user can generate multiple identities, and the corresponding public information is not related, so the user's identity privacy can be effectively protected.

- (iii) Our analysis and comparison proved that the proposed identity authentication scheme meets the security requirements, and the feasibility of the scheme was proved through simulation experiments.

1.1. Related Work. Traditional identity management solutions often rely on a centralized trusted third party [17, 18], where users' personally identifying information is controlled by an organization rather than the user himself/herself. This means that the third party has complete control over the user's information. Third-party entities may leak user information due to software vulnerabilities, hardware damage, and economic benefits. In addition, a centralized system inevitably brings a single point of failure problem, and due to the limited capacity of a single node, it is difficult to achieve effective identity authentication when the system user is very large, that is, it lacks scalability.

In order to solve the centralization problem, some studies have proposed federated identity management [15, 19, 20]. Allow users to log in to the system with the same identity in multiple different scenarios. Although this solution avoids the storage of a large amount of identity information to some extent, the user's identity is still controlled and managed by the joint service provider. At the same time, there have been many proposed schemes to help meet user privacy protection requirements [21–24]. They focus on user-centric identity management, enabling users to selectively authorize personal data under various conditions and display credentials provided in response to authentication requests.

Recently, some researchers have introduced blockchain technology into identity authentication [25–27]. In [15], the authors proposed a blockchain-based identity management and authentication scheme for mobile networks, where users' identifying information is controlled by the users themselves. In [13], the authors proposed a blockchain-based multi-WSN authentication scheme for IoT. In their scheme, the nodes of IoT are divided into base stations, cluster head nodes, and ordinary nodes according to capability, which are formed to a hierarchical network. A blockchain network is constructed among different types of nodes to form a hybrid blockchain model, including local chain and public chain. In this hybrid model, nodes' identity mutual authentication in various communication scenarios is realized, ordinary node identity authentication operation is accomplished by the local blockchain, and cluster head

node identity authentication is realized in the public blockchain. In [28], the authors proposed a new EHR paradigm which can help in dealing with the centralized problem of cloud-based EHRs. After that, they proposed an authentication scheme for blockchain-based EHRs. The proposed scheme is an identity-based signature scheme with multiple authorities which can resist the collusion attack out of N from $N - 1$ authorities. In [29], the authors presented a permissioned blockchain-based identity management and user authentication (PBBIMUA) scheme for the e-health environment. The proposed scheme satisfies the security requirements of medical data.

It can be seen that the existing blockchain-based identity authentication schemes can be divided into two categories according to their application scenarios: multidomain and single domain. Among the multidomain authentication schemes, the existing schemes are difficult to solve the cross-domain system compatibility issues and the privacy security issues between different domains. In the single-domain authentication scheme, most of the information used for authentication is stored in the blockchain in plaintext messages. However, in the process of identity management and authentication, the openness and immutability of the blockchain will inevitably bring security risks and difficulties in changing identity information.

2. Preliminaries

In this section, we illustrate background knowledge used in this paper, including the definition of discrete logarithm and its security assumptions, chameleon hash algorithm, and description of the verifiable claim.

2.1. Blockchain. Blockchain [30] is a distributed hyperledger with irreversibility and traceability. Generally, the blockchain integrates various technologies such as cryptographic algorithms, P2P communication, consensus, and smart contracts and can establish trust relationships without a special trust relationship between peers and no trusted central authority. Cryptographic algorithms, such as hash functions and signature algorithms, can guarantee the integrity and unforgeability of information. P2P technology can realize point-to-point communication between nodes. The consensus mechanism (such as PoW, PoS, and DPoS) is the core of the blockchain. The nodes participating in the consensus in the blockchain system are called miners. They are responsible for packaging the transaction data in the system into a block and obtain the accounting rights by participating in the consensus, thereby recording the block on the blockchain.

2.2. Elliptic Curve Digital Signature Algorithm. Elliptic Curve Digital Signature Algorithm (ECDSA) [31] is used to create a digital signature of the data (a file, for example) in order to allow one to verify their authenticity without compromising their security. We use Sign and Verify to represent the signing process and the verification process in ECDSA, respectively.

The signing process is as follows:

- (1) Choose an elliptic curve $E_p(a, b)$ and the base point G
- (2) Select the private key k ($k < n$, n is the order of G), and use the base point G to calculate the public key $K = kG$
- (3) Generate a random integer r ($r < n$), and calculate the point $R = rG$
- (4) Take the original data m and the coordinate values x, y of point R as parameters, and calculate $h = \text{Hash}(m, x, y)$
- (5) Calculate $s = r - h * k \text{ mod } n$
- (6) As the signature value, r and s , if one of r and s is 0, restart from Step 3

The verification process is as follows. After receiving the message m and signature value (r, s) , the recipient performs the following operations:

- (1) Calculation: $sG + H(m)P = (x_1, y_1)$ and $r_1 = x_1 \text{ mod } p$
- (2) Verify the equation: $r_1 = r \text{ mod } p$
- (3) If the equation holds, accept the signature; otherwise, the signature is invalid

2.3. RSA Encryption. RSA encryption algorithm [32, 33] is an asymmetric key encryption algorithm. The encryption key (i.e., public key) PK is public information, and the decryption key (i.e., secret key) SK needs to be kept secret. The encryption algorithm Enc and the decryption algorithm Dec are also public.

The specific description of the RSA algorithm is as follows:

- (1) Choose two different large prime numbers p and q to calculate the product $n = pq$, $\varphi(n) = (p - 1)(q - 1)$
- (2) Choose a large integer e arbitrarily and satisfy $\text{gcd}(e, \varphi(n)) = 1$, and the integer e is used as the encryption key
- (3) The determined solution key d satisfies $e d = 1 \text{ mod } \varphi(n)$
- (4) The integers n and e are disclosed, and d is kept secret
- (5) Encrypt the plaintext m ($m < n$ is an integer) into ciphertext c ; the encryption algorithm is $c = \text{Enc}(e, m) = m^e \text{ mod } n$
- (6) Decrypt ciphertext c into plaintext m ; the decryption algorithm is $m = \text{Dec}(d, c) = c^d \text{ mod } n$

3. System Model

As shown in Figure 1, in the system model, we assume a blockchain network in which each member holds a related distributed ledger. The network systems are formed with the data owner (DO) and the data user (DU). In the e-health system, data owners are generally patients with wearable medical equipment, and data users are doctors

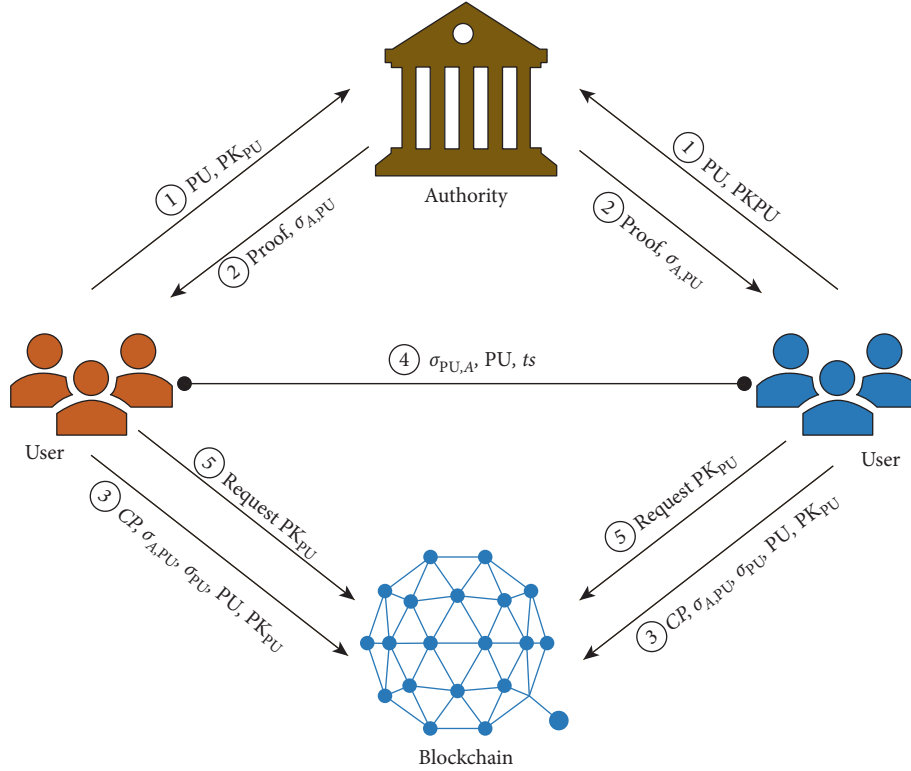


FIGURE 1: System model.

who provide medical assistance to patients. Users establish a blockchain network maintained by miners. Authority is responsible for the registration of users and providing the proof of their valid identity. The responsibility of the miner is to check the user's identity information and add this information to the blockchain as a transaction for mining user enrolment requests. After successful execution of the process, users can complete the authentication process by accessing the blockchain.

- (i) User: the user realizes its own identity control and management by generating its own identity identification (PU) and its corresponding public and private keys. The user can have multiple independent PUs at the same time and store PUs and public and private keys locally. Only when necessary, the PU and public key are disclosed to other users. According to different roles, users can be divided into data owners (DO, such as patients) and data users (DU, such as doctors).
- (ii) Authority: the authority is an entity that distributes certificates to users (Steps ① and ②), such as governments or medical management agencies. The certificate distributed to users contains the signature of the authorities and can be verified by other users. It is worth pointing out that although the authority distributes certificates to users, the authority does not participate in the verification process in the process of performing identity authentication (Step ④).

- (iii) Blockchain: It is a consortium blockchain maintained by miners for publishing users' PU and public keys (Step ③). The miner is the execution node of the packaged transaction block in the blockchain. It verifies the signature of the transaction and stores the verified transaction on the blockchain. Any entity can read the information on the blockchain (Step ⑤).

4. The Proposed Identity Authentication Scheme

In this section, we first give the overview of our proposed privacy-preserving identity authentication scheme. In the following, we provide a detailed description of our scheme, which mainly consists of three phases: initialization, registration, and authentication.

4.1. Overview. In the privacy-preserving identity authentication scheme, the user independently generates their identity information (PU) and corresponding public and private key pairs (PK, SK). Before implementing the authentication process, the user should send PU and PK to the authority for registration in order to obtain a valid identity proof (\mathcal{PF}). It is worth noting that the user can generate several different (PU, PK, SK) certifications by different authorities to obtain multiple verifiable proofs. In order to achieve the authentication process, the user sends the publicly verifiable identity proof generated by the authority and the corresponding public information to the blockchain

network in the form of a transaction. The transaction is finally added to the blockchain. After that, the user in the system can query other users' public information through the blockchain and verify the user's identity. After the authentication is completed, the users can negotiate a session key through shared secret parameters to ensure the privacy of subsequent session information.

4.2. Details of the Proposed Scheme. Next, we divided the proposed system into three phases which are described in detail, namely, initialization phase, registration phase, and authentication phase. The overall process of authentication is shown in Figure 2.

4.2.1. Phase 1: Initialization. The initialization phase can be divided into two parts. One part is the authorities and the blockchain network initialization. The other part is the user initialization. Initially, the users and the authorities initialize the system, and the system constructs a permissioned blockchain network, where users (DO and DU) are the participant and the miners are the maintainer of the blockchain. The users write transactions in order to provide identity authentication function. The miners verify the transactions in order to provide valid information for identity authentication. Specifically, the users and the authorities establish a consortium blockchain, and the miners who maintain the blockchain network rely on a practical Byzantine fault tolerance (PBFT) consensus mechanism. They execute the following operations to initialize a series of system parameters:

- (1) For two large primes p, q and an elliptic curve E_p , there is a nonregular elliptic curve additive cyclic group G of order q and a generator P of G . Choose SHA256 as the encryption hash function H , elliptic curve digital signature algorithm (ECDSA) as the signature algorithm Sig , and RSA encryption algorithm as the asymmetric encryption algorithm Enc .
- (2) The identity of the authority is marked as $\text{AuthorityID}(A_i)$. The authority A_i generates public and private key pairs $(\text{PK}_{A_i}, \text{SK}_{A_i})$. Then, A_i publishes PK_{A_i} to the users in the system and the miner in the blockchain network.
- (3) The identity of the user is marked as $\text{UserID}(U_i)$. The user U_i generates its own pseudo-identity PU_i and calculates PK_{PU_i} by choosing a secret key SK_{PU_i} . Then, PU_i pushes PK_{PU_i} to the other users in the system and the miners in the blockchain network.
- (4) The users write a smart contract (SC) in order to provide the registration function, in which public and private key pairs are PK_{SC} and SK_{BC} .
- (5) The public parameters can be represented as $(G, g, H, \text{Sig}, \text{Enc}, \text{PK}_{A_i}, \text{PK}_{\text{PU}_i}, \text{PK}_{\text{BC}})$.

4.2.2. Phase 2: Registration

- (1) The user U_i sends $(\text{PU}_i, \text{PK}_{\text{PU}_i})$ to A_i through a secure channel.

- (2) Upon receiving the user's message, A_i firstly verifies $(\text{PU}_i, \text{PK}_{\text{PU}_i})$. If PU_i has already been registered or it is invalid, PU_i rejects the request. Otherwise, A_i generates a verifiable proof $\mathcal{PF}_{A_i, \text{PU}_i}$ and its signature

$\sigma_{A_i, \text{PU}_i} = \text{Sig}(\text{SK}_{A_i}, (H(\text{PU}_i), \text{PK}_{\text{PU}_i}, \mathcal{PF}_{A_i, \text{PU}_i}, \text{VT}))$ for PU_i , where VT is the valid time of the proof. Then, A_i sends $(\text{PU}_i, \text{PK}_{\text{PU}_i}, \mathcal{PF}_{A_i, \text{PU}_i}, \text{VT}, \sigma_{A_i, \text{PU}_i})$ to the user U_i through a secure channel.

- (3) Upon receiving $\sigma_{A_i, \text{PU}_i}$ from A_i , the user U_i sends the public information to the blockchain. Firstly, U_i generates a timestamp ts_r and then computes $\text{CP} = \text{Enc}(\text{PK}_{\text{BC}}, (\mathcal{PF}, \text{VT}))$ and signature $\sigma_{\text{FU}_i, R} = \text{Sig}(\text{SK}_{\text{PU}_i}, (H(\text{ts}_r), \mathcal{PF}_{A_i, \text{PU}_i}, \text{VT}, \sigma_{A_i, \text{PU}_i}))$. Finally, the user sends $(\text{PU}_i, \text{PK}_{\text{PU}_i}, \text{ts}_r, \text{CP}, \sigma_{A_i, \text{PU}_i}, \sigma_{\text{FU}_i, R})$ to the blockchain network.
- (4) Upon receiving the message from PU_i , the miner verifies whether the timestamp ts_r is within the allowed range compared to the current time. If not, miner rejects the transaction; otherwise, miner continues to check whether the lifetime VT is within the allowed time. If not, miner stops the session. Otherwise, miner decrypts CP to get the proof and verifies the signature $\sigma_{A_i, \text{PU}_i}$. If the signature is valid, miner writes this transaction to the blockchain. The user can generate several PU_i and corresponding public and private keys to obtain verifiable proofs of different authorities and store them locally.

4.2.3. Phase 3: Authentication. After a user's PU_i and public key are added to the blockchain, the detailed authentication process is as follows:

- (1) User U_i with identity PU_i first generates random value r and timestamp ts_a and computes signature $\sigma_{\text{PU}_i, A} = \text{Sig}(\text{SK}_{\text{PU}_i}, H(\text{PU}_i, r, \text{ts}_a))$. Then, U_i sends $(\text{PU}_i, r, \text{ts}_a, \sigma_{\text{PU}_i, A})$ to user U_j .
- (2) Upon receiving the message from U_i , U_j first verifies the timestamp and the signature. If ts_a is not within the allowed range compared to the current time or the signature is invalid, U_j rejects the access request; otherwise, U_j searches for PK_{PU_i} on the blockchain with PU_i . If there is no PK_{PU_i} , U_j rejects the access request. Otherwise, U_j verifies the signature $\sigma_{\text{PU}_i, A}$ with PK_{PU_i} . If $\sigma_{\text{PU}_i, A}$ is invalid, U_j stops the session; otherwise, the user's identity is verified.

5. Security and Performance Analysis

5.1. Security Analysis. In this section, we first compare the proposed scheme with four other representative authentication schemes in terms of authentication, privacy preservation, scalability, and centralized trusted authority. Then, we introduce the security requirements and give the corresponding analysis.

The security requirements mainly include integrity, availability, scalability, nonrepudiation, identity authentication, and communication security. In addition, we

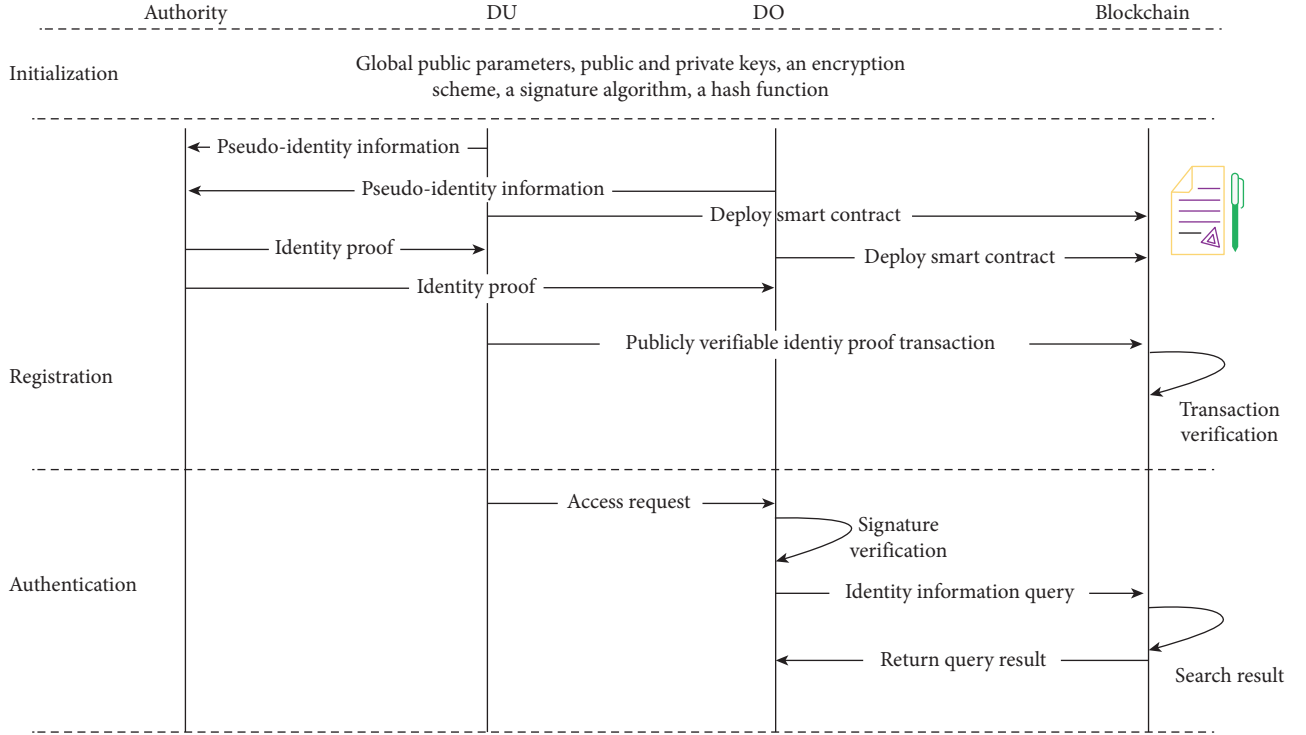


FIGURE 2: The process of authentication.

compared the solution with the existing blockchain-based solutions in a comprehensive function. The comparison results are shown in Table 1. It can be seen from the table that our scheme not only supports identity anonymity, authentication, nonrepudiation, scalability, and decentralized functions but also has more advantages in privacy protection and communication security. In particular, the proposed scheme does not need to wait for the block confirmation and cross-blockchain operations during the authentication process.

TABLE 1: Security features' comparison.

Features	[34]	[35]	[36]	[13]	[21]	Our
Identity anonymity	√	√	√	√	√	√
Authentication	√	√	√	√	√	√
Nonrepudiation	√	√	√	√	√	√
Privacy preservation	√	√	×	×	×	√
Scalability	—	—	√	√	√	√
Decentralized	—	—	√	√	√	√
Cross-blockchain	—	—	×	√	×	×
Blockchain confirmation	—	—	√	√	√	×

- (i) Identity anonymity: identity anonymity means that other users cannot obtain the user's true information through the user's access request. In the proposed scheme, the user completes the identity registration by generating the identity information PU_i independently and uploads the corresponding valid proof to the blockchain network. First, the user can have multiple PU_i information independent of the real identity, and the PU_i information is also independent of each other. Secondly, in the process of performing authentication, users also use PU_i information independent of identity information, so the validity of user identity can be guaranteed.
- (ii) Authentication: authentication means that two users need to be identified before they interact. The authentication scheme proposed in this paper is the identity information generated by the user independently, and the registration of the identity and the disclosure of the effective proof are completed by the blockchain network, that is, the

effective proof is stored in the blockchain network. The authenticating party can identify the authenticated party by accessing the blockchain and realize identity authentication.

- (iii) Integrity: the security requirements for integrity mainly include two aspects: data integrity and message integrity. Data integrity means that unauthorized users and devices cannot access and modify the data. Message integrity means that the message sent by the user and the device cannot be tampered with illegally during the interaction. The authentication process of this scheme is realized with the help of the blockchain. The core of the verification is that the user transmits the valid identity certificate to the blockchain network and stores it in the form of a transaction. In the blockchain network, every transaction will be verified by miners, so the integrity of the message can be guaranteed. In the proposed scheme, the user's data are stored on the blockchain network.

Once the verified data are stored, it will be difficult to be tampered with, so the data integrity can be effectively guaranteed.

- (iv) **Nonrepudiation:** nonrepudiation means that users and devices cannot reject the operations they have implemented and the messages they send. Since the scheme is carried out through the blockchain, all operations are stored in the blockchain in the form of transaction records, and all access requests and transactions are signed; therefore, the scheme is undeniable.
- (v) **Scalability:** scalability is one of the important security requirements of blockchain identity authentication. Due to the time delay characteristics of the blockchain, if users frequently complete identity authentication through transactions, it will consume a lot of resources and time. In the scheme designed in this paper, users only need to complete the corresponding proof data on the blockchain during the registration phase. In the identity authentication phase, there is no need to wait for block confirmation, and there is only a need to search the data on the blockchain to complete the identity authentication. For scalability requirements, this solution can be well adapted.
- (vi) **Privacy preservation:** privacy protection mainly refers to the privacy and security of the user data and identity in the storage process. In the schemes in [13, 21, 36], the authority can know the identity of the user during the registration phase, and then the authority will store the information on the blockchain. In addition, the identity identifier used in the above solution is the unique identity of the device/user. This results in that the user's identity information is stored in the blockchain in the plaintext, which will result in the user's identity information not being protected during the communication process, and it also faces the security risks of the storage process. Different from using unique identities to achieve authentication, users in our solution can create multiple independent identities according to their needs. Although the authority can still know the user's identity, the user can hash the identity information and independently decide whether to store the information on the blockchain. In addition, when storing, the message is encrypted by an encryption algorithm, so the proposed scheme has more advantages in privacy protection. Therefore, the proposed scheme can more comprehensively realize privacy protection.
- (vii) **Communication security:** communication security refers to the security of the user's communication data during the identity authentication process. In the scheme proposed in [13, 36], the certification information used for authentication not only contains the unique identities of the authenticated parties but is also transmitted in the blockchain network in the form of a plaintext. In the scheme proposed in [21], the security of communication is

achieved by establishing a blockchain-level bubble, which can be seen as establishing a safe environmental space. Different from the method in the above schemes, the communication security in the proposed scheme is realized by cryptography methods. In the registration stage, the user uses public key information to register, and then when transmitting the identity certificate, the method of symmetric data encryption is used to ensure the security of data transmission. In the authentication process, on the one hand, the user does not need to send a complete certificate. On the other hand, the identity identifier used in the authentication is not unique. Therefore, the proposed scheme has obvious advantages in communication security.

- (viii) **Cross-blockchain:** cross-blockchain authentication refers to whether a hybrid blockchain combining a private blockchain and a public blockchain is used in the process of implementing the authentication scheme. For different blockchains, each individual blockchain network is a relatively independent network. The block structure and the deployment of the consensus mechanism may be different, data information is difficult to interconnect and synchronize, and there is a problem of information islands. This makes it difficult to collaborate between different blockchain networks and greatly limits the development of blockchain applications. Therefore, avoiding the use of hybrid blockchains to complete identity verification and avoiding cross-domain identity verification are also issues that need to be considered. Different from the cross-blockchain identity authentication scheme designed in scheme [13], the proposed scheme in this article, only a single blockchain is used to record the credential information, thereby avoiding the security risks caused by cross-chain authentication.
- (ix) **Block confirmation:** block confirmation refers to whether it is necessary to wait for a transaction during the identity authentication process. In the scheme proposed in [13, 36], the identity authentication process needs to invoke the smart contract in the blockchain, so it needs to wait for the execution of the smart contract and the confirmation of the relevant block, but in this proposed scheme, the verifier only needs to search the blockchain once to complete the identity verification without waiting for the confirmation of the transaction block. In terms of time cost, the authentication time of using smart contracts depends on the time to reach consensus in the blockchain. In the proposed scheme, the authentication time mainly depends on the search time for related information.

5.2. Performance Evaluation. In this section, we conduct experiments to evaluate the effectiveness and feasibility of our scheme. We employ the related cryptographic opera-

TABLE 2: Parameter definitions.

Symbol	Description	Size
G	Bit length of an element in G	512
PU_i	Bit length of an identity	256
ts	Bit length of a timestamp	32
r	Bit length of a random number	256
h	Bit length of a hash function	256
σ	Bit length of a signature	1024
\mathcal{PF}	Bit length of a proof	1024

tions in the C/C++ OPENSLL library [37]; the parameters used are shown in Table 2.

The complex calculations and large-capacity storage required in the authentication process are placed on the blockchain. In order to realize user identity authentication based on the blockchain, a valid and public identity proof is stored on the blockchain. In this part, we mainly analyze the performance of the registration process and the identity authentication process. Since there are few existing blockchain-based identity authentication schemes, starting from the core idea of the scheme, the feasibility of the scheme is analyzed by analyzing the calculation cost, the communication cost, and the storage cost of each process in the scheme.

In the registration phase, the user first sends a request message to the authority. After receiving the proof returned by the authority, the user sends a registration transaction to the blockchain network. From the user's point of view, it is necessary to execute the signature generation algorithm twice, the verification algorithm once, and the encryption algorithm once. Besides, the user needs to store the proof returned by A_i and the pseudo-identity information (PU_i) generated by himself. From the authority perspective, the signature generation algorithm needs to be executed once. For miners on the blockchain, it is necessary to execute the signature verification algorithm twice and the decryption algorithm once.

In the identity authentication phase, the user U_i first sends a request to the verifier U_j . After the verifier U_j completes the message integrity check, it visits the blockchain network and completes the identity authentication by querying whether valid identity information exists. It needs to be pointed out that, at this stage, the verifier U_j does not need to store any U_i 's information, which reduces a lot of storage overhead for the verifier. Therefore, at this stage, the access requester U_i needs to execute a signature algorithm; the verifier U_j needs to execute a verification algorithm and a blockchain search request. The search process here can be implemented by miners or corresponding smart contracts.

In order to show the performance of the solution more intuitively, the communication cost and the calculation cost at different phases are shown in Figures 3 and 4, respectively. Through the above analysis, this solution meets expectations and is feasible in terms of computing and storage overhead.

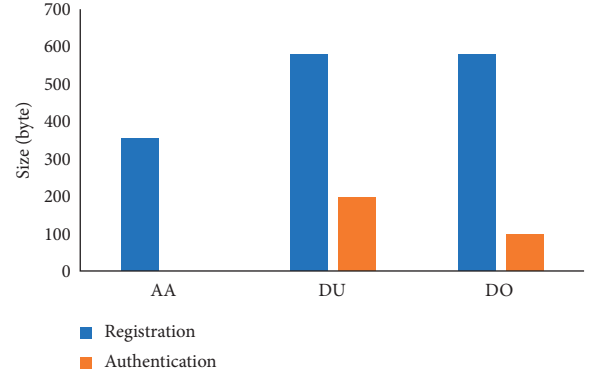


FIGURE 3: Communication cost.

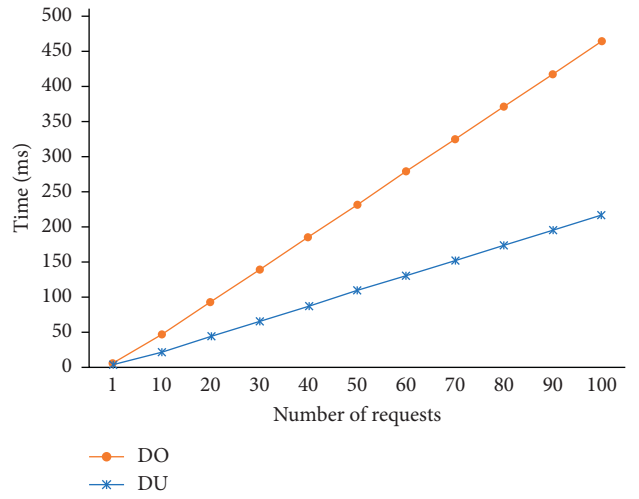


FIGURE 4: Calculation cost.

6. Conclusion

In this article, an identity authentication scheme based on blockchain-based privacy protection is proposed. The user generates identity information independently and completes the registration of identity certification through the blockchain. On the one hand, it can realize the protection of real identity information; on the other hand, it can avoid the storage overhead caused by the need to store a large number of certificates or key pairs. Due to the use of blockchain, there is no single point of failure in the authentication process, and it can be applied to distributed scenarios. Finally, the security analysis and performance evaluation demonstrate that the proposed scheme can meet the security requirements and is feasible.

Data Availability

The parameter 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 Key R&D Program of China (Grant no. 2017YFB1400700), the National Natural Science Foundation of China (Grant no. 62072487), the Natural Science Foundation of Beijing (Grant no. M21036), and the National Statistical Science Foundation of China (Grant no. 2020LD01).

References

- [1] N. Kumar, D. Acharya, and D. Lohani, "An IoT-based vehicle accident detection and classification system using sensor fusion," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 869–880, 2020.
- [2] Y. Zhao, J. Zhao, L. Jiang et al., "Privacy-preserving blockchain-based federated learning for iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1817–1829, 2020.
- [3] Y. Liu, X. Ma, L. Shu et al., "Internet of Things for noise mapping in smart cities: state of the art and future directions," *IEEE Network*, vol. 34, no. 4, pp. 112–118, 2020.
- [4] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
- [5] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2019.
- [6] F. Alshehri and G. Muhammad, "A comprehensive survey of the Internet of Things (IoT) and ai-based smart healthcare," *IEEE Access*, vol. 9, pp. 3660–3678, 2020.
- [7] K. Monteiro, E. Rocha, E. Silva, G. L. Santos, W. Santos, and P. T. Endo, "Developing an e-health system based on IoT, fog and cloud computing," in *Proceedings of the 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, pp. 17–18, Zurich, Switzerland, December 2018.
- [8] M. Elhoseny, G. Ramirez-González, O. M. Abu-Elnasr, S. A. Shawkat, N. Arunkumar, and A. Farouk, "Secure medical data transmission model for IoT-based healthcare systems," *IEEE Access*, vol. 6, pp. 20596–20608, 2018.
- [9] M. B. Mollah, J. Zhao, D. Niyato et al., "Blockchain for future smart grid: a comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, pp. 1–26, 2020.
- [10] Z. Chen, Y. Tian, and C. Peng, "An incentive-compatible rational secret sharing scheme using blockchain and smart contract," *Science China Information Sciences*, vol. 64, no. 10, pp. 1–21, 2021.
- [11] J. Wang, M. Li, Y. He, H. Li, K. Xiao, and C. Wang, "A blockchain based privacy-preserving incentive mechanism in crowdsensing applications," *IEEE Access*, vol. 6, pp. 17 545–17 556, 2018.
- [12] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
- [13] 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, pp. 241–251, 2020.
- [14] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.
- [15] J. Xu, K. Xue, H. Tian, J. Hong, D. S. L. Wei, and P. Hong, "An identity management and authentication scheme based on redactable blockchain for mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6688–6698, 2020.
- [16] Z. Lu, Q. Wang, G. Qu, H. Zhang, and Z. Liu, "A blockchain-based privacy-preserving authentication scheme for VANETs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 12, pp. 2792–2801, 2019.
- [17] F. Wu, X. Li, A. Sangaiah et al., "A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks," *Future Generation Computer Systems*, vol. 82, p. 9, 2017.
- [18] Z. Liu, Z. Liu, L. Zhang, and X. Lin, "MARF: a distributed mac layer attack resistant pseudonym scheme for VANET," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 4, pp. 869–882, 2020.
- [19] U. Premarathne, I. Khalil, Z. Tari, and A. Zomaya, "Cloud-based utility service framework for trust negotiations using federated identity management," *IEEE Transactions on Cloud Computing*, vol. 5, no. 2, pp. 290–302, 2015.
- [20] G. Bendiab, S. Shiaeles, S. Boucherkha, and B. Ghita, "FCMDT: a novel fuzzy cognitive maps dynamic trust model for cloud federated identity management," *Computers and Security*, vol. 86, pp. 270–290, 2019.
- [21] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of trust: a decentralized blockchain-based authentication system for IoT," *Computers and Security*, vol. 78, pp. 126–142, 2018.
- [22] Q. Lai, L. Xu, M. Yuan, F. Wang, and H. Fang, "User privacy-preserving scheme based on anonymous authentication in smart grid," in *Proceedings of the International Conference on Security and Privacy in Digital Economy*, pp. 676–691, Quzhou, China, October 2020.
- [23] A. K. Das and A. Goswami, "A secure and efficient uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care," *Journal of Medical Systems*, vol. 37, no. 3, p. 9948, 2013.
- [24] C. B. Avoussoukpo, C. Xu, and M. Tchenagnon, "Ensuring users privacy and mutual authentication in opportunistic networks: a survey," *International Journal of Network Security*, vol. 22, pp. 118–125, 2019.
- [25] M. Wagner and B. McMillin, "An efficient blockchain authentication scheme for vehicular ad-hoc networks," in *Proceedings of the International Conference on Critical Infrastructure Protection*, pp. 87–109, Arlington National, VA, USA, 2020.
- [26] A. Mohsin, A. Zaidan, B. Bahaa et al., "Blockchain authentication of network applications: taxonomy, classification, capabilities, open challenges, motivations, recommendations and future directions," *Computer Standards and Interfaces*, vol. 64, pp. 41–60, 2019.
- [27] W. A. Ali, N. M. Sahib, and J. Waleed, "Preservation authentication and authorization on blockchain," in *Proceedings of the 2019 2nd International Conference on Engineering Technology and its Applications (IICETA)*, pp. 83–88, Al-Najef, Iraq, August 2019.

- [28] F. Tang, S. Ma, Y. Xiang, and C. Lin, "An efficient authentication scheme for blockchain-based electronic health records," *IEEE Access*, vol. 7, pp. 41 678–41 689, 2019.
- [29] X. Xiang, M. Wang, and W. Fan, "A permissioned blockchain-based identity management and user authentication scheme for e-health systems," *IEEE Access*, vol. 8, pp. 171 771–171 783, 2020.
- [30] 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.
- [31] Z. Wu, R. Liu, and H. Cao, "ECDSA-based message authentication scheme for BeiDou-II navigation satellite system," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 4, pp. 1666–1682, 2018.
- [32] K. Balasubramanian, "Variants of RSA and their cryptanalysis," in *Proceedings of the 2014 International Conference on Communication and Network Technologies*, pp. 145–149, Hefei, China, July 2014.
- [33] F. Mallouli, A. Hellal, N. Sharief Saeed, and F. Abdurraheem Alzahrani, "A survey on cryptography: comparative study between RSA vs ECC Algorithms, and RSA vs El-Gamal algorithms," in *Proceedings of the 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pp. 173–176, Paris, France, 2019.
- [34] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.
- [35] C. Chang and H. Tsai, "An anonymous and self-verified mobile authentication with authenticated key agreement for large-scale wireless networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3346–3353, 2010.
- [36] 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 2018.
- [37] M. I. Mihailescu and S. L. Nita, *Cryptography Libraries in C/C++20*, Apress, Berkeley, CA, USA, 2021.

Research Article

Privacy-Preserving Attribute-Based Keyword Search with Traceability and Revocation for Cloud-Assisted IoT

Kai Zhang , Yanping Li , and Laifeng Lu 

School of Mathematics and Statistics, Shaanxi Normal University, Xi'an 710119, China

Correspondence should be addressed to Laifeng Lu; lulaifeng@snnu.edu.cn

Received 19 March 2021; Accepted 23 May 2021; Published 30 May 2021

Academic Editor: Qing Yang

Copyright © 2021 Kai Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of cloud computing and Internet of Things (IoT) technology, it is becoming increasingly popular for source-limited devices to outsource the massive IoT data to the cloud. How to protect data security and user privacy is an important challenge in the cloud-assisted IoT environment. Attribute-based keyword search (ABKS) has been regarded as a promising solution to ensure data confidentiality and fine-grained search control for cloud-assisted IoT. However, due to the fact that multiple users may have the same retrieval permission in ABKS, malicious users may sell their private keys on the Internet without fear of being caught. In addition, most of existing ABKS schemes do not protect the access policy which may contain privacy information. Towards this end, we present a privacy-preserving ABKS that simultaneously supports policy hiding, malicious user traceability, and revocation. Formal security analysis shows that our scheme can not only guarantee the confidentiality of keywords and access policies but also realize the traceability of malicious users. Furthermore, we provide another more efficient construction for public tracing.

1. Introduction

As a prevalent Internet technology, Internet of Things (IoT) [1] has been widely used in various industries, such as smart healthcare, transportation, and city [2–5]. Due to the limited computing and storage capacity of many IoT devices, users often need to store IoT data in the cloud. The cloud-assisted IoT [6] technology can be used to collect and store massive medical data, so it is expected to greatly improve the efficiency of medical institutions and promote the development of smart healthcare. Apart from the efficiency concern, security issue is an important concern hindering the widespread application of IoT technology [7–10]. Especially for the smart healthcare system based on cloud-assisted IoT, the data security issue has become a key challenge, due to the fact that the sensitive personal health record (PHR) outsourced in the cloud is vulnerable to hacker attacks.

Although the traditional encryption technology [11] can protect the data security, it makes the ciphertext data unable to retrieve, thus greatly reducing the availability of IoT data. An inefficient solution is that data users download ciphertext

data from the cloud, decrypt it, and then search on plaintext data. However, ordinary users do not have enough storage and computing power to retrieve the huge amount of cloud data locally. Public key encryption with keyword search (PEKS) [12, 13] is a more efficient solution, which can realize the retrieval of ciphertext by a cloud server without decryption. In a PEKS scheme, a data user can delegate the cloud server to retrieve all cloud ciphertexts by sending a search token to it. However, in order to avoid the abuse of retrieval ability, data owners usually want to control the retrieval permission.

As an efficient and flexible solution to meet the above requirements, attribute-based keyword search (ABKS) [14, 15] can realize data confidentiality, ciphertext retrieval, and fine-grained access control simultaneously. In a ciphertext-policy ABKS (CP-ABKS) system, a data owner encrypts the file keyword by an access policy and only users whose attributes satisfy the access policy can retrieve the ciphertext file. However, the public access policy in CP-ABKS may disclose privacy information in the smart medical cloud system. For example, a medical institution

wants to share PHR with users whose attributes meet the policy “(Institution: hospital A AND Patient ID: 202007953) OR (Institution: hospital B AND Position: oncologist)”; then it encrypts the PHR keyword by this policy and generates the corresponding ciphertext. Note that the access policy is exposed together with the ciphertext in the traditional CP-ABKS, so anyone can infer that the patient with the identity “202007953” is likely to have a tumor. Moreover, multiple users with the same attributes have the same retrieval ability and the user identity cannot be determined by the user private key in CP-ABKS, so malicious users may sell their private keys without worrying about being caught. As in the above example, if one of the multiple oncologists in hospital B sells his private key online, it is difficult to accurately identify and revoke the malicious user who sells his private key.

1.1. Our Contributions. Up till now, there is no secure ABKS scheme that simultaneously supports hidden policy, traceability, and revocation. To address these issues, we propose a traceable and revocable hidden ABKS (TR-HABKS) scheme and an enhanced TR-HABKS (eTR-HABKS) scheme, which support the above three properties at once. Moreover, the eTR-HABKS scheme achieves two other remarkable properties: (1) no identity table for tracing: the scheme only needs to maintain an identity table for revocation but does not require any identity table for tracing; (2) public traceability: besides the trusted authority, anyone without additional secret information can also run the tracing algorithm to capture malicious users. Specifically, our TR-HABKS and eTR-HABKS schemes provide the following properties:

- (i) *Fine-Grained Search Control.* In our schemes, a data user’s search token is corresponding to his attributes and can be used to retrieve ciphertext only when the attributes satisfy the ciphertext policy. To control the user search permission, our schemes allow the data owner to encrypt the keyword by a specified access policy, which can be expressed as an AND-gates on multivalued attributes.
- (ii) *Hidden Policy.* Our schemes not only guarantee the confidentiality of the keyword but also protect the privacy of the policy. Different from those ABKS schemes which only prove the keyword security, we also prove that access policies are also indistinguishable in the selective security model. Moreover, our schemes require the data owner to encrypt the keyword by his private key, so that the adversary cannot launch the keyword guessing attacks (KGA) by generating the ciphertext himself.
- (iii) *Traceability.* Both the TR-HABKS and eTR-HABKS schemes achieve the user traceability in ABKS. When a malicious user leaks his private key in our TR-HABKS scheme, then the trusted authority can determine the identity of the malicious user by a tracing identity table. In our eTR-HABKS scheme, everyone can trace the malicious user’s identity without the help of any identity table.

- (iv) *Revocation.* When the malicious user’s identity is determined, our schemes can effectively revoke the user by managing a registration table. In our schemes, the trusted authority adds each legitimate user to a registered identity table in the key generation stage and can easily revoke the malicious user by deleting his identity from the identity table.

The properties comparison between our schemes with other related works can be seen in Table 1. The symbol “—” means not applicable.

1.2. Related Work

1.2.1. Attributed-Based Encryption. Attribute-based encryption (ABE) [19] is a practical method for fine-grained access control and can be divided into key-policy ABE (KP-ABE) [20] and ciphertext-policy ABE (CP-ABE) [21, 22]. Based on KP-ABE and CP-ABE, dual-policy ABE (DP-ABE) [23, 24] was also introduced for achieving content-based and role-based access control simultaneously. However, in traditional CP-ABE and DP-ABE, the access policy corresponding to the ciphertext may disclose the user’s privacy. To address this problem, Nishide et al. [25] proposed the first CP-ABE in which access policies can be hidden by the encryptor. Later, Lai et al. [26] presented a high expressive CP-ABE with partially hidden access structure that can be expressed as a linear secret-sharing scheme (LSSS) [27]. Yang et al. [28] proposed a privacy-preserving CP-ABE to hide both the attribute names and the attribute values in the access policy. Based on an optimized vector transformation approach, Sun et al. [29] proposed a lightweight hiding CP-ABE scheme for IoT-oriented smart health. Their scheme can not only support policy hiding but also support offline encryption and outsourcing decryption. In order to prevent key abuse, Hinek et al. [30] first considered the trace problem in ABE and constructed a traceable ABE scheme. Liu et al. [31] proposed a high expressive white-box traceable ABE that supports traceability of the malicious user who sold his decryption key on the Internet. For a decryption black-box in ABE, Liu et al. [32] later proposed a black-box traceable CP-ABE that can trace the malicious user whose private key was used to construct the decryption device. To support more flexible attributes, Ning et al. [33] presented a traceable CP-ABE that simultaneously supports white-box traceability and large universe. Ying et al. [34] presented a black-box traceable CP-ABE with hidden policy in e-healthcare cloud. Recently, several novel ABE schemes [35–37] were proposed for stronger security and user revocation in cloud storage system. Unfortunately, the above ABE schemes cannot search the ciphertext data in the cloud.

1.2.2. Attribute-Based Keyword Search. Boneh et al. [12] first introduced the concept of PEKS and constructed the first concrete PEKS scheme. In the scheme, the user authorizes a third party to search the ciphertext by giving him a search token that is associated with a keyword; the third party

TABLE 1: Properties comparison.

Scheme	[12]	[16]	[14]	[15]	[17]	[18]	TR-HABKS	eTR-HABKS
Search	✓	✓	✓	✓	✓	✓	✓	✓
Fine-grained search control	×	×	✓	✓	✓	✓	✓	✓
Hidden policy	—	—	×	×	×	✓	✓	✓
Resist KGA	×	✓	×	×	×	✓	✓	✓
Traceability	×	×	×	×	×	×	Private traceability	Public traceability
Revocation	×	×	×	✓	✓	×	✓	✓

returns the search results to the user but without learning the keyword information. However, Byun et al. [38] pointed out that the PEKS scheme [12] cannot resist KGA. Specifically, anyone can generate a ciphertext by encrypting a keyword in PEKS scheme, so the third party can use the search token to continuously retrieve the ciphertexts corresponding to different keywords to guess the keyword corresponding to the search token. To resist the above attack in PEKS, Huang and Li [16] presented a public key authenticated encryption with keyword search, in which the keyword needs to be authenticated by the data owner during the encryption phase. Miao et al. [39] proposed a verifiable searchable encryption, which can achieve verifiable searchability and resist KGA. In order to support fine-grained search authorization, Zheng et al. [14] proposed a CP-ABKS scheme based on PEKS and CP-ABE [21]. In the CP-ABKS scheme, a data owner encrypts a keyword by an access policy and only users whose attributes meet the access policy can retrieve the ciphertext. With the help of proxy reencryption and lazy reencryption techniques, Sun et al. [15] presented a revocable ABKS scheme that can delegate the search and update workload to the cloud server. Liu et al. [17] proposed a searchable ABE with efficient revocation and outsourced decryption for cloud IoT. Based on online/offline encryption and outsourced decryption techniques, Miao et al. [40] presented an efficient ABKS scheme in the cloud-assisted healthcare IoT system. To protect access policies, Qiu et al. [18] presented a hidden policy CP-ABKS against KGA. Later, Miao et al. [41] presented a privacy-preserving CP-ABKS in multiowner setting. However, Sun et al. [42] pointed out that four types of KGA exist in this scheme. To achieve hidden policy and traceability simultaneously, Liu et al. [43] presented a privacy-preserving ABKS with user tracing. However, the security proof cannot ensure the policy hiding property due to its flawed security model. Unlike with the formal security model in hidden policy CP-ABKS [18, 41], the security model in [43] only shows the indistinguishability of keywords and does not consider the indistinguishability of access policies.

1.3. Organization. The rest of this paper is organized as follows. Section 2 introduces the necessary background information of the paper. Section 3 defines the algorithm and model for TR-HABKS. Section 4 presents the TR-HABKS construction and proves its correctness and security. Section 5 presents the eTR-HABKS construction and compares the efficiencies of the TR-HABKS and eTR-HABKS schemes. Section 6 concludes the paper.

2. Background

For a set S , let $s \leftarrow_R S$ denote that an element s is chosen uniformly at random from S . Let \mathbb{Z}_p denote the set $\{0, 1, 2, \dots, p-1\}$, where p is a prime, let $[n]$ denote the set $\{1, 2, \dots, n\}$, where n is a natural number, and let PPT denote probabilistic polynomial time.

2.1. Access Policy. In our system, the total number of attributes is n , and the access policy is represented by an AND-gates on multivalued attributes [25]. For each $i \in [n]$, let A_i be the attribute index, and let $S_i = \{v_{i,t}\}_{t \in \mathcal{N}_i}$ be the possible values of A_i , where n_i is the number of possible values for A_i . Let $L = \{L_i\}$ be a user attributes set, where $L_i \in S_i$, and let $P = \{P_i\}_{i \in [n]}$ be an access policy, where $P_i \subseteq S_i$. If $L_i \in P_i$ for $i \in [n]$, we say that the attributes set L satisfies the access policy P , written as $L \models P$; otherwise, we say that the attributes set L does not satisfy the access policy P , written as $L \not\models P$. For ease of description, we use i instead of A_i to represent attribute index in our schemes.

2.2. Bilinear Map. An asymmetric bilinear group generator \mathcal{G} takes as input a security parameter λ and outputs a tuple $\mathbb{G} = (p, G_1, G_2, G_T, g_1, g_2, e)$, where p is a prime, G_1, G_2 , and G_T are multiplicative cyclic groups of order p , g_1 (resp., g_2) is a generator of G_1 (resp., G_2), and $e: G_1 \times G_2 \rightarrow G_T$ is an efficiently computable bilinear map with the following properties:

- (1) Bilinear: $\forall g \in G_1, h \in G_2, a, b \in \mathbb{Z}_p, e(g^a, h^b) = e(g, h)^{ab}$
- (2) Nondegenerate: $e(g_1, g_2) \neq 1$

2.3. Signature. A signature scheme consists of the following algorithms:

$(PK, SK) \leftarrow \text{KeyGen}(\lambda)$: The key generation algorithm gets the security parameter λ as input. It outputs a random key pair (PK, SK) .

$(\sigma) \leftarrow \text{Sign}(SK, M)$: The signing algorithm gets a private key SK and a message M as input. It outputs a signature σ .

$(0/1) \leftarrow \text{Verify}(PK, M, \sigma)$: The verifying algorithm gets a public key PK , a message M , and a signature σ as input. It outputs 1 if the signature is valid, and outputs 0 otherwise.

The existential unforgeability under a weak chosen message attack [44] is defined by the following game:

Query: the adversary sends messages $\{M_j\}_{j \in [q_s]}$ to the challenger, where q_s is the maximum number of signatures that the adversary can query.

Response: the challenger runs the key generation algorithm and generates the signatures $\{\sigma_j\}_{j \in [q_s]}$ on the messages $\{M_j\}_{j \in [q_s]}$. Then, the challenger gives the public key PK and the signatures $\{\sigma_j\}_{j \in [q_s]}$ to the adversary.

Output: the adversary outputs a pair (M, σ) .

The adversary wins this game if $\text{verify}(\text{PK}, M, \sigma) = 1$ and $(M, \sigma) \notin \{(M_j, \sigma_j)\}_{j \in [q_s]}$. The adversary's advantage is defined as the probability that he wins this game.

Definition 1. A signature scheme is said to be existentially unforgeable under a weak chosen message attack if all PPT adversaries have only a negligible advantage in this game.

3. Problem Formulation

In this section, we describe the algorithm definition, system model, and security model of TR-HABKS.

3.1. Algorithm Definition. A TR-HABKS scheme is formally defined as follows:

$(\text{MK}, \text{PK}) \leftarrow \text{Setup}(\lambda)$: the setup algorithm gets the security parameter as input. It outputs the master key MK and the public parameter PK. In addition, it also generates two empty identity tables T_1 and T_2 .

$(\text{SK}_{\text{id},L}, \text{SK}_o) \leftarrow \text{KeyGen}(\text{MK}, \text{PK}, \text{id}, L)$: the key generation algorithm gets an attributes set L , an identity id, the master key MK, and the public parameter PK as input. It outputs a private key $\text{SK}_{\text{id},L}$ for a data user and a private key SK_o for the data owner. In addition, it adds id to T_1 and T_2 .

$\text{CT} \leftarrow \text{Enc}(\omega, P, \text{SK}_o, \text{PK})$: the encryption algorithm gets a keyword ω , an access policy P , the data owner's private key SK_o , and the public parameter PK as input. It outputs a ciphertext CT.

$\text{TK}_{\text{id},L} \leftarrow \text{TokenGen}(\text{SK}_{\text{id},L}, \text{PK}, \omega')$: the token generation algorithm gets a data user's private key $\text{SK}_{\text{id},L}$, the public parameter PK, and a keyword ω' as input. It outputs a search token $\text{TK}_{\text{id},L}$.

$(0/1) \leftarrow \text{Search}(\text{TK}_{\text{id},L}, \text{CT}, T_1)$: the searching algorithm gets a ciphertext CT, a search token $\text{TK}_{\text{id},L}$, and an identity table T_1 as input. It outputs 1 if (1) $L \neq P$, (2) $\text{id} \in T_1$, and (3) $\omega = \omega'$ and outputs 0 otherwise.

$(\text{id}/\top) \leftarrow \text{Trace}(\text{SK}_{\text{id},L}, \text{PK}, T_2)$: the tracing algorithm gets a secret key $\text{SK}_{\text{id},L}$, the public parameter PK, and an identity table T_2 as input. It outputs a user identity id if $\text{SK}_{\text{id},L}$ passes the key sanity check and outputs symbol \top otherwise. Key sanity check is a deterministic algorithm to test whether $\text{SK}_{\text{id},L}$ needs to be traced.

$T_1 \leftarrow \text{Revoke}(\text{id}, T_1)$: the revocation algorithm gets a revocation user identity id and an identity table T_1 as input. It outputs an updated table T_1 .

3.1.1. Correctness. A TR-HABKS scheme is correct if the following condition holds: Given $(\text{MK}, \text{PK}) \leftarrow \text{Setup}(\lambda)$, $(\text{SK}_{\text{id},L}, \text{SK}_o) \leftarrow \text{KeyGen}(\text{MK}, \text{PK}, \text{id}, L)$, $\text{CT} \leftarrow \text{Enc}(\omega, P, \text{SK}_o, \text{PK})$, $\text{TK}_{\text{id},L} \leftarrow \text{TokenGen}(\text{SK}_{\text{id},L}, \text{PK}, \omega')$, where $L \neq P$ and $\text{id} \in T_1$; then $\text{Search}(\text{TK}_{\text{id},L}, \text{CT}, T_1)$ outputs 1 when $\omega = \omega'$.

3.2. System Model. As depicted in Figure 1, our TR-HABKS system includes four entities: a trusted authority (TA), a data owner (DO), a cloud sever (CS), and multiple data users (DUs). Specifically, the role of each entity in our system model is described below.

TA: TA first runs the setup algorithm, keeps the master key MK secretly, and publishes the public parameter PK. Then, he uses his master key to generate private keys for DO and DUs. In addition, he creates an identity table T_1 for user revocation and another identity table T_2 for the malicious user tracing. When a malicious user sells his private key on the Internet, TA runs the tracing algorithm and then obtains the malicious user identity id from T_2 . Finally, TA deletes id from table T_1 and sends T_1 to CS to revoke the malicious user's search ability.

DO: when DO wants to encrypt a keyword ω under an access policy P , he runs the encryption algorithm with his private key SK_o and then generates a ciphertext CT corresponding to (P, ω) . Finally, he outsources the corresponding ciphertext CT to the cloud.

DU: when DU wants to search the data files with the keyword ω' , he runs the token generation algorithm with his private key $\text{SK}_{\text{id},L}$ and then generates a search token $\text{TK}_{\text{id},L}$ corresponding to (id, L, ω') . Finally, he sends $\text{TK}_{\text{id},L}$ to CS to query documents containing the keyword ω' .

CS: when CS receives the search token $\text{TK}_{\text{id},L}$ from DU, he first searches id in the table T_1 . If $\text{id} \notin T_1$, CS returns 0 and aborts; otherwise, CS runs the searching algorithm and returns the search result to DU.

In our threat model, TA and DO are assumed to be fully trusted; that is, they execute the above algorithm honestly and will not attack the system. CS is assumed to be an honest-but-curious adversary who honestly executes the searching algorithm but tries to infer the privacy of keywords. Note that the generation of ciphertext needs to use the private key of DO, so CS cannot generate ciphertext by itself and carry out keyword guessing attack. DUs in our system may be malicious adversaries who not only try to retrieve the ciphertext beyond their retrieval permission but also leak their private keys to others.

3.3. Security Model. In order to realize the confidentiality of keywords and access policies simultaneously, the security

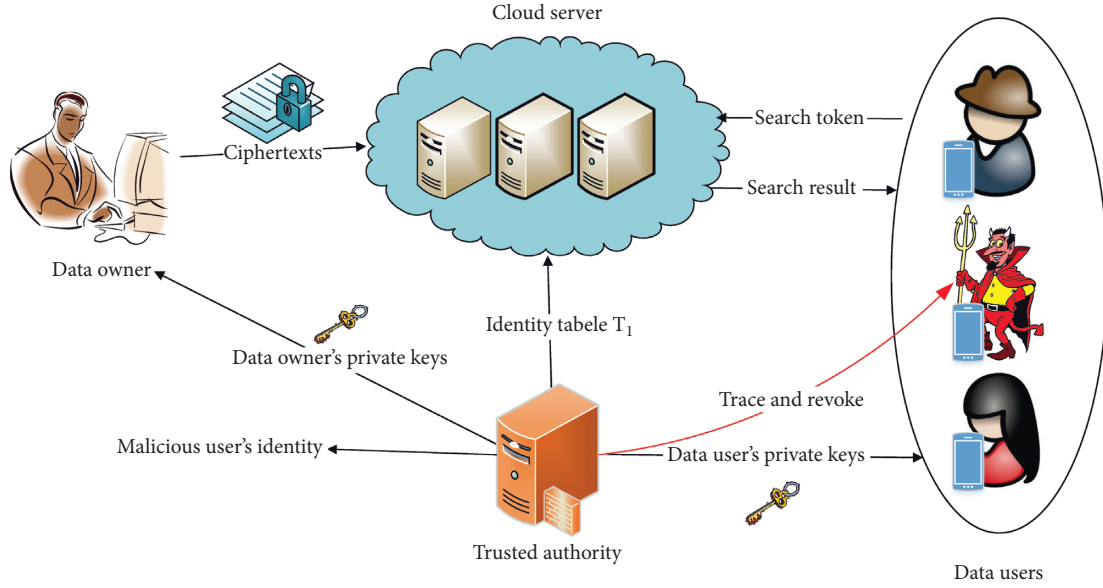


FIGURE 1: System model of TR-HABKS.

model of our TR-HABKS scheme requires that the adversary cannot distinguish between the encryption of a keyword ω_0 under an access policy P_0 and the encryption of a keyword ω_1 under an access policy P_1 . In the selective security model, the adversary needs to submit two challenge access policies P_0 and P_1 before the Setup phase. The selective security game includes the following phases:

Inti: the adversary declares two challenge access policies P_0 and P_1 that he tries to attack and gives them to the challenger.

Setup: the challenger calls the setup algorithm $(MK, PK) \leftarrow \text{Setup}(\lambda)$ and gives the public parameter PK to the adversary.

Query Phase1: the adversary can repeatedly ask for private keys and search tokens as follows:

- (1) *Private Key Query* $\mathcal{O}_{\text{KeyGen}}(\text{id}, L)$: the adversary submits an identity id and an attributes set L to the challenger. If $(L \neq P_0 \wedge L \neq P_1)$ or $(L \neq P_0 \wedge L \neq P_1)$, then abort; otherwise, the challenger returns the corresponding private key $SK_{\text{id}, L}$.
- (2) *Search Token Query* $\mathcal{O}_{\text{TokenGen}}(\text{id}, L, \omega)$: the adversary submits an identity id , an attributes set L , and a keyword ω to the challenger. The challenger returns the corresponding search token $TK_{\text{id}, L}$.

Challenge: the adversary submits two keywords ω_0 and ω_1 that satisfy the following constraint. If the adversary has queried the private key or search token for the attributes set L that satisfies both access policies P_0 and P_1 , then we require that $\omega_0 = \omega_1$. The challenger flips a random coin $\gamma \in \{0, 1\}$ and returns the challenge ciphertext $CT^* \leftarrow \text{Enc}(\omega_\gamma, P_\gamma, SK_o, PK)$ to the adversary.

Query Phase2: phase 1 is repeated with the restriction that the adversary cannot query the private key or search token for the attributes set L when $(L \neq P_0 \wedge L \neq P_1)$ and $\omega_0 \neq \omega_1$.

Guess: the adversary outputs a guess $\gamma' \in \{0, 1\}$.

The adversary wins this game if $\gamma = \gamma'$, and his advantage is defined as $\Pr[\gamma = \gamma'] - (1/2)$.

Definition 2. A TR-HABKS scheme is said to be selectively secure if all PPT adversaries have only a negligible advantage in the above security game.

The traceability game of TR-HABKS is described as follows:

Setup: the challenger runs the setup algorithm $(MK, PK) \leftarrow \text{Setup}(\lambda)$ and gives the public parameter PK to the adversary.

Key query: the adversary queries the private keys corresponding to pairs $\{(\text{id}_j, L_j)\}_{j \in [q_s]}$, where id_j is an identity, L_j is an attributes set, and q_s is the maximum number of private keys that the adversary can query. The challenger returns the corresponding user private keys $\{SK_{\text{id}_j, L_j}\}_{j \in [q_s]}$.

Key forgery: the adversary outputs a user private key SK^* .

In this game, the adversary's advantage is defined as $\Pr[\text{Trace}(SK^*, PK, T_2) \notin \{\text{id}_1, \text{id}_2, \dots, \text{id}_{q_s}, \top\}]$.

Definition 3. A TR-HABKS scheme is said to be fully traceable if all PPT adversaries have only a negligible advantage in this traceability game.

4. Our TR-HABKS Scheme

In this section, we propose the construction of our TR-HABKS scheme and prove that it is selectively secure and fully traceable in the generic bilinear group model. We first adopt the technique from [18, 25] to realize hidden policy. The access policy P is embedded in the ciphertext CT as follows: if $v_{i,t} \in P_i$, we set $C_{i,t,1} = A_{i,t,1}^{\alpha_i}, C_{i,t,2} = A_{i,t,2}^{\alpha_i}$; otherwise, we set $C_{i,t,1}$ and $C_{i,t,2}$ as two random elements in G_1 . That is, if $v_{i,t} \in P_i$, these ciphertext components $C_{i,t,1}, C_{i,t,2}$ are well formed and can be used for successful search; otherwise, the ciphertext components $C_{i,t,1}, C_{i,t,2}$ are malformed. As it is hard to distinguish the well-formed ciphertext components from the malformed ciphertext components, the user cannot get the access policy from the corresponding ciphertext. Then, we exploit the signature technique in [31, 44] to realize the user traceability. On one hand, we inject the message y_{id} and its signature into the user private key; then DU cannot rerandomize the private key component y_{id} . On the other hand, we add the message y_{id} and the corresponding user identity id in the identity table T_2 ; then TA can identify the malicious user by the private key and the table T_2 . Finally, we add the legitimate user to the system by storing the user identity id and its corresponding element C_{id} in the registered identity table T_1 and revoke the malicious user by deleting the corresponding pair (id, C_{id}) from the table T_1 .

4.1. Construction. Setup (λ) : TA first runs $\mathcal{G}(\lambda)$ to obtain $(p, G_1, G_2, G_T, g_1, g_2, e)$, where G_1, G_2 , and G_T are cyclic groups with prime order p , and $e: G_1 \times G_2 \rightarrow G_T$ is a bilinear map. Then, TA picks $a, b, c \leftarrow_R \mathbb{Z}_p$ and a one-way hash function $H: \{0, 1\}^* \rightarrow \mathbb{Z}_p$. For each $i \in [n]$, TA chooses random exponents $\{a_{i,t} \in \mathbb{Z}_p\}_{t \in [n_i]}$ and computes $\{A_{i,t,1} = g_1^{a_{i,t}}, A_{i,t,2} = g_1^{ca_{i,t}}\}_{t \in [n_i]}$. Next, TA sets $MK = (a, b, c, \left\{ \{a_{i,t}\}_{t \in [n_i]} \right\}_{i \in [n]})$ as his master key and publishes the public parameter $PK = (p, G_1, G_2, G_T, g_1, g_2, e, e(g_1, g_2)^a, g_1^b, g_1^c, g_1^{bc}, H, \left\{ \{A_{i,t,1}, A_{i,t,2}\}_{t \in [n_i]} \right\}_{i \in [n]})$. Finally, TA creates two empty identity tables T_1 and T_2 .

KeyGen (MK, PK, id, L) : DU submits his identity id and attributes set $L = \{v_{i,t_i}\}_{i \in [n]}$ to TA in order to apply for the user private key. TA first picks $x_{id}, y_{id}, \beta \leftarrow_R \mathbb{Z}_p$ and sets $K = x_{id}, K_0 = g_2^{a/b(c+y_{id})} g_2^{\beta/b}, K_1 = y_{id}$. For each $i \in [n]$, TA picks $\lambda_i \leftarrow_R \mathbb{Z}_p$ and computes $K_{i,1} = g_2^{\beta+\lambda_i a_{i,t_i}}, K_{i,2} = g_2^{\lambda_i}$. Then, TA sets $SK_{id,L} = (K, K_0, K_1, \{K_{i,1}, K_{i,2}\}_{i \in [n]})$ as DU private key and sends it to the DU with identity id . Next, TA picks $\alpha \leftarrow_R \mathbb{Z}_p$, sets $SK_o = \alpha$ as DO private key, and sends it to DO. After that, TA computes $C_{id} = e(g_1, g_2)^{-\alpha x_{id}}$, stores

(id, C_{id}) in the registered identity table T_1 , and sends T_1 to CS for search permission revocation. Finally, TA adds (id, y_{id}) in the identity table T_2 and secretly stores T_2 for user tracing.

Enc (ω, P, SK_o, PK) : to encrypt a keyword under an access policy $P = \{P_i\}_{i \in [n]}$, DO computes $C = e(g_1, g_2)^{\alpha \omega}, C_0 = g_1^{ba/H(\omega)}, C_1 = g_1^{bca/H(\omega)}$. For each $i \in [n]$, DO chooses $\alpha_i \leftarrow_R \mathbb{Z}_p$ such that $\sum_{i \in [n]} \alpha_i = \alpha$, computes $C_{i,1} = g_1^{\alpha_i}, C_{i,2} = g_1^{c\alpha_i}$, and sets $C_{i,t}$ for each $t \in [n_i]$ as follows: if $v_{i,t} \in P_i$, it sets $C_{i,t,1} = A_{i,t,1}^{\alpha_i}, C_{i,t,2} = A_{i,t,2}^{\alpha_i}$; otherwise, it sets $C_{i,t,1}$ and $C_{i,t,2}$ as two random elements in G_1 . Finally, DO uploads the ciphertext $CT = (C, C_0, C_1, \left\{ C_{i,1}, C_{i,2}, \{C_{i,t,1}, C_{i,t,2}\}_{t \in [n_i]} \right\}_{i \in [n]})$ into the cloud.

TokenGen $(SK_{id,L}, PK, \omega')$: to generate a search token for a keyword $\omega' \in \{0, 1\}^*$, DU picks $s \leftarrow_R \mathbb{Z}_p$ and computes $tok_0 = K_0^{H(\omega')^s}, tok = K + s, tok_1 = K_1$. For each $i \in [n]$, DU computes $T_{i,1} = K_{i,1}^s, T_{i,2} = K_{i,2}^s$. Finally, DU sends the search token $TK_{id,L} = (tok_0, tok, tok_1, \{T_{i,1}, T_{i,2}\}_{i \in [n]})$ to CS.

Search $(TK_{id,L}, CT, T_1)$: when CS receives the search token $TK_{id,L} = (tok_0, tok, tok_1, \{T_{i,1}, T_{i,2}\}_{i \in [n]})$ from the DU with identity id , it first searches the entry (id, C_{id}) in the table T_1 . If no such entry exists, CS returns 0 and aborts; otherwise, CS obtains C_{id} from T_1 and runs the following search algorithm. *If Algorithm.* If $L = \{v_{i,t_i}\}_{i \in [n]}$, it computes $E = \prod_{i \in [n]} (e(C_{i,1}^{tok_1} C_{i,2}^{tok}, T_{i,1}) / e(C_{i,t,1}^{tok_1} C_{i,t,2}^{tok}, T_{i,2})) = e(g_1, g_2)^{\alpha \beta s (y_{id} + c)}$. Finally, CS returns 1 if $EC_{id}^{tok} = e(C_0^{tok_1} C_1, tok_0)$ and 0 otherwise.

Trace $(SK_{id,L}, PK, T_2)$: if the private key is not in the form of $SK_{id,L} = (K, K_0, K_1, \{K_{i,1}, K_{i,2}\}_{i \in [n]})$, TA returns \perp and aborts; otherwise, TA runs the following key sanity check algorithm. $K, K_1 \in \mathbb{Z}_p, K_0, K_{i,1}, K_{i,2} \in G_2, \exists i \in [n], s.t.$

$$e(g_1^{bc} g_1^{bK_1}, K_0) e(A_{i,t,1}^{K_1} A_{i,t,2}, K_{i,2}) = e(g_1, g_2)^a e(g_1^c g_1^{K_1}, K_{i,1}). \quad (1)$$

If the private key $SK_{id,L}$ does not pass the above check, TA returns \perp and aborts; otherwise, TA searches the entry (id, K_1) in table T_2 and returns the corresponding id .

Revoke (id, T_1) : to revoke the search permission of the malicious user with identity id , TA updates table T_1 by deleting the entry (id, C_{id}) and sends new table T_1 to CS.

4.2. Correctness Proof. We now prove the correctness of our TR-HABKS scheme. *If Scheme.* If the user attributes $L = \{v_{i,t_i}\}_{i \in [n]}$ satisfy the access policy $P = \{P_i\}_{i \in [n]}$, we have $v_{i,t_i} \in P_i$ and $C_{i,t,1} = A_{i,t,1}^{\alpha_i}, C_{i,t,2} = A_{i,t,2}^{\alpha_i}$ for each $i \in [n]$. Then,

$$\begin{aligned}
& \prod_{i \in [n]} \frac{e(C_{i,1}^{\text{tok}_1} C_{i,2}, T_{i,1})}{e(C_{i,t,1}^{\text{tok}_1} C_{i,t,2}, T_{i,2})} \\
&= \prod_{i \in [n]} \frac{e(g_1^{\alpha_i y_{\text{id}}} g_1^{c \alpha_i}, K_{i,1}^s)}{e(A_{i,t,1}^{\alpha_i y_{\text{id}}} A_{i,t,2}^{\alpha_i}, K_{i,2}^s)} \\
&= \prod_{i \in [n]} \frac{e(g_1^{y_{\text{id}}+c}, g_2^{\beta + \lambda_i a_{i,t_i}})^{s \alpha_i}}{e(g_1^{(y_{\text{id}}+c) a_{i,t_i}}, g_2^{\lambda_i})^{s \alpha_i}} \\
&= \prod_{i \in [n]} \frac{e(g_1^{y_{\text{id}}+c}, g_2^\beta)^{s \alpha_i} e(g_1^{y_{\text{id}}+c}, g_2^{\lambda_i a_{i,t_i}})^{s \alpha_i}}{e(g_1^{(y_{\text{id}}+c) a_{i,t_i}}, g_2^{\lambda_i})^{s \alpha_i}} \tag{2} \\
&= \prod_{i \in [n]} e(g_1, g_2)^{\beta s (y_{\text{id}}+c) \alpha_i} \\
&= e(g_1, g_2)^{\beta s (y_{\text{id}}+c) \sum_{i \in [n]} \alpha_i} \\
&= e(g_1, g_2)^{\alpha \beta s (y_{\text{id}}+c)}.
\end{aligned}$$

If the user id is in table T_1 , then CS can obtain the corresponding $C_{\text{id}} = e(g_1, g_2)^{-\alpha \alpha x_{\text{id}}}$. Therefore,

$$\begin{aligned}
& \text{EC}^{\text{tok}} C_{\text{id}} \\
&= e(g_1, g_2)^{\alpha \beta s (y_{\text{id}}+c)} e(g_1, g_2)^{\alpha \alpha (x_{\text{id}}+s)} e(g_1, g_2)^{-\alpha \alpha x_{\text{id}}} \tag{3} \\
&= e(g_1, g_2)^{\alpha \beta s (y_{\text{id}}+c)} e(g_1, g_2)^{\alpha \alpha s}.
\end{aligned}$$

In this case, if $\omega = \omega'$, we have

$$\begin{aligned}
& e(C_0^{\text{tok}_1} C_1, \text{tok}_0) \\
&= e\left(g_1^{b \alpha y_{\text{id}}/H(\omega)} g_1^{bc \alpha/H(\omega)}, K_0^H(\omega')^s\right) \\
&= e\left(g_1^{b \alpha (c+y_{\text{id}})}, g_2^{(a/b)(c+y_{\text{id}})} g_2^{\beta/b}\right)^s \tag{4} \\
&= e\left(g_1^{b \alpha (c+y_{\text{id}})}, g_2^{a/b(c+y_{\text{id}})}\right)^s e\left(g_1^{b \alpha (c+y_{\text{id}})}, g_2^{\beta/b}\right)^s \\
&= e(g_1, g_2)^{\alpha \alpha s} e(g_1, g_2)^{\alpha \beta s (y_{\text{id}}+c)} \\
&= \text{EC}^{\text{tok}} C_{\text{id}}.
\end{aligned}$$

4.3. Proof of Selective Security. In this part, we prove the confidentiality of keywords and access policies in our scheme by a security reduction to the QLSZ scheme [18]. More specifically, if there are any attacks in our TR-HABKS scheme, then we can use these attacks to break the QLSZ scheme in the generic bilinear group model [18, 45]. Followed by the definition in [45], we consider three random encodings $\varphi_1, \varphi_2, \varphi_T: F_p \rightarrow \{0, 1\}^m$, where F_p is an additive group and $m > 3 \log(p)$. For $i = 1, 2, T$, let

$G_i = \{\varphi_i(x): x \in F_p\}$. Therefore, there are three oracles to compute the group action on G_1, G_2, G_T and an oracle to compute the bilinear map e . We refer to G_1 as a generic bilinear group. In addition, our TR-HABKS scheme only allows DO to generate ciphertext by his private key, so the adversary cannot successfully carry out the keyword guessing attack.

Theorem 1. *If the QLSZ scheme is selectively secure in the generic bilinear group model, then our TR-HABKS scheme is selectively secure.*

Proof. Suppose that there exists a PPT adversary \mathcal{A} that can break our TR-HABKS scheme with advantage ε in the selective security model. We will build a simulator \mathcal{B} that can break the QLSZ scheme with advantage ε . Let \mathcal{C} be the challenger corresponding to \mathcal{B} in the security game of QLSZ scheme. For more information about the QLSZ scheme and its security, please refer to [18].

Inti: simulator \mathcal{B} receives two challenge access policies P_0 and P_1 from adversary \mathcal{A} and then sends these policies to challenger \mathcal{C} .

Setup: \mathcal{C} sends the QLSZ public parameter $\overline{\text{PK}} = (p, G_1, G_2, G_T, g_1, g_2, e, e(g_1, g_2)^a, g_1^b, H, \left\{ \{A_{i,t}\}_{t \in [n_i]} \right\}_{i \in [n]})$ to \mathcal{B} . Then, \mathcal{B} picks $c \leftarrow_{\mathcal{R}} \mathbb{Z}_p$, sets $\{A_{i,t,1} = A_{i,t}, A_{i,t,2} = A_{i,t}^c\}_{t \in [n_i]}$, and sends the public parameter $\text{PK} = (p, G_1, G_2, G_T, g_1, g_2, e, e(g_1, g_2)^a, g_1^b, g_1^c, g_1^{bc}, H, \left\{ \{A_{i,t,1}, A_{i,t,2}\}_{t \in [n_i]} \right\}_{i \in [n]})$ to \mathcal{A} .

$1, K_0^{(K_1+c)} = e(g_1, g_2^{a+(K_1+c)\beta_i-\lambda_i(K_1+c)a_{i,t_i}})$. Therefore, $K_0^b = g_2^{(a+(K_1+c)\beta_i-\lambda_i(K_1+c)a_{i,t_i})/ (K_1+c)} = g_2^{a/(K_1+c)} g_2^{\beta_i} g_2^{-\lambda_i a_{i,t_i}} = g_2^{a/(K_1+c)} K_{i,1} K_{i,2}^{-a_{i,t_i}}$.

Finally, \mathcal{B} computes $\sigma = [K_0^b K_{i,2}^{a_{i,t_i}} / K_{i,1}]^{1/a} = g_2^{1/(K_1+c)}$ and then obtains a valid signature σ on message K_1 , where $K_1 \notin \{y_1, y_2, \dots, y_{q_i}\}$. Hence, if \mathcal{A} has advantage ε in the traceability game, then \mathcal{B} can forge a valid BB basic signature scheme with advantage ε under a weak chosen message attack.

5. Our eTR-HABKS System

In this section, we describe our enhanced TR-HABKS system based on our TR-HABKS scheme in Section 4. Different from the TR-HABKS scheme, the tracing algorithm in this system is public traceable and does not require any identity table. In addition, the efficiency comparison shows that the storage overhead of the eTR-HABKS system is much smaller than that of the TR-HABKS scheme.

5.1. Concrete System

5.1.1. System Initialization. In this phase, TA generates the system parameter, the master key for himself, and an identity table for revocation.

TA first runs $\mathcal{G}(\lambda)$ to obtain $(p, G_1, G_2, G_T, g_1, g_2, e)$. For each $i \in [n]$, TA chooses random exponents $\{a_{i,t} \in \mathbb{Z}_p\}_{t \in [n_i]}$ and computes $\{A_{i,t} = g_1^{a_{i,t}}\}_{t \in [n_i]}$. Then, TA picks $a, b, c, d \leftarrow_R \mathbb{Z}_p$ and a one-way hash function $H: \{0, 1\}^* \rightarrow \mathbb{Z}_p$. Next, TA sets $MK = (a, b, c, d, \{ \{a_{i,t}\}_{t \in [n_i]} \}_{i \in [n]})$ as his master key and publishes the public parameter $PK = (p, G_1, G_2, G_T, g_1, g_2, e, e(g_1, g_2)^a, g_1^b, g_1^c, g_1^d, g_1^{bc}, g_1^{bd}, H, \{ \{A_{i,t}\}_{t \in [n_i]} \}_{i \in [n]})$. Finally, TA creates an empty identity table T_1 .

5.1.2. User Registration. In this phase, TA uses his master key to generate the private keys for the registered DUs and DO.

When DU wants to join the system, he submits his identity $\text{id} \in \mathbb{Z}_p$ and attributes set $L = \{v_{i,t_i}\}_{i \in [n]}$ to TA to apply for his private key. TA first picks $x_{\text{id}}, r, \beta \leftarrow_R \mathbb{Z}_p$ and sets $K = x_{\text{id}}, K_0 = g_2^{a/b(c+\text{id}+dr)} g_2^{\beta/b}, K_1 = \text{id}, K_2 = r$. For each $i \in [n]$, TA picks $\lambda_i \leftarrow_R \mathbb{Z}_p$ and computes $K_{i,1} = g_2^{\beta+\lambda_i a_{i,t_i}}, K_{i,2} = g_2^{\lambda_i}, K_{i,3} = g_2^{(c+dr)\lambda_i}$. TA sets $SK_{\text{id},L} = (K, K_0, K_1, K_2, \{K_{i,1}, K_{i,2}, K_{i,3}\}_{i \in [n]})$ as the user private key and sends it to the corresponding DU. Then, TA picks $\alpha \leftarrow_R \mathbb{Z}_p$ and sets $SK_o = \alpha$ as the data owner private key and sends it to

DO. Finally, TA computes $C_{\text{id}} = e(g_1, g_2)^{-\alpha x_{\text{id}}}$, stores $(\text{id}, C_{\text{id}})$ in the identity table T_1 , and sends T_1 to CS.

5.1.3. Secure Index Generation. In this phase, DO uses his private key to generate a secure index for each file and outsources all the files and indexes in the cloud.

When DO wants to share a file with the specific data users, he extracts a keyword $\omega \in \{0, 1\}^*$ from the file and encrypts the keyword ω under an access policy $P = \{P_i\}_{i \in [n]}$. DO first computes $C = e(g_1, g_2)^{a\alpha}, C_0 = g_1^{b\alpha/H(\omega)}, C_1 = g_1^{bc\alpha/H(\omega)}, C_2 = g_1^{bd\alpha/H(\omega)}$. For each $i \in [n]$, DO chooses $\alpha_i \leftarrow_R \mathbb{Z}_p$ such that $\sum_{i \in [n]} \alpha_i = \alpha$, computes $C_{i,1} = g_1^{\alpha_i}, C_{i,2} = g_1^{c\alpha_i}, C_{i,3} = g_1^{d\alpha_i}$, and sets $C_{i,t,2}$ for each $t \in [n_i]$ as follows: if $v_{i,t} \in P_i$, it sets $C_{i,t,2} = A_{i,t}^{\alpha_i}$; otherwise, it sets $C_{i,t,2}$ as a random element in G_1 . Finally, DO stores the encrypted index $CT = (C, C_0, C_1, C_2, \{C_{i,1}, C_{i,2}, C_{i,3}, \{C_{i,t,2}\}_{t \in [n_i]}\}_{i \in [n]})$ in the cloud.

5.1.4. Search Token Generation. In this phase, DU generates a search token for a keyword $\omega' \in \{0, 1\}^*$, and sends the search token to CS for the data retrieval request.

DU first picks $s \leftarrow_R \mathbb{Z}_p$, computes $\text{tok}_0 = K_0^{H(\omega')^s}, \text{tok} = K + s, \text{tok}_1 = K_1, \text{tok}_2 = K_2$. For each $i \in [n]$, DU computes $T_{i,1} = K_{i,1}^s, T_{i,2} = K_{i,2}^s, T_{i,3} = K_{i,3}^s$. Finally, DU sets the search token $TK_{\text{id},L} = (\text{tok}_0, \text{tok}, \text{tok}_1, \text{tok}_2, \{T_{i,1}, T_{i,2}, T_{i,3}\}_{i \in [n]})$.

5.1.5. Data Retrieval. In this phase, CS uses the token to search the data in the cloud and responds the search results to DU.

When CS receives the retrieval request and the search token $TK_{\text{id},L} = (\text{tok}_0, \text{tok}, \text{tok}_1, \text{tok}_2, \{T_{i,1}, T_{i,2}, T_{i,3}\}_{i \in [n]})$ from DU, it first searches the entry $(\text{id}, C_{\text{id}})$ in T_1 . If no such entry exists, CS returns error symbol \perp and aborts; otherwise, CS obtains C_{id} from T_1 and then runs the following search algorithm. *If Algorithm.* If $L = \{v_{i,t_i}\}_{i \in [n]}$, it computes $E = \prod_{i \in [n]} (e(C_{i,1}^{\text{tok}_1} C_{i,2}^{\text{tok}_2} T_{i,1}) / e(C_{i,t,2}, T_{i,2} \text{tok}_1 T_{i,3})) = (g_1, g_2)^{\alpha\beta s(c+\text{id}+dr)}$. Finally, CS returns 1 if $EC_{\text{id}}^{\text{tok}} C_{\text{id}} = e(C_0^{\text{tok}_1} C_1 C_2^{\text{tok}_2}, \text{tok}_0)$ and 0 otherwise.

5.1.6. User Tracing. In this phase, TA traces the malicious user who sales his private key $SK_{\text{id},L}$ on the Internet and outputs the malicious user's identity.

TA first checks whether $SK_{\text{id},L}$ is a well-formed key. If the private key is not in the form of $SK_{\text{id},L} = (K, K_0, K_1, K_2, \{K_{i,1}, K_{i,2}, K_{i,3}\}_{i \in [n]})$, it returns \top and aborts;

otherwise, it runs the following key sanity check algorithm.
 $K, K_1, K_2 \in \mathbb{Z}_p, K_0, K_{i,1}, K_{i,2}, K_{i,3} \in G_2, \exists i \in [n], \text{ s.t.}$

$$\begin{aligned} e(g_1^c g_1^{dK_2}, K_{i,2}) &= e(g_1, K_{i,3}), \\ e(g_1^{bc} g_1^{bK_1} g_1^{b dK_2}, K_0) e(A_{i,t_i}, K_{i,2}^{K_1} K_{i,3}) &= e(g_1, g_2)^a e(g_1^c g_1^{K_1} g_1^{dK_2}, K_{i,1}). \end{aligned} \quad (5)$$

If $\text{SK}_{\text{id},L}$ does not pass the above check, it returns τ and aborts; otherwise, it returns K_1 as the corresponding user identity.

5.1.7. User Revocation. In this phase, TA revokes the search permissions of the malicious users. When TA obtains the malicious user identity id , he updates table T_1 by deleting the entry $(\text{id}, C_{\text{id}})$ and sends the new table T_1 to CS.

5.2. Correctness Proof. The correctness of our TR-HABKS scheme is proved as follows. If the user attributes $L = \{v_{i,t_i}\}_{i \in [n]}$ satisfy the access policy $P = \{P_i\}_{i \in [n]}$, we have $v_{i,t_i} \in P_i$ and $C_{i,t_i,2} = A_{i,t_i}^{\alpha_i}$ for each $i \in [n]$. Then,

$$\begin{aligned} & \prod_{i \in [n]} \frac{e(C_{i,1}^{\text{tok}_1} C_{i,2} C_{i,3}^{\text{tok}_2}, T_{i,1})}{e(C_{i,t_i,2}, T_{i,2}^{\text{tok}_1} T_{i,3})} \\ &= \prod_{i \in [n]} \frac{e(g_1^{\text{id} \cdot \alpha_i} g_1^{c \alpha_i} g_1^{d \cdot r \alpha_i}, K_{i,1}^s)}{e(A_{i,t_i}^{\alpha_i}, K_{i,2}^{\text{id} \cdot s} K_{i,3}^s)} \\ &= \prod_{i \in [n]} \frac{e(g_1^{\text{id}+c+dr}, g_2^{\beta + \lambda_i a_{i,t_i}})^{s \alpha_i}}{e(g_1^{a_{i,t_i}}, g_2^{\lambda_i (\text{id}+c+dr)})^{s \alpha_i}} \\ &= \prod_{i \in [n]} \frac{e(g_1^{\text{id}+c+dr}, g_2^\beta)^{s \alpha_i} e(g_1^{\text{id}+c+dr}, g_2^{\lambda_i a_{i,t_i}})^{s \alpha_i}}{e(g_1^{a_{i,t_i}}, g_2^{\lambda_i (\text{id}+c+dr)})^{s \alpha_i}} \quad (6) \\ &= \prod_{i \in [n]} e(g_1, g_2)^{\beta s (\text{id}+c+dr) \alpha_i} \\ &= e(g_1, g_2)^{\beta s (\text{id}+c+dr) \sum_{i \in [n]} \alpha_i} \\ &= e(g_1, g_2)^{\alpha \beta s (\text{id}+c+dr)}. \end{aligned}$$

If the user id is in the table T_1 , then CS has the corresponding $C_{\text{id}} = e(g_1, g_2)^{-\alpha x_{\text{id}}}$. Therefore,

$$\begin{aligned} & \text{EC}^{\text{tok}} C_{\text{id}} \\ &= e(g_1, g_2)^{\alpha \beta s (\text{id}+c+dr)} e(g_1, g_2)^{\alpha x (\text{id}+s)} e(g_1, g_2)^{-\alpha x_{\text{id}}} \\ &= e(g_1, g_2)^{\alpha \beta s (\text{id}+c+dr)} e(g_1, g_2)^{\alpha s}. \end{aligned} \quad (7)$$

In this case, if $\omega = \omega'$, we have

$$\begin{aligned} & e(C_0^{\text{tok}_1} C_1 C_2^{\text{tok}_2}, \text{tok}_0) \\ &= e(g_1^{(\text{id} \cdot b a)/H(\omega)} g_1^{b c a/H(\omega)} g_1^{r \cdot b d a/H(\omega)}, K_0^H(\omega')^s) \\ &= e(g_1^{b a (\text{id}+c+dr)}, g_2^{a/b (\text{id}+c+dr)} g_2^{\beta/b})^s \\ &= e(g_1^{b a (\text{id}+c+dr)}, g_2^{a/b (\text{id}+c+dr)})^s e(g_1^{b a (\text{id}+c+dr)}, g_2^{\beta/b})^s \\ &= e(g_1, g_2)^{a a s} e(g_1, g_2)^{\alpha \beta s (\text{id}+c+dr)} \\ &= \text{EC}^{\text{tok}} C_{\text{id}}. \end{aligned} \quad (8)$$

The security proofs of our eTR-HABKS scheme are almost the same as that in Section 4, so we omit the details here.

5.3. Comparison. Table 2 compares the storage costs of our schemes with that of QLSZ scheme [18]. The length of the public parameter/ciphertext of all three schemes increases linearly with $\sum_{i \in [n]} n_i$, where n is the total number of attributes in the system and n_i is the number of possible values for attribute index i . Compared with QLSZ scheme, the public key and ciphertext size of our TR-HABKS scheme have almost doubled, but the public key and ciphertext size of our eTR-HABKS scheme are only increased by 4 and $2 + n$ elements, respectively. The user private key/token size of all schemes grows linearly with the total number of attributes, and the user private key/token of the eTR-HABKS scheme is about 1.5 times as long as that of other schemes. Note that n is far less than $\sum_{i \in [n]} n_i$, so the system storage overhead of the eTR-HABKS scheme is much less than that of the TR-HABKS scheme, although the user storage overhead of the eTR-HABKS scheme is slightly greater. In addition, the eTR-HABKS scheme only needs to maintain an identity table T_1 for revocation but does not require any identity table for tracing, which makes our eTR-HABKS scheme more practical. Figure 2 illustrates the system storage overhead for tracing (including the public parameter and the storage for tracing) in our TR-HABKS and eTR-HABKS schemes. We set the group element size to 160 bits and the random number and identity size to 1024 bits, and $\sum_{i \in [n]} n_i = 100$. From Figure 2, it is easy to see that the system storage overhead for tracing in our eTR-HABKS scheme is constant and significantly smaller than that grows linearly with the number of users in TR-HABKS scheme.

Table 3 gives a computation cost comparison that ignores nondominant operations in the schemes. E_1, E_2 , and E_T denote an exponentiation operation in groups G_1, G_2 , and G_T , respectively. P is a bilinear pairing operation and $|P_i|$ ($|P_i| \leq n_i$) is the number of attribute values in P_i . Let ‘‘Trace (max)’’ and ‘‘Trace (min)’’ denote the maximum and

TABLE 2: Storage cost comparison.

Scheme	[18]	TR-HABKS	eTR-HABKS
Public parameter size	$4 + \sum_{i \in [n]} n_i$	$6 + 2 \sum_{i \in [n]} n_i$	$8 + \sum_{i \in [n]} n_i$
User private key size	$2 + 2n$	$3 + 2n$	$4 + 3n$
Ciphertext size	$2 + n + \sum_{i \in [n]} n_i$	$3 + 2n + 2 \sum_{i \in [n]} n_i$	$4 + 3n + \sum_{i \in [n]} n_i$
Token size	$2 + 2n$	$3 + 2n$	$4 + 3n$
The storage for tracing	—	$ T_2 $	0
The storage for revocation	—	$ T_1 $	$ T_1 $

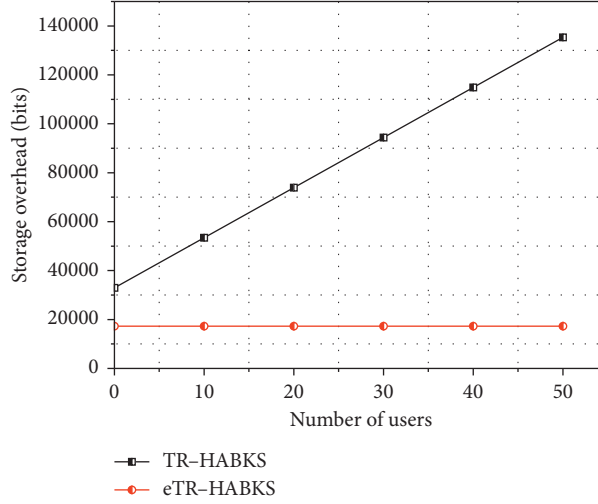


FIGURE 2: System storage overhead for tracing.

TABLE 3: Computation cost comparison.

Scheme	[18]	TR-HABKS	eTR-HABKS
Setup	$(1 + \sum_{i \in [n]} n_i)E_1 + E_T$	$(3 + 2 \sum_{i \in [n]} n_i)E_1 + E_T$	$(5 + \sum_{i \in [n]} n_i)E_1 + E_T$
KeyGen	$(1 + 2n)E_2 + E_T$	$(1 + 2n)E_2 + E_T$	$(1 + 3n)E_2 + E_T$
Enc	$(1 + n + \sum_{i \in [n]} P_i)E_1 + E_T$	$(2 + 2n + 2 \sum_{i \in [n]} P_i)E_1 + E_T$	$(3 + 3n + \sum_{i \in [n]} P_i)E_1 + E_T$
TokenGen	$(1 + 2n)E_2$	$(1 + 2n)E_2$	$(1 + 3n)E_2$
Search	$(1 + 2n)P + E_1$	$(1 + 2n)P + (2 + 2n)E_1$	$(1 + 2n)P + (3 + 2n)E_1 + nE_2$
Trace (max)	—	$(1 + 2n)P + (2 + n)E_1$	$(1 + 4n)P + 4E_1 + nE_2$
Trace (min)	—	$3P + 3E_1$	$5P + 4E_1 + E_2$
Revoke	—	0	0

minimum computation cost for successful tracing, respectively. From Table 3, we can see that the computation cost in the setup and encryption algorithms of the eTR-HABKS scheme is almost the same as that of QLSZ scheme, but it is obviously smaller than that of the TR-HABKS scheme. All three schemes have the same level of computation overhead in key generation and token generation algorithms. Compared with QLSZ scheme, our TR-HABKS and eTR-HABKS schemes do not add any computation overhead to achieve user revocation but add more computation overhead to realize user accountability. However, the increased computation burden has little effect on the performance of our eTR-HABKS system, because the search and tracing algorithms can be executed by the cloud with powerful computing capability.

6. Conclusion

In this paper, we first presented a new privacy-preserving ABKS construction for cloud-assisted IoT and then proved

that it is selectively secure and fully traceable in the generic bilinear group model. We also proposed another ABKS construction with public traceability and showed that it is more efficient than the first construction. In short, our two constructions not only reduce privacy leakage by hiding access policies but also prevent private key abuse by tracing and revoking malicious users. As our schemes are designed for just one-owner setting, we aim to construct a traceable and revocable ABKS scheme with policy protection in multiowner setting in the future.

Data Availability

No data were used to support the findings of this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61802243 and 11801345), the Natural Science Foundation of Shaanxi Province (Grant nos. 2019JQ-273 and 2020JM-288), and the Key Research and Development Program in Industry Field of Shaanxi Province (Grant no. 2019GY-013).

References

- [1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [2] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: efficient policy-hiding attribute-based access control," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2130–2145, 2018.
- [3] L. Zhao and X. Dong, "An industrial internet of things feature selection method based on potential entropy evaluation criteria," *IEEE Access*, vol. 6, pp. 4608–4617, 2018.
- [4] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
- [5] Y. Liu, X. Ma, L. Shu et al., "Internet of things for noise mapping in smart cities: state of the art and future directions," *IEEE Network*, vol. 34, no. 4, pp. 112–118, 2020.
- [6] A. Sajid, H. Abbas, and K. Saleem, "Cloud-assisted iot-based scada systems security: a review of the state of the art and future challenges," *IEEE Access*, vol. 4, pp. 1375–1384, 2016.
- [7] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of industrial internet of things security: requirements and fog computing opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2489–2520, 2020.
- [8] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of iot," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2018.
- [9] X. Han, L. Wang, S. Xu, D. Zhao, and G. Liu, "Recognizing roles of online illegal gambling participants: an ensemble learning approach," *Computers & Security*, vol. 87, Article ID 101588, 2019.
- [10] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in dwsns," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
- [11] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pp. 213–229, Springer, Santa Barbara, CA, USA, August 2001.
- [12] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proceedings of the 2004 International conference on the theory and applications of cryptographic techniques*, pp. 506–522, Inter-laken, Switzerland, May 2004.
- [13] M. Abdalla, M. Bellare, D. Catalano et al., "Searchable encryption revisited: consistency properties, relation to anonymous ibe, and extensions," *Journal of Cryptology*, vol. 21, no. 3, pp. 350–391, 2008.
- [14] Q. Zheng, S. Xu, and G. Ateniese, "Vabks: verifiable attribute-based keyword search over outsourced encrypted data," in *Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 522–530, IEEE, Toronto, ON, Canada, April 2014.
- [15] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," in *Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 226–234, IEEE, Toronto, ON, Canada, April 2014.
- [16] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," *Information Sciences*, vol. 403–404, pp. 1–14, 2017.
- [17] S. Liu, J. Yu, Y. Xiao, Z. Wan, S. Wang, and B. Yan, "Bc-sabe: blockchain-aided searchable attribute-based encryption for cloud-iot," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7851–7867, 2020.
- [18] S. Qiu, J. Liu, Y. Shi, and R. Zhang, "Hidden policy ciphertext-policy attribute-based encryption with keyword search against keyword guessing attack," *Science China Information Sciences*, vol. 60, no. 5, Article ID 052105, 2017.
- [19] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Aarhus, Denmark, May 2005.
- [20] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, Alexandria, VA, USA, October 2006.
- [21] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE symposium on security and privacy (SP'07)*, pp. 321–334, IEEE, 2007.
- [22] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *International Workshop on Public Key Cryptography*, pp. 53–70, Berkeley, CA, USA, May 2011.
- [23] N. Attrapadung and H. Imai, "Dual-policy attribute based encryption," in *Proceedings of the 2009 International Conference on Applied Cryptography and Network Security*, pp. 168–185, Paris-Rocquencourt, France, June 2009.
- [24] S. Xu, Y. Li, R. Deng, Y. Zhang, X. Luo, and X. Liu, "Lightweight and expressive fine-grained access control for healthcare internet-of-things," *IEEE Transactions on Cloud Computing*, 2019.
- [25] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proceedings of the 2008 International conference on applied cryptography and network security*, pp. 111–129, New York, NY, USA, June 2008.
- [26] J. Lai, R. H. Deng, and Y. Li, "Expressive cp-abe with partially hidden access structures," in *Proceedings of the 7th ACM symposium on information, computer and communications security*, pp. 18–19, Seoul, South Korea, May 2012.
- [27] A. Beimel, *Secure schemes for secret sharing and key distribution*, Ph.D. Dissertation, Technion-Israel Institute of Technology, Haifa, Israel, 1996.
- [28] K. Yang, Q. Han, H. Li, K. Zheng, Z. Su, and X. Shen, "An efficient and fine-grained big data access control scheme with privacy-preserving policy," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 563–571, 2016.
- [29] J. Sun, H. Xiong, X. Liu, Y. Zhang, X. Nie, and R. H. Deng, "Lightweight and privacy-aware fine-grained access control for iot-oriented smart health," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6566–6575, 2020.

- [30] M. J. Hinek, S. Jiang, R. S. Naini, and S. F. Shahandashti, "Attribute-based encryption without key cloning," *International Journal of Applied Cryptography*, vol. 2, no. 3, pp. 250–270, 2012.
- [31] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 76–88, 2012.
- [32] Z. Liu, Z. Cao, and D. S. Wong, "Blackbox traceable cp-abe: How to catch people leaking their keys by selling decryption devices on ebay," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pp. 475–486, Hangzhou China, May 2013.
- [33] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1274–1288, 2015.
- [34] Z. Ying, Y. Si, J. Ma, X. Liu, and S. Xu, "Fhpt: fine-grained ehr sharing in e-healthcare cloud with hidden policy and traceability," in *Proceedings of the 2020 GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–6, IEEE, Taipei, Taiwan, December 2020.
- [35] S. Xu, J. Ning, Y. Li et al., "Match in my way: fine-grained bilateral access control for secure cloud-fog computing," *IEEE Transactions on Dependable and Secure Computing*, 2020.
- [36] B. Qin, Q. Zhao, D. Zheng, and H. Cui, "(Dual) server-aided revocable attribute-based encryption with decryption key exposure resistance," *Information Sciences*, vol. 490, pp. 74–92, 2019.
- [37] S. Xu, G. Yang, Y. Mu, and X. Liu, "A secure iot cloud storage system with fine-grained access control and decryption key exposure resistance," *Future Generation Computer Systems*, vol. 97, pp. 284–294, 2019.
- [38] J. W. Byun, H. S. Rhee, H.-A. Park, and D. H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," in *Proceedings of the 2006 Workshop on secure data management*, pp. 75–83, Seoul, South Korea, September 2006.
- [39] Y. Miao, Q. Tong, R. Deng, K.-K. R. Choo, X. Liu, and H. Li, "Verifiable searchable encryption framework against insider keyword-guessing attack in cloud storage," *IEEE Transactions on Cloud Computing*, 2020.
- [40] Y. Miao, Q. Tong, K.-K. R. Choo, X. Liu, R. H. Deng, and H. Li, "Secure online/offline data sharing framework for cloud-assisted industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8681–8691, 2019.
- [41] Y. Miao, X. Liu, K.-K. R. Choo et al., "Privacy-preserving attribute-based keyword search in shared multi-owner setting," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, 2019.
- [42] J. Sun, H. Xiong, X. Nie, Y. Zhang, and P. Wu, "On the security of privacy-preserving attribute-based keyword search in shared multi-owner setting," *IEEE Transactions on Dependable and Secure Computing*, 2019.
- [43] Z. Liu, Y. Liu, J. Xu, and B. Wang, "Privacy-preserving attribute-based multi-keyword search encryption scheme with user tracing," in *Proceedings of the 2019 International Symposium on Cyberspace Safety and Security*, pp. 382–397, Guangzhou, China, December 2019.
- [44] D. Boneh and X. Boyen, "Short signatures without random oracles and the sdh assumption in bilinear groups," *Journal of Cryptology*, vol. 21, no. 2, pp. 149–177, 2008.
- [45] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Proceedings of the 2005 Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 440–456, Aarhus, Denmark, May 2005.

Research Article

Aggregating Heterogeneous Sensor Ontologies with Fuzzy Debate Mechanism

Xingsi Xue ¹, **Xiaojing Wu** ¹, **Jie Zhang**², **Lingyu Zhang** ³, **Hai Zhu** ⁴,
and **Guojun Mao**¹

¹Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, Fujian, 350118, China

²School of Computer Science and Engineering, Yulin Normal University, Yulin, Guanxi, 537000, China

³School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou, Fujian, 350118, China

⁴School of Network Engineering, Zhoukou Normal University, Zhoukou, Henan, 466001, China

Correspondence should be addressed to Xingsi Xue; jack8375@gmail.com

Received 5 April 2021; Revised 29 April 2021; Accepted 15 May 2021; Published 27 May 2021

Academic Editor: James Ying

Copyright © 2021 Xingsi Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at enhancing the communication and information security between the next generation of Industrial Internet of Things (Nx-IIoT) sensor networks, it is critical to aggregate heterogeneous sensor data in the sensor ontologies by establishing semantic connections in diverse sensor ontologies. Sensor ontology matching technology is devoted to determining heterogeneous sensor concept pairs in two distinct sensor ontologies, which is an effective method of addressing the heterogeneity problem. The existing matching techniques neglect the relationships among different entity mapping, which makes them unable to make sure of the alignment's high quality. To get rid of this shortcoming, in this work, a sensor ontology extraction method technology using Fuzzy Debate Mechanism (FDM) is proposed to aggregate the heterogeneous sensor data, which determines the final sensor concept correspondences by carrying out a debating process among different matchers. More than ever, a fuzzy similarity metric is presented to effectively measure two entities' similarity values by membership function. It first uses the fuzzy membership function to model two entities' similarity in vector space and then calculate their semantic distance with the cosine function. The testing cases from Bibliographic data which is furnished by the Ontology Alignment Evaluation Initiative (OAEI) and six sensor ontology matching tasks are used to evaluate the performance of our scheme in the experiment. The robustness and effectiveness of the proposed method are proved by comparing it with the advanced ontology matching techniques.

1. Introduction

In the research era of the Next generation of Industrial Internet of Things (Nx-IIoT), the network technology and intelligent computing has become a huge technical model for the government to establish a smart world [1, 2]. Security issues in the Internet of Things (IoT) have also sparked concern with researches rolling in. Particularly, Xiong et al. [3] proposed a LightPrivacy scheme to achieve the tradeoff between user's personalization privacy protection and the availability of task data in mobile group awareness, whose

computational efficiency was significantly improved. Later, they further presented an ATG framework, which was both effective and efficient, and suitable for IoT Mobile Edge Crowd Sensing (MECS) [4]. More recently, Lin et al. [5] proposed an Ant Colony Optimization (ACO) approach to protect information by the transaction deletion, which was able to reduce the side effects while keeping the overall computing cost low. In this fashion, a large number of physical objects embedded with sensors devices exchange information through heterogeneous networks in various applications such as the smart grid, electronic medical

treatment, and smart cities [6–10]. To aggregate the information of systems efficiently, their entities should be able to interact with one another in meaningful ways without special effort by humans or machines. And it is worth mentioning that many diverse sensor data management application frameworks have been proposed for uniting and dealing sensors. At the same time, value-added information is provided by spatial data sources for public applications, including sensor networks such as Global Sensor Network (<https://gsn.sourceforge.net/>), Hourglass (<https://www.eecs.harvard.edu/~syrah/hourglass/>), and IrisNet (<https://www.intel-iris.net/>). However, the heterogeneity of different sensor networks resulting in a lack of interoperability. Therefore, to build a secure Nx-IIoT, the way of aggregate heterogeneous sensor data in different sensor networks is prominent.

Specifically, there are two aspects of work to be done; one is to improve the expressive ability of sensor network models, and the other is to enhance the interaction between sensor networks to achieve data integration. To address the first aspect, an increasing number of sensor ontologies have appeared because of the preponderance of sensor ontology technique, which is able to model the corresponding networks integrally. And to address the other aspect, the ontology matching technique has been on the stage of history in recent years. Furthermore, one of the cutting-edge research institutions in this field is Ontology Alignment Evaluation Initiative (OAEI) (<https://oei.ontologymatching.org/>). Recently, ontology alignment extraction technique has been used to strengthen the team. To advance the relevant work, we propose a mechanism for sensor ontology matching with the Fuzzy Debate Mechanism (FDM) based ontology alignment extraction technique, which aims to extract the correct sensor ontology matching pairs in different alignments generated by different basic matching measures. To be specific, we first express the similarity between two sensor entities in the three-dimensional vector space through a fuzzy membership function and then evaluate the similarities in multiple dimensions and the cosine theorem is introduced to evaluate the distance of similarity vectors and the golden one.

The following sections are arranged as follows. Section 2 is an overview of the related work. Section 3 presents a preliminary analysis of the relevant concepts. Details of FDM are provided by Section 4. Section 5 externalizes experiments' results and makes the corresponding analysis, and Section 6 concludes the work.

2. Related Work

A growing number of sensor ontologies have appeared due to the sensor ontology possesses powerful sensor network model expression ability, i.e., SensorOntology 2009 ontology, SSN ontology and IoT-Lite ontology, and so on [11–13]. And to enhance the interaction between sensor networks to achieve data integration, the sensor ontology matching technique has been brought out these years [14].

In the research upsurge, there are two technical routes: ontology meta-matching (OMM) techniques and ontology

entity matching techniques. The ontology entity matching techniques try to determine the entity correspondence set between two ontologies directly, while the OMM techniques try to solve the problem of aggregate different similarity measures with appropriate weights [15]. There are plenty of popular technical approaches in computing intelligence (CI) to solve OMM problems, e.g., machine learning (ML), evolutionary computing (EC), and swarm intelligence (SI). For example, many ML technologies [16–21] have been proposed to automatically determine ontology alignment, and experiments have shown that ML greatly improves matching efficiency, and the genetics for ontology alignment (GOAL), which was designed to optimize aggregate weight sets for different matchers [22–25]. To overcome the disadvantage of excessive reliance on reference alignment, Xue et al. put forward the partial reference alignment (PRA) and the unanimous improvement rate (UIR) [26]. Furthermore, Xue et al. proposed a series solution using compact algorithm (CA) and sensor ontology meta-matching technique to aggregate weight sets for different matchers [27].

However, in the existing ontology matching methods based on CI, various ontology matchers are regarded as tools with the same effect and try to aggregate their outputs by determining the optimal weights [28]. The matching quality will decrease if ignoring the influence of different entity mappings on the matching results of different matching devices. Furthermore, adjusting weights with this method can be problematic; that means they may not be reusable in different ontology matching scenarios. Hence, the influence of entity mappings on matching result cannot be ignored, which is addressed by ontology alignment extraction technique [29]. Recently, the context extraction technique has been widely applied in semantic field [26, 30], in which OntoLT uses terminology extraction, ontology structure mapping, the statistical method, and the language model of definition to extract ontology concept [31]. Besides, Gaeta et al. implemented several statistical and data mining algorithms to identify and extract the concepts as well as their relationships in ontologies [32]. In addition, the rule of extraction process is typically described as a series of agreement reaching processes, such as argumentation frameworks [33]. In concrete terms, the argumentation framework proposed by Laera et al. relies on preferences between a formal argument operation pattern and a particular type of argument, taking into account ontology-based arguments and propositions specific to the matching task [34]. Dos-Santos and Euzenat used argumentation as a supporting or rejecting parameter and proposed a computational strategy to remove inconsistencies in the result alignment and allow consistency in the argumentation system [35].

It is worth mentioning that the similarity of ontology matching and the relation between entities and the similarity threshold can all be regarded as uncertain problems, in which fuzzy logic is highly adept [36]. In the proposal of [37], the ontology matches are expressed by the fuzzy set of reference concepts or instances that makes the new ontology be directly compared with the original one. Todorov et al. proposed a fuzzy ontology alignment using background

knowledge [38]. UFOM adopts fuzzy set theory as the general framework of fuzzy ontology matching, which represents many types of correspondences and describe the uncertainty in the process of correspondence discovery [39]. And Cross discusses how to extend the process of ontology concept matching by using similarity measure and integration of fuzzy sets [40].

3. Problem Definition

3.1. Sensor Ontologies. An ontology is composed of concept set, attribute set, and instance set, and the ontology O refers to three tuples (C, P_d, P_o) , among which C, P_d, P_o refer to concept set, properties of datatype, and properties of object, respectively, called ontology entities. The semantic sensor network (SSN) (<https://purl.oclc.org/NET/ssnx/ssn#>) ontology can be regarded as the authoritative ancestor of sensor ontology [41]. It is an OWL 2 ontology put forward by the W3C Semantic Sensor Network Incubator group (SSN-XG) (<https://www.w3.org/2005/Incubator/ssn>), which models sensors and observation data and represents sensors in accordance with the function, measurement process, observation data, and so on. SensorOntology 2009 (<https://www.w3.org/2005/Incubator/ssn/wiki/SensorOntology2009>) ontology developed by Michael Compton, i.e., from CSIRO (Australia) has come into use as the source of the SSN Ontology. And OSSN (<https://www.w3.org/ns/ssn>) is an ontology established by SSN-XG in the year from 2009 to 2011 [42]. Furthermore, the resources, entities, and services in the Internet of Things (IoT) are summarized by the IoT-Lite (<https://www.w3.org/Submission/2015/SUBM-iot-lite-20151126/>) ontology outlines, which is a lightweight ontology and a case of an SSN ontology and the latest version was submitted in 2015 [43]. The Sensors, Observations, Samples, and Actuators (SOSA) (<https://www.w3.org/ns/sosa>) ontology is designed for a broad target audience and applications that have access to the ontology and was released in 2017. In addition, SOSA acts as a minimal interoperability fallback layer; that is, it defines those public classes and attributes whose data can be securely exchanged between the SSN, its modules, and all information used by SOSA.

3.2. Ontology Matching and Sensor Ontology Alignment Extraction. Ontology matching is the process of determining the entity correspondence between source ontology and target ontologies to bridge the semantic gap between them. As shown in Figure 1, the input of ontology matching process is a pair of ontologies to-be-matched, and the output is the final alignment. The research on ontology matching mostly involves the calculation and refinement of similarity measure, but the research on extracting the final alignment from similarity measure matrix, which is called ontology alignment extraction technique, is less.

To be specific, sensor ontology alignment extraction technique works by extracting entity correspondences from different matching suggestions generated from different similarity measures for the same sensor ontology matching task to form the final alignment [28]. The framework of

ontology alignment extraction is shown in Figure 1, in which the systems to finish the matching work are regarded as ontology matchers. The set of entity correspondences determined by an ontology matcher is called an ontology alignment. In addition, a corresponding set A is the alignment between two sensor ontologies, where the entity correspondence is referred to a 4-tuple $\text{corr} = (e, e', n, \text{relation})$, e and e' are, respectively, two ontologies' entities, and $n \in [0, 1]$ is their confidence value, while relation acts as the equivalence relation.

3.3. Fuzzy Similarity Measure. Since some of the wireless sensor network (WSN) domain concepts have not yet been incorporated into a common dictionary, there is no lexical tool to define the linguistic relationships between all concepts for mapping purposes. To conquer the difficulty, our proposal uses a variety of metrics as tools in similarity calculations. Each tool gives different matching suggestions; the matching results between entities become uncertain information. In this case, a unified framework for representing many different modes of inconclusive information is provided by the use of a fuzzy measure to depict inconclusive information [44]. For this reason, we decide to use the fuzzification process to combine them to get more accurate alignment. In other words, the work of fuzzy similarity measurement adopted here is supplemented by two parts; one is the basic similarity measurements, also known as the entity matching measure (EMM), and the other is the membership function of fuzzification.

An EMM is always described as the function to output similar values with information from two entities as input in the interval $[0, 1]$, which always plays the role of a basic ontology matcher. EMM can be grouped into three types: first is the string-based type, second is the linguistics-based type, and the third one is called taxonomy-based EMM.

A string-based EMM outputs the edit distance between entities by considering their IDs, tags, comments, and so on. In this work, two well-known EMMs were used to improve the quality of the matching results, namely, the Levenshtein similarity [45] and the Jaro similarity metric [46]. Given two strings, s_{e1} and s_{e2} , Levenshtein-similarity is defined in the following:

$$\text{Levenshtein}(s_{e1}, s_{e2}) = \frac{\max(0, \min(|s_{e1}|, |s_{e2}|) - d(s_{e1}, s_{e2}))}{\min(|s_{e1}|, |s_{e2}|)}, \quad (1)$$

where $|s_{e1}|$ and $|s_{e2}|$ are, respectively, the length of s_{e1} and s_{e2} ; $d(s_{e1}, s_{e2})$ is the number of required operations to transform s_{e1} to s_{e2} ; given the strings s and t , define s' as a character common to t in s ; that is, they appear in the same order in s and t . Similarly define t' in t . Now, assume that s' and t' are transposed at position i when their i th bits are not the same and the value of $T_{s', t'}$ is 50% of the transpositions of s' and t' in number. According to the assumption above, the Jaro similarity measure of s and t is defined in the following:

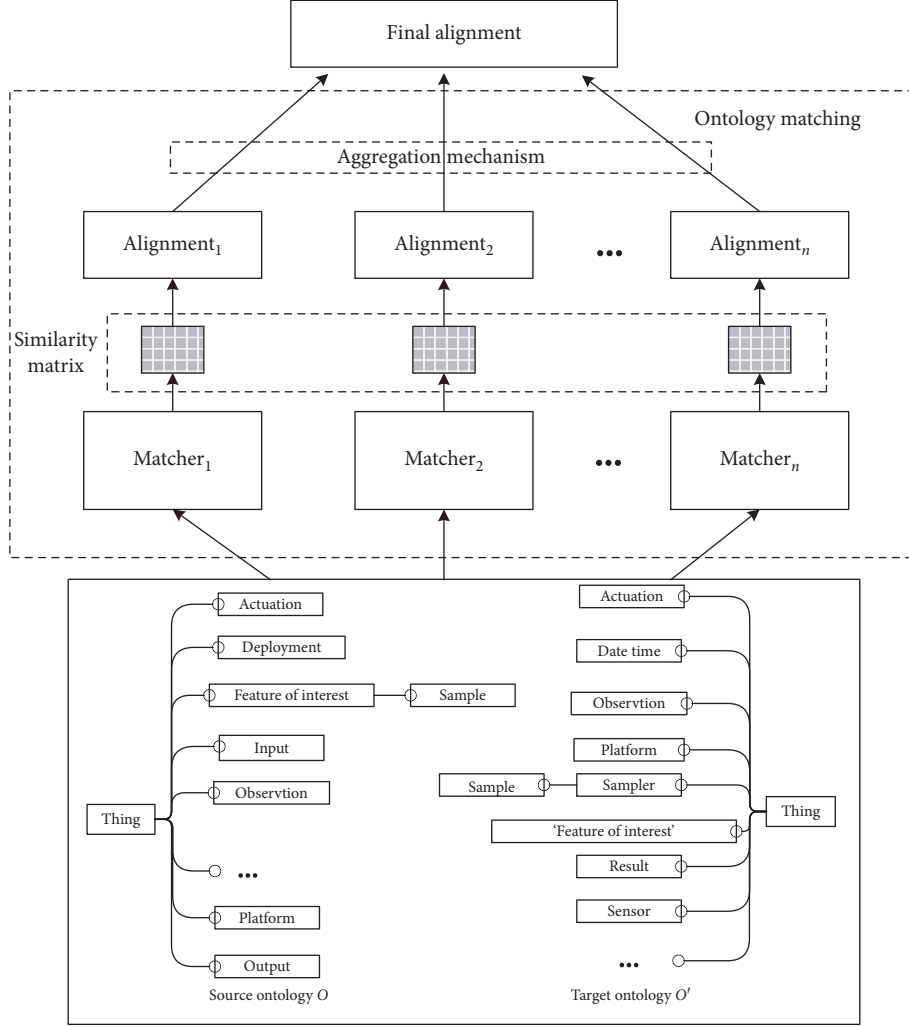


FIGURE 1: Framework of ontology alignment extraction.

$$\text{Jaro}(s, t) = \frac{1}{3} \cdot \left(\frac{|s'|}{|s|} + \frac{|t'|}{|t|} + \frac{|s'| - T_{s',t'}}{|s'|} \right). \quad (2)$$

Linguistics-based EMMs often use external corpus or dictionaries (e.g., WordNet [47]) to calculate similarity values through entity tags. Because of the effectiveness of the WordNet, we used WordNet-based distances. Besides, based on the distance WordNet-based on the WordNet electronic vocabulary, the equation is defined as follows:

$$\text{WordSim}(w_1, w_2) = \max_{c_1 \in \text{sen}(w_1), c_2 \in \text{sen}(w_2)} [\text{sim}(c_1, c_2)], \quad (3)$$

where w_1 and w_2 are a word derived from two entities and $\text{sen}(w_i)$ denotes the number of words w_i 's meanings.

The similarity values obtained by the taxonomy-based EMMs are based on the consideration of the entity structure information as well as the superentity and subentity information. To be specific, in our proposal, the taxonomy-based EMM makes use of the structure-based distance calculated by noted algorithm, which adopts a versatile

graph matching method called similarity flooding (SF) [48], where an iterative fixpoint computation below is applied to produce congruent relationship between the two ontologies' elements:

$$\delta^{i+1} = \text{norm}(\delta^i + f(\delta^i)). \quad (4)$$

In this formula, norm is the normalized process, δ^i is the value of the last iteration that changes in each iteration, and f is a function of increasing the similarity of pairs of elements based on the similarity of their adjacent elements.

The flowchart of the fuzzy similarity measure application process is depicted in Figure 2. The work in pre-processing generally includes conversion ontology format and ontology analysis. And the similarity matrixes are generated from the four basically similar measures adopted in this article. Usually, a matcher determines whether two entities are correct matching pairs by comparing the threshold value with the similarity value. But it is difficult to find an accurate threshold to make the matching result completely correct. Therefore, we introduce the membership function in fuzzy theory and consider the similarity value from the ‘‘low,’’

“medium,” and “high” dimensions. The fuzzy process changes the elements in the similarity matrixes and changes them into vectors through membership function before participating in the Debate Mechanism. In this work, the membership function $\mu(x)$ of fuzzy process is defined as three subfunctions as follows:

$$\begin{aligned} \mu_{\text{Low}}(x) &= \begin{cases} -\frac{10}{7}x + 1, & x \in [0, 0.7), \\ 0, & x \in [0.7, 1], \end{cases} \\ \mu_{\text{Medium}}(x) &= \begin{cases} 2x, & x \in [0, 0.5), \\ -2x + 2, & x \in [0.5, 1], \end{cases} \\ \mu_{\text{High}}(x) &= x, \quad x \in [0, 1], \end{aligned} \quad (5)$$

where x is an element of a similarity matrix which stands for the similarity n . According to previous practical experience, we believe that matching pairs with similarity less than 0.7 are mostly mismatched, and 0.5 is a medium similarity in similarity interval $[0, 1]$ [28]. Therefore, μ_{Low} is used to measure the degree of low similarity. μ_{Medium} describes the degree of medium size and then uses the direct proportionality function to describe the degree of high similarity as the μ_{High} . After that, we take the three function values as the coordinates of the 3D vector \vec{f} and input them into the Debate Mechanism.

4. Debate Mechanism

In this work, ontology alignment extraction process is carried out by the Debating Mechanism, in which the debating rules is utilised to extract the target information. The Debating Mechanism contains a classification module of correspondences, where a fuzzy measure is built to express the similarities of correspondences, and an argumentation framework, which is used to negotiate different matching suggestions between matchers to reach agreement. In addition, a fuzzy measure is built to express the similarities of correspondences as is shown in Figure 3 that depicts the framework of Debate Mechanism.

The classification module and argumentation framework are described in the following paragraphs.

In the classification module, assume that a matching task is working on two ontologies, O and O' , by k ($k \geq 2$) basic ontology matchers. Extend an entity correspondence in an ontology matcher as an argument ar , which is defined as follows:

$$ar = \{c, n, v, h\}, \quad (6)$$

where $c = (e, e')$, v ($v \in N$), and h ($h \in \{0, 1\}$), respectively, express a correspondence, the artificially preset matcher number, and the measure factor of similarity value.

Assume that reference vector $\vec{m} = (0, 0, 1)$; change the similarity value n into similarity fuzzy vector $\vec{f} = (\mu_{\text{Low}}(n), \mu_{\text{Medium}}(n), \mu_{\text{High}}(n))$ in vector space. Then, we describe h as follows:

$$h = \begin{cases} 1, & \text{if } \cos(\vec{f}, \vec{m}) \geq \delta, \\ 0, & \text{if } \cos(\vec{f}, \vec{m}) < \delta, \end{cases} \quad (7)$$

where \vec{f} is a similarity fuzzy vector and δ ($\delta \in [0, 1]$) is set as the similarity's threshold. Especially, while $h = 0$, the matcher rejects c ; otherwise, it accepts it. Suppose that c is allocated to one of the five groups C_i , $i = 1, \dots, 5$, that are, respectively, defined as follows: k ($k \geq 2$) is the number of matchers, k_a is the number of matchers that accepts c , k_r is the number of matchers that rejects c , $k_r = k - k_a$. Next, classify c into groups from C_1 to C_5 in category column according to the above situation. For detailed classification, see Table 1.

Since $c \in C_1$ is regarded as a correct correspondence which is accepted by all the matchers, while $c \in C_5$ is rejected oppositely, the correspondences in groups C_1 and C_5 are straightway judged as right correspondences or the false ones in the process of extraction without participating in the following process. Besides, correspondences of C_2 , C_3 , and C_4 groups are in list of the argumentation process.

In the process of argumentation, two arguments are given as $a = \{c_1, \vec{f}_1, v_1, h_1\}$ and $b = \{c_2, \vec{f}_2, v_2, h_2\}$. Four relationships are defined between b and a , which are unite, attack, support, and disprove. To be specific, unite is marked as $U(b, a)$, and attack, support, and disprove are expressed as $A(b, a)$, $S(b, a)$, and $D(b, a)$. The details are listed in the following descriptions:

- (i) When $c_1 = c_2, v_1 \neq v_2, h_1 = h_2$, b is united with a , which is denoted as $U(b, a)$.
- (ii) When $c_1 = c_2, v_1 \neq v_2, h_1 \neq h_2$, b attacks a , which is denoted as $A(b, a)$.
- (iii) When $\vec{c}_2 = C_2 \rightarrow$ or $\vec{c}_2 = C_3, v_1 = v_2, \cos(\vec{f}_1, \vec{m}) > \cos(\vec{f}_2, \vec{m}), h_1 = h_2 = 1$, or when $\vec{c}_2 = C_4$ or $C_3, \cos(\vec{f}_1, \vec{m}) < \cos(\vec{f}_2, \vec{m}), h_1 = h_2 = 0$, b supports a , which is represented by $S(b, a)$.
- (iv) When $v_1 = v_2, c_1 = C_i, c_2 = C_j, i > j$ ($i, j \in (2, 3, 4)$), $n_1 > n_2, h_1 = 1, h_2 = 0$, or when $v_1 = v_2, c_1 = C_i, c_2 = C_j, i < j$ ($i, j \in (2, 3, 4)$), $\cos(\vec{f}_1, \vec{m}) < \cos(\vec{f}_2, \vec{m}), h_1 = 0, h_2 = 1$, b disproves a , that is depicted as $D(b, a)$.

The four relationships between arguments are depicted in Figure 4. There are three matchers, i.e., Matcher_a , Matcher_b , and Matcher_c including their arguments a_i , b_i , and c_i . Unite and attack happened between arguments from different matchers, but support and disprove from the same one.

The arguments set is defined as a 7-tuple: $\{ar, strength, U, A, S, D, M\}$, where U, A, S, D are the relationships mentioned above, $M = \{m_1, m_2, \dots, m_n\}$ is defined as the set of matchers that contains n basic ontology

matchers, argument ar is related to correspondence c , and $astrength$ n d acts as strength value of c according to a matcher m_i , which is defined as follows:

$$\text{Strength}_c^{m_i} = \cos\left(\sum_{ar \in AR} \{\vec{f}_x | x \in AR \wedge S(x, ar)\}, \vec{m}\right) - \cos\left(\sum_{ar \in AR} \{\vec{f}_x | x \in AR \wedge D(x, ar)\}, \vec{m}\right). \quad (8)$$

In this paper, c is an element of corresponding arguments existing in every basic ontology matcher, and we need to calculate its judgment factor r_c ($r_c \in \{0, 1\}$) whose value is determined in argumentation process to determine whether it can be extracted into the final alignment. Consequently, an essential challenge is to improve the reliability of an entity mapping's judgment element. To meet that challenge, as is defined in formula (9), each matcher's correctness factor is adopted to evaluate the credibility of the matcher depending on how similar it is to the matching recommendations of other matchers. And the support strength and disprove strength in debating process are utilized to offer the evidence of right mappings. In addition, the r_c can be obtained after full assessment of the factors mentioned above.

To be specific, the argumentation process is arranged as follows:

Step 1. Apparently, r_c is 1 (or 0) when c belongs to C_1 (or C_5), and the similarity values of corresponding rows and columns of c can be deleted from the similarity matrix.

Step 2. Matcher m_i 's correctness factor is calculated in the following:

$$\sigma_{m_i} = \frac{\sum_{m_i} |\{c | c \in (C_1, C_5)\}|}{\sum_{m_i} |c|}. \quad (9)$$

Step 3. In every matcher, the debating process is brought forward in accordance with the relationships "support" and "disprove":

- (1) In the C_2 group, most matchers support these correspondences. Therefore, the success of the supporting part in defeating the disproving part is calculated. The above situation is explained as follows. The support strength Ss of matcher m_1 is defined as follows:

$$Ss_t^{m_1} = \cos\left(\sum_{x \in AR} \vec{f}_x, \vec{m}\right) - \cos\left(\sum_{y \in AR} \vec{f}_y, \vec{m}\right), \quad (10)$$

where argument $x = \{c, n_x, v_x, h_x\}$, argument $y = \{c, n_y, v_y, h_y\}$, $S(x, t)$, $D(y, t)$ and $v_x = v_y$. When three matchers m_1, m_2 , and m_3 support c but m_4 disproves it, it is determined that the mapping can be established between e and e' when $Ss_t^{m_1} > Ss_t^{m_4}$, $Ss_t^{m_2} > Ss_t^{m_4}$ and $Ss_t^{m_3} > Ss_t^{m_4}$, and r_c is set to 1. Otherwise, c is transformed into C_3 , performing step 4.

- (2) In a similar way, for the C_4 group of argument, it is necessary to calculate whether the attack of the opposing side against the supporting side is successful. Assuming that the matcher m_1 's disprove strength Ds is defined as follows:

$$Ds_t^{m_1} = \cos\left(\sum_{x \in AR} \vec{f}_x, \vec{m}\right) - \cos\left(\sum_{y \in AR} \vec{f}_y, \vec{m}\right), \quad (11)$$

where argument $x = \{c, n_x, v_x, h_x\}$, argument $y = \{c, n_y, v_y, h_y\}$, $D(x, t)$, $S(y, t)$ and $v_x = v_y$. In this situation, if the three matchers m_1, m_2 , and m_3 disprove c , but m_4 support, obviously, the mapping cannot be established between e_i and e'_j when $Ds_t^{m_1} > Ds_t^{m_4}$, $Ds_t^{m_2} > Ds_t^{m_4}$ and $Ds_t^{m_3} > Ds_t^{m_4}$, and $r_c = 0$. Otherwise, c is converted into C_3 group.

Step 4. For the arguments in C_3 group, the number of matchers who took the opposite view was almost even, due to what they can be divided into two party called accept party and reject party. Therefore, the core challenge is to figure out which of the two parties defeat the other successfully. To end this, the power P_{stren} of each party is calculated, which is defined as follows:

$$P_{\text{stren}}^{\text{party}} = \delta_{m_1} \cdot \text{Strength}_c^{m_1} + \delta_{m_2} \cdot \text{Strength}_c^{m_2} + \dots + \delta_{m_n} \cdot \text{Strength}_c^{m_n}, \quad (12)$$

where m_1, m_2, \dots, m_n are the members of the party. Assume that for c , if $P_{\text{stren}}^{\text{accept}} \geq P_{\text{stren}}^{\text{reject}}$, $r_c = 1$. Otherwise, $r_c = 0$.

Step 5. Select the correspondences with $r_c = 1$, which are further used to decide the final alignment.

5. Experiment and Results

5.1. Alignment Evaluation Metric. The alignment can typically be assessed with two measures, often referred to as recall and precision [49], which are severally ruled in the following:

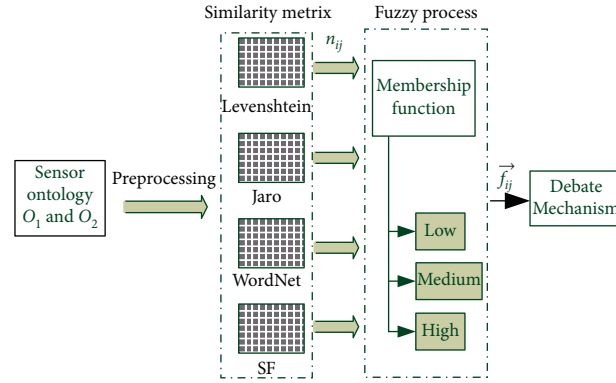


FIGURE 2: The flowchart of fuzzy similarity measure application process.

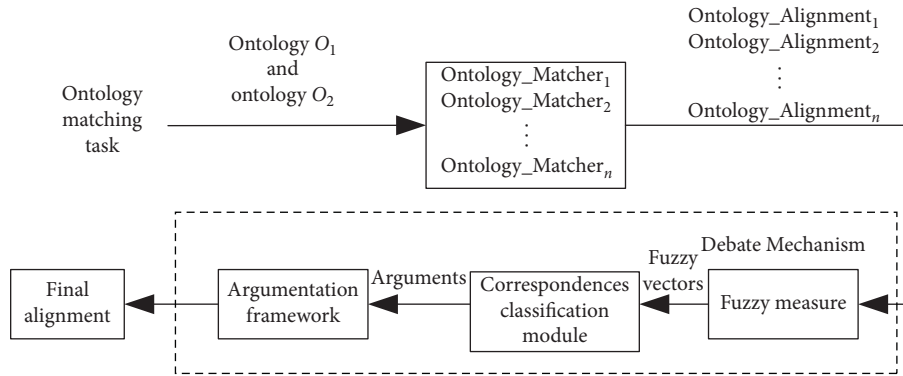


FIGURE 3: The framework of Debate Mechanism.

$$\text{recall} = \frac{|R \cap A|}{|R|}, \quad (13)$$

$$\text{precision} = \frac{|R \cap A|}{|A|},$$

where the alignment given is A , and the reference alignment is R . Particularly, $\text{recall} = 1$ when found all correct matching pairs, $\text{recall} = 1$. And $\text{precision} = 1$ stands for that all the matching pairs found are correct. In order to combine the two metrics, the f -measure is further employed, which is regarded as a comprehensive measure of recall and precision [38]:

$$f\text{-measure} = \frac{2 \text{ precision} \cdot \text{recall}}{\text{recall} + \text{precision}}. \quad (14)$$

5.2. Experimental Testing Cases. In this experiment, we take advantage of the testing cases in Bibliographic track (<https://oaei.ontologymatching.org/2016/results/benchmarks/index.html>) from OAEI as well as six pairs of real sensor ontology matching tasks to verify the sensitivity and availability of our recommendation. Table 2 shows a brief description of OAEI's Bibliographic track, where two ontologies to be mapped and a reference alignment to evaluate the

effectiveness of ontology matcher are included by each test case. Table 3 depicts the main features of sensor ontologies.

In this experiment, the similarity threshold, set empirically to 0.85, guarantees the highest alignment quality achieved on average in all test cases.

5.3. Experimental Results. When performing testing cases from the Bibliographic track, we compare the result of our suggestion with OAEI's participants, i.e., AML, edna, and LogMapLt from the standpoint of f -measure. Figure 5 shows the experimental results for all types of testing cases in the Bibliographic track of OAEI.

As can be seen from the presentation information in Figure 5, for most testing cases, our proposal outperformed other methods due to the application of a Debate Mechanism that integrates the advantages of various basic similarity measures; the matching problem can be considered synthetically from different angles in ontology matching. In some cases, the f -measure for all matching techniques is nearly zero, which is due to the complexity of testing cases. By contrast, our proposal does a better job in these cases as listed in Table 4, which masks a small step forward in our proposal on the basis of cutting-edge work.

When performing sensor ontology matching tasks, we compare the result of our proposal with four basic EMM that

TABLE 1: Classification method of correspondences.

k_a	Category
k	C_1
$((k/2), k)$	C_2
$(k/2)$	C_3
$(0, (k/2))$	C_4
0	C_5

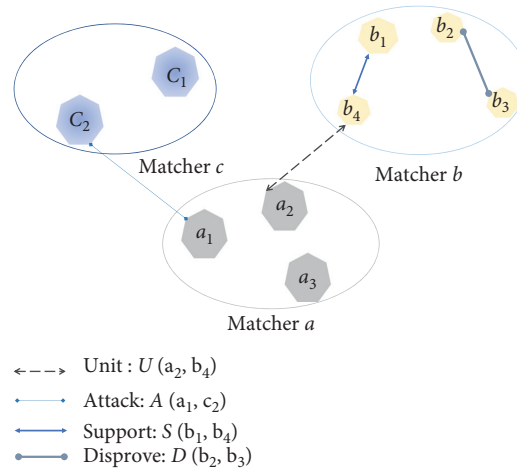


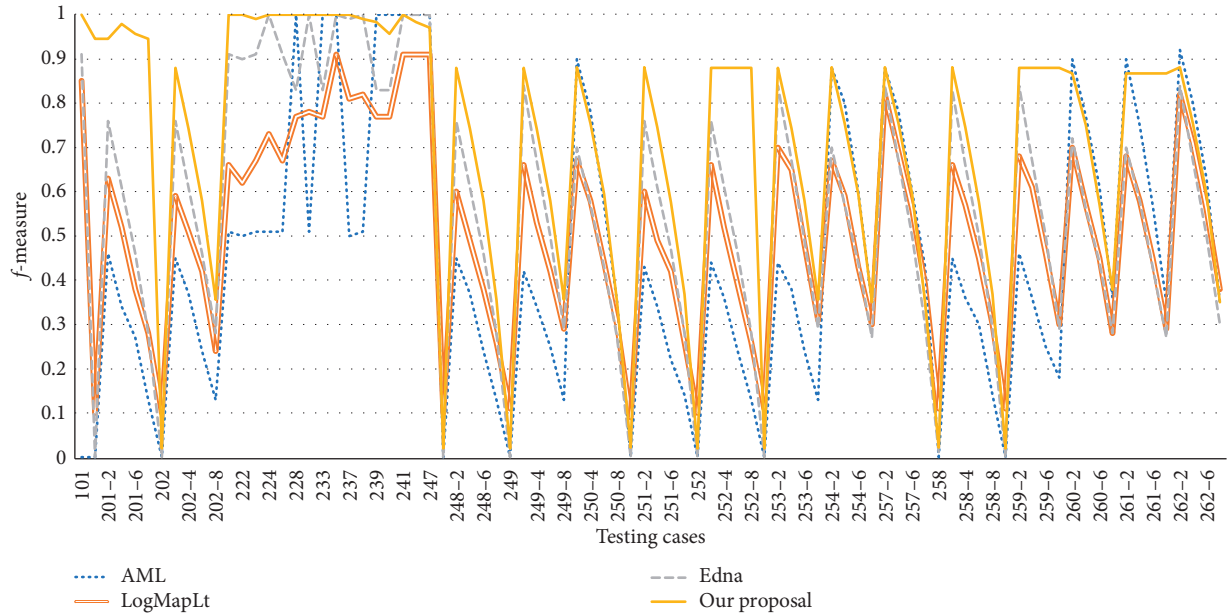
FIGURE 4: Four relationships between arguments.

TABLE 2: Descriptions on OAEP's Bibliographic track.

Testing case	Description
101	Two identical ontologies
201–202	Ontologies varying in terminology and semantics characteristics
221–247	Ontologies varying in structure characteristics
248–262	Ontologies varying in terminology, semantics, and structure characteristics

TABLE 3: Descriptions on sensor ontologies.

Sensor ontology	Ontology scale	Description
Semantic sensor network ontology (SSN)	55 entities	It is about sensors, actuators and observations, and related concepts
Sensor, observation, sample, and actuator ontology (SOSA)	42 entities	It defines those common classes and attributes whose data can be securely exchanged in SSN, its modules, and all SOSA uses
IoT-lite ontology (IoT)	40 entities	It is about key concepts of IoT
SensorOntology2009 ontology (SN)	152 entities	It is the initial version of the SSN ontology, which was developed in 2009
Original semantic sensor network ontology (OSSN)	107 entities	It is an original version of SSN, which was developed in 2009–2011

FIGURE 5: Comparison with OAIE's participants in terms of f -measure.TABLE 4: Comparison with OAIE's participants in terms of f -measure.

Testing case	AML	Edna	LogMapLt	Our proposal
202-8	0.13	0.24	0.28	0.3553
248-8	0.13	0.26	0.28	0.3553
249-8	0.13	0.29	0.29	0.3553
251-8	0.14	0.26	0.28	0.3687
253-6	0.24	0.48	0.49	0.5794
253-8	0.13	0.31	0.29	0.3553
258-8	0.14	0.3	0.29	0.3687
260-8	0.36	0.28	0.28	0.3779

are mentioned above, i.e., Levenshtein-similarity, Jaro similarity metric, WordNet-based distances, and similarity flooding in terms of recall, precision, and f -measure.

Figure 6 depicts the result of our scheme for matching six pairs of real sensor ontologies and compares them with four basic EMMs, and the results show our proposal typically achieves very high capacity with the golden alignment. Furthermore, the application of fuzzy measure extends the

single-dimensional evaluation on similarities judging by basic similarity measures to three-dimensional assessment, which fully express the similarity to gain a high-quality alignment.

In a word, FDM can significantly improve the accuracy of search results, and at the same time ensure a high recall rate, in all kinds of matching tasks that are superior to other competitors.

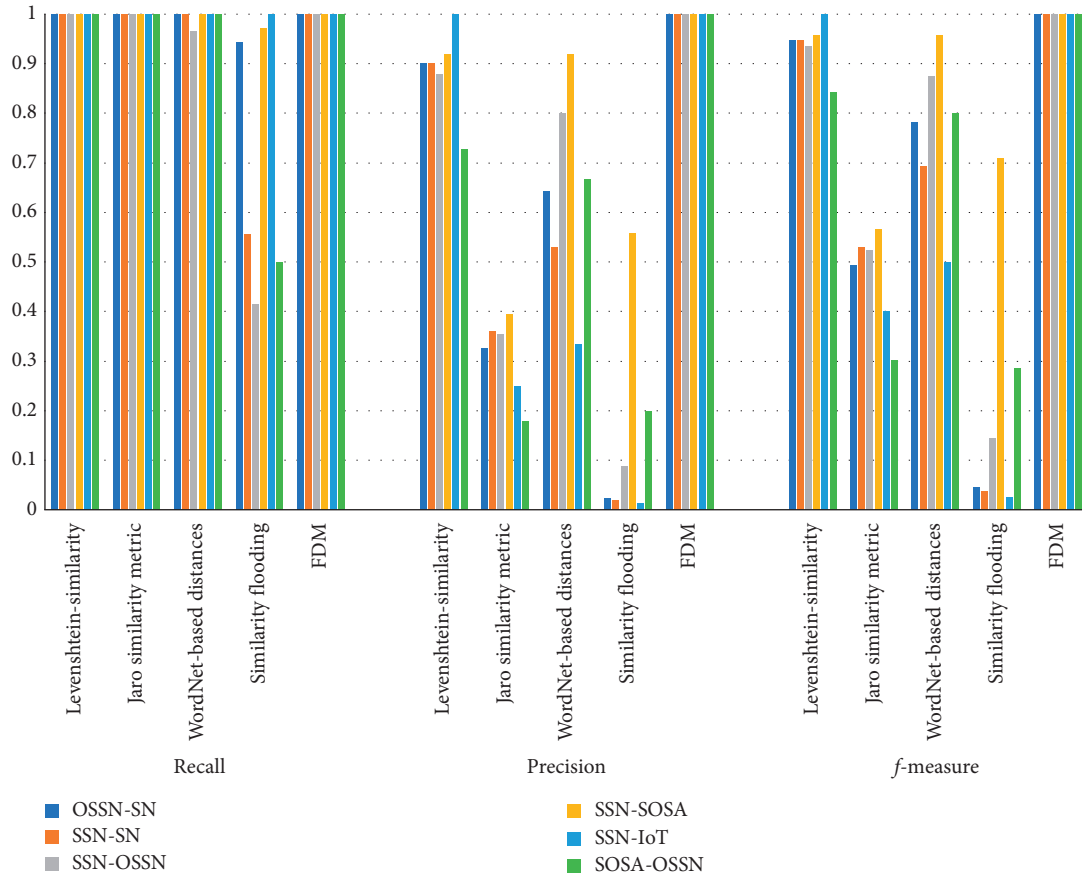


FIGURE 6: Comparison on two pairs of real sensor ontologies with four matchers.

6. Conclusions

Semantic connections among different sensor ontologies are of great significance to Nx-IIoT's communication quality and information security [50]. Therefore, a sensor ontology aggregating method based on Fuzzy Debate Mechanism is proposed, which extracts the ultimate alignment by performing arguments between different entity matching measures. A fuzzy similarity measure is presented to improve the alignment's quality, which models two entities' similarity in the vector space and their semantic distance is calculated by using cosine function. The Bibliographic tracks provided by OAEI and five real sensor ontologies were used to calculate the performance of the proposed method in this experiment. Compared with the most advanced ontology matching technology and four basic ontology matchers, the robustness and effectiveness of our proposal are verified.

Looking to the future, there are two challenges in sensor ontology alignment extraction technique: one is to measure entity similarity, and the other is how to develop extraction rules to tune the quality of alignment. In one hand, we need to further innovate the similarity measurement of domain-specific ontology to adapt to its fine-grained and complex structure. In the other hand, there is the need of approaches that can deal with the problem of uncertainty generated in the matching process. [50].

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest in the work.

Acknowledgments

This work was supported by the Natural Science Foundation of Fujian Province (No. 2020J01875) and the National Natural Science Foundation of China (Nos. 61773415, 61801527, and 61103143).

References

- [1] H. Liu, Y. Wang, and N. Fan, "A hybrid deep grouping algorithm for large scale global optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 6, pp. 1112–1124, 2020.
- [2] J. Pan, P. Song, S. Chu et al., "Improved compact cuckoo search algorithm applied to location of drone logistics hub," *Mathematics*, vol. 8, no. 3, pp. 1–19, 2020.
- [3] J. Xiong, H. Liu, B. Jin et al., "A lightweight privacy protection scheme based on user preference in mobile crowdsensing,"

- Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 6, pp. 1–16, 2020.
- [4] J. Xiong, M. Zhao, M. Bhuiyan et al., “An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.
 - [5] J. Lin, G. Srivastava, Y. Zhang et al., “Privacy preserving multi-objective sanitization model in 6G IoT environments,” *IEEE Internet of Things Journal*, vol. 8, pp. 5340–5349, 2020.
 - [6] D. Bunker, L. Levine, and C. Woody, “Repertoires of collaboration for common operating pictures of disasters and extreme events,” *Information Systems Frontiers*, vol. 17, no. 1, pp. 51–65, 2015.
 - [7] T. Wu, T. Wang, Y. Lee et al., “Improved authenticated key agreement scheme for fog-driven IoT healthcare system,” *Security and Communication Networks*, vol. 2021, Article ID 6658041, 16 pages, 2021.
 - [8] F. Alamdar, M. Kalantari, and A. Rajabifard, “Towards multi-agency sensor information integration for disaster management,” *Computers, Environment and Urban Systems*, vol. 56, pp. 68–85, 2016.
 - [9] E. U. Ogbodo, D. Dorrell, and A. M. Abu-Mahfouz, “Cognitive radio based sensor network in smart grid: architectures, applications and communication technologies,” *IEEE Access*, vol. 5, pp. 19084–19098, 2017.
 - [10] C. Michael, B. Payam, B. Luis et al., “The ssn ontology of the w3c semantic sensor network incubator group,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25–32, 2012.
 - [11] M. Ganzha, M. Paprzycki, W. Pawłowski et al., “Semantic interoperability in the Internet of Things: an overview from the INTER-IoT perspective,” *Journal of Network and Computer Applications*, vol. 81, pp. 111–124, 2017.
 - [12] K. Janowicz, A. Haller, S. J. D. Cox et al., “SOSA: a lightweight ontology for sensors, observations, samples, and actuators,” *Journal of Web Semantics*, vol. 56, pp. 1–10, 2019.
 - [13] P. Barnaghi, W. Wang, C. Henson et al., “Semantics for the Internet of Things: early progress and back to the future,” *International Journal on Semantic Web and Information Systems*, vol. 8, no. 1, pp. 1–21, 2012.
 - [14] X. Xue, C. Yang, C. Jiang et al., “Optimizing ontology alignment through linkage learning on entity correspondences,” *Complexity*, vol. 2021, Article ID 5574732, 12 pages, 2021.
 - [15] X. Xue, H. Yang, J. Zhang et al., “An automatic biomedical ontology meta-matching technique,” *Journal of Network Intelligence*, vol. 4, no. 3, pp. 109–113, 2019.
 - [16] N. Alboukaey and A. Joukhar, “Ontology matching as regression problem,” *Journal of Digital Information Management*, vol. 16, no. 1, pp. 34–42, 2018.
 - [17] M. A. Khoudja, M. Fareh, and H. Bouarfa, “Ontology matching using neural networks: survey and analysis,” in *Proceedings of the 2018 International Conference on Applied Smart Systems (ICASS)*, pp. 1–6, Medea, Algeria, November, 2018.
 - [18] M. T. Dhoubib, C. F. Zucker, and A. G. B. Tettamanzi, “An ontology alignment approach combining word embedding and the radius measure,” in *Proceedings of the International Conference on Semantic Systems*, pp. 191–197, Karlsruhe, Germany, September, 2019.
 - [19] A. Ali, M. Hamid, K. Ahmad et al., “Context aware instance matching through graph embedding in lexical semantic space,” *Knowledge-Based Systems*, vol. 186, pp. 422–433, 2019.
 - [20] F. Ali, K. Kwak, and Y. Kim, “Opinion mining based on fuzzy domain ontology and support vector machine: a proposal to automate online review classification,” *Applied Soft Computing*, vol. 47, pp. 235–250, 2016.
 - [21] A. Siham, M. Sihem, and F. Muhammad, “Decision trees in automatic ontology matching,” *International Journal of Metadata, Semantics and Ontologies*, vol. 11, no. 3, pp. 180–190, 2016.
 - [22] G. Acampora, P. Avella, V. Loia et al., “Improving ontology alignment through memtic algorithms,” in *Proceedings of the 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pp. 240–259, Taipei, Taiwan, June, 2010.
 - [23] M. G. Jorge, A. Enrique, and F. A. M. José, “Optimizing ontology alignments by using genetic algorithms,” in *Proceedings of the workshop on nature based reasoning for the semantic Web*, pp. 1–15, Karlsruhe, Germany, October, 2008.
 - [24] M. G. Jorge and F. A. M. Jose, “Evaluation of two heuristic approaches to solve the ontology meta-matching problem,” *Knowledge and Information Systems*, vol. 26, no. 2, pp. 225–247, 2011.
 - [25] A. L. Ginsca and I. Adrian, “Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment,” in *Proceedings of the 9th RoEduNet IEEE International Conference*, pp. 118–122, Sibiu, Romania, June, 2010.
 - [26] Z. Xu, W. Zhang, T. Zhang et al., “HRCNet: high-resolution context extraction network for semantic segmentation of remote sensing images,” *Remote Sensing*, vol. 13, no. 71, pp. 1–23, 2021.
 - [27] X. Xue and J. Chen, “Optimizing sensor ontology alignment through Compact co-firefly algorithm,” *Sensors*, vol. 20, no. 7, pp. 1–15, 2020.
 - [28] X. Xue and J. Zhang, “Matching large-scale biomedical ontologies with central concept based partitioning algorithm and adaptive Compact evolutionary algorithm,” *Applied Soft Computing*, vol. 106, p. 107343, 2021.
 - [29] X. Xue, X. Wu, C. Jiang et al., “Integrating sensor ontologies with global and local alignment extractions,” *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–10, Article ID 6625184, 2021.
 - [30] V. Franzoni and A. Milani, “Semantic context extraction from collaborative networks,” in *Proceedings of the 2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 131–136, Calabria, Italy, May, 2015.
 - [31] P. Buitelaar and M. Sintek, “Ontolt version 1.0: middleware for ontology extraction from text,” in *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, pp. 1–4, Hiroshima, Japan, November 2004.
 - [32] M. Gaeta, F. Orciuoli, S. Paolozzi et al., “Ontology extraction for knowledge reuse: the e-learning perspective,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 4, pp. 798–809, 2011.
 - [33] L. Laera, V. Tamma, J. Euzenat et al., “Reaching agreement over ontology alignments,” in *Proceedings of the International Semantic Web Conference*, pp. 371–384, Athens, GA, USA, November 2006.
 - [34] L. Laera, I. Blacoe, V. Tamma et al., “Argumentation over ontology correspondences in mas,” in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pp. 1–8, Honolulu, HI, USA, May, 2007.
 - [35] C. T. dos Santos and J. Euzenat, “Consistency-driven argumentation for alignment agreement,” in *Proceedings of the 5th*

- ISWC workshop on ontology matching (OM)*, pp. 37–48, Shanghai, China, November 2010.
- [36] S. Fernandez, I. Marsa-Maestre, J. R. Velasco et al., “Ontology alignment architecture for semantic sensor web integration,” *Sensors*, vol. 13, no. 9, pp. 12581–12604, 2013.
- [37] K. Todorov, P. Geibel, and C. Hudelot, “A framework for a fuzzy matching between multiple domain ontologies,” in *Proceedings of the International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 538–547, Kaiserslautern, Germany, September, 2011.
- [38] K. Todorov, C. Hudelot, A. Popescu et al., “Fuzzy ontology alignment using background knowledge,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 22, no. 1, pp. 75–112, 2014.
- [39] Y. Zhang, A. Panangadan, and V. K. Prasanna, “UFOM: unified fuzzy ontology matching,” in *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, pp. 787–794, Redwood City, CA, USA, August 2014.
- [40] V. Cross, “Fuzzy semantic distance measures between ontological concepts,” in *Proceedings of the IEEE Annual Meeting of the Fuzzy Information*, vol. 2, pp. 635–640, Banff, AB, Canada, June 2004.
- [41] H. Neuhaus and M. Compton, “The semantic sensor network ontology,” in *Proceedings of the 12th AGILE International Conference on Geographic Information Science: Pre-Conference Workshop Challenges in Geospatial Data Harmonisation*, pp. 1–33, Hannover, Germany, June 2009.
- [42] K. Taylor, A. Haller, M. Lefrançois et al., “The semantic sensor network ontology, revamped,” in *Proceedings of the 18th International Semantic Web Conference*, New Zealand, October 2019.
- [43] M. Diaz, C. Martín, and B. Rubio, “State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing,” *Journal of Network and Computer Applications*, vol. 67, pp. 99–117, 2016.
- [44] R. Ronald and Yager, “Uncertainty modeling using fuzzy measures,” *Knowledge-Based Systems*, vol. 92, pp. 1–8, 2016.
- [45] J. Euzenat and P. Shvaiko, *Ontology Matching*, Springer, Heidelberg, Germany, 2007.
- [46] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, “A comparison of string distance metrics for name-matching tasks,” in *Proceedings of the 2003 International Conference on Information Integration on the Web (IIWeb)*, pp. 73–78, Acapulco, Mexico, August 2003.
- [47] G. A. Miller, “WordNet: a lexical database for English,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [48] S. Melnik, H. Garcia-Molina, and E. Rahm, “Similarity flooding: a versatile graph matching algorithm and its application to schema matching,” in *Proceedings of 18th International Conference on Data Engineering*, pp. 117–128, San Jose, CA, USA, March 2002.
- [49] E. Jerome, “Semantic precision and recall for ontology alignment evaluation,” in *Proceedings of the International Joint Conferences on Artificial Intelligence*, vol. 7, pp. 348–353, Hyderabad, India, January 2007.
- [50] T. Wu, L. Yang, Z. Lee et al., “Improved ECC-based three-factor multiserver authentication scheme t,” *Security and Communication Networks*, vol. 2021, pp. 1–14, Article ID 6627956, 2021.

Research Article

PUF-Based Mutual-Authenticated Key Distribution for Dynamic Sensor Networks

Yanan Liu ¹, Yijun Cui,² Lein Harn,³ Zheng Zhang ¹, Hao Yan,¹ Yuan Cheng,¹
and Shuo Qiu ¹

¹School of Network Security, Jinling Institute of Technology, Nanjing 211169, China

²College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211100, China

³Department of Computer Science Electrical Engineering, University of Missouri, Kansas City 64110, MO, USA

Correspondence should be addressed to Zheng Zhang; zhangzheng@jit.edu.cn

Received 5 February 2021; Accepted 22 April 2021; Published 3 May 2021

Academic Editor: Qing Yang

Copyright © 2021 Yanan Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Because of the movements of sensor nodes and unknown mobility pattern, how to ensure two communicating (static or mobile) nodes authenticate and share a pairwise key is important. In this paper, we propose a mutual-authenticated key distribution scheme based on physical unclonable functions (PUFs) for dynamic sensor networks. Compared with traditional key predistribution schemes, the proposal reduces the storage overhead and the key exposure risks and thereby improves the resilience against node capture attacks. Mutual authentication is provided by the PUF challenge-response mechanism. However, the PUF response is not transmitted in plain forms so as to resist the modelling attacks, which is vulnerable in some existing PUF-based schemes. We demonstrate the proposed scheme to improve the secure connectivity and other performances by analysis and experiments.

1. Introduction

Many applications of wireless sensor networks (WSNs) are working in hostile battlefield environments or unmanned areas with poor conditions. Sensor nodes and wireless channels are vulnerable to malicious attacks, such as physical capture nodes, data tampering, and side channel attacks [1–3]. Data encryption is a crucial technology to ensure secure communication between the cloud and end-devices [4–6]. The authentication and key distribution are the premise and foundation [7, 8].

In 2002, Eschenauer and Gligor proposed a random key predistribution scheme [9] for the resource limited sensors. In 2007, Du et al. applied Eschenauer's scheme into hierarchical sensor networks and proposed an asymmetric key predistribution scheme (AP) [10]. This kind of “probabilistic” schemes had low computation and communication overhead but cannot ensure that any two of communicating nodes share a pairwise key. Besides, the key storage amount showed a tradeoff between the network connectivity and

resilience against node capture attacks. In 2009, Boujelben proposed a key management scheme based on the Blom matrix [11] to improve the resilience against node capture; however, the computation cost for matrix operation was too complicated for common sensors [12]. In terms of public key algorithms, in 2012, Benamar et al. [13] proposed a dynamic security key management model for hierarchical sensor networks based on public key infrastructure (PKI). In 2015, Lee and Kim [14] proposed a key renewal scheme with sensor authentication under clustered wireless sensor networks based on modular exponentiation which was similar to the Diffie–Hellman key exchange. These schemes increased the connectivity; however, the public key computational overhead was too large for sensors. In 2010, Han et al. [15] proposed an approach for dynamic node authentication and key exchange, which reduces the overhead of mobile node reauthentication. Each sink node authenticates other neighboring sink and sensor nodes and supports reauthentication with less communication and computation overhead. In 2015, Erfani et al. [16] proposed a

key management scheme, which used key predistribution and postdeployment key establishment mechanisms for dynamic sensor networks. The predistributed keys are loaded to the memory of sensor nodes before network deployment, and after that, some postdeployment keys are generated and stored in each sensor node. In Erfani's approach, the base station is involved in intracluster authentication and key distribution, which costs too much communication overheads. In 2020, Tian et al. [17] proposed a blockchain-based secure key management scheme with trustworthiness in dynamic wireless sensor networks, which designed a secure cluster formation algorithm and a secure node movement algorithm to implement key management.

This paper proposed a mutual authenticated key distribution scheme based on physical unclonable functions (PUFs) in dynamic sensor networks, so as to help the sink node to authenticate and distribute session keys to the static and mobile sensors. Lightweight mutual authentication is guaranteed by a challenge-response mechanism based on the PUF. To address the PUF challenge-response pairs (CRPs) exposure problem, the CRPs are not transmitted as plaintext in order to resist the modelling attack to PUF. In addition, sensors are not required to prestore any keys in memory, which not only saves the storage overhead but also improves the resilience against sensor node capture attacks.

2. Physical Unclonable Function (PUF)

2.1. Review of PUFs. Physical unclonable function (PUF) is a new encryption component that can extract random differences introduced by inconsistencies in manufacturing processes between gate circuits or connection lines (wires) in integrated circuits (IC). These random differences can be used to generate an encrypted (response) signal with certain rules [18]. Random differences in a physical object can be interpreted as the unique "fingerprint" of a hardware instant. In addition to IC PUFs [19], there are silicon PUFs [20], coated PUFs [21], and so on. We use a one-way mapping function P to describe PUF, which can be expressed as

$$P: C \longrightarrow R: P(c) = r, \quad c \in C, r \in R. \quad (1)$$

The functional mapping between input c and output r is instance-specific and unpredictable prior to the actual fabrication of the circuit. When an electrical stimulus is applied to the structure, it reacts in an unpredictable (but instance-wise repeatable) manner due to the complex interaction of the stimulus with the physical microstructure of the device. The exact nature of this microstructure depends on physical factors introduced during manufacturing. The applied stimulus is considered as the "challenge," while the reaction generated by the PUF is considered as the "response." A specific challenge and its response together form a challenge-response pair (CRP) (c, r) , and the CRP dataset acts as a unique fingerprint for the instance.

The attractive features of PUFs are light-weightness, unpredictability, unclonability, and uniqueness, which make PUFs valuable in designing ultralightweight authentication, key generation, and other security protocols [22, 23]. Device

authentication is the process that an authenticator verifies the identity of a device client before communication. PUF CRP can be implemented in the challenge-response authentication mechanism. The authenticator creates a CRP database that stores all the challenges and their expected responses from registered clients. To verify the identity of a client, the authenticator first selects a challenge from the database and sends it to the client. The client generates a response to the challenge using its on-board PUF and provides it to the authenticator. By comparing the current client's response against the one stored in the CRP database, the authenticator infers whether the client is trusted or not.

This new type of schemes speeds up the authentication process and also lightens the key storage and thereby reduces key exposure risk. A PUF with a large enough challenge space to make exhaustive enumeration of its CRP set infeasible is termed a strong PUF and is the PUFs of choice in most practical security applications. We keep ourselves confined to strong PUFs in this work. Since the assessment of a PUF implies a physical measurement, it is very susceptible to circuit noise. Hence, to make it reliable and to have full entropy, [22] had proposed an error correction circuit with a very low hardware overhead to reduce the fuzziness of the PUF's responses and make it more robust and reliable. However, in our work, we consider each PUF structure as a black-box challenge-response system, where a set of challenges are available and the system responds with a set of sufficiently different responses.

In 2015, Allam proposed a scheme that depends on the physical layer mechanisms, which consist of PUF and Channel Status Information (CSI) for providing point-to-point real-time hardware-based authentication technique between two parties communicating directly through wireless media and effective key exchange to assure an authenticated secure channel between them [23]. In 2013, Bahrapour and Atani proposed a Key Management Protocol for Wireless Sensor Networks based on PUFs, in which the PUFs were used to design the public keys [24]. In 2017, Chatterjee et al. proposed a PUF-based secure communication protocol for PUF [25]. The PUF was used to generate the public key based on the bilinear pairing of each device in the key agreement protocol. In 2018, Braeken improved Chatterjee's protocol efficiency by way of employing the Elliptic Curve Qu Vanstone (ECQV) [26]. In 2019, Li et al. proposed a PUF-based secure communication system for the Internet of Things [27]. In 2020, Zhang et al. proposed a PUF-based Key Distribution in Wireless Sensor Networks [28].

2.2. Configurable RO PUFs. The PUF circuit, which is the core of authentication and key distribution in our scheme, should be easily implemented on the FPGA with good uniqueness and reliability. In our previous work, several types of configurable RO PUF are proposed, including MUX based RRO PUF in [29], XOR gate based XCRO PUF in [30], and tristate configurable TCRO PUF in [31]. In this paper, the MUX based RRO PUF is chosen. The MUX based configurable RO (CRO) PUF was first introduced in [32],

where each ring oscillator can be reconfigured by using a multiplexer to select one of two inverters that are connected to the multiplexer to form an RO. Our reconfigurable RRO design, as shown in Figure 1, is consisted of a chain of inverter delay units and an AND gate delay unit. When the configurable signal of a MUX is “0,” the upper path will be chosen. On the contrary, when the signal is “1,” the lower path will be chosen to construct the RO structure. The configure procedure extracts the transfer difference of each MUX and the delay of the upper and lower path.

2.3. Implement of PUFs. The PUF used in our approach is implemented and studied based on Xilinx SoC FPGAs and will be applied to real-world scenarios based on ASIC or SoC FPGA including ARM core (e.g., Xilinx Zynq-7000 series, Altera SoC or Microsemi Smart Fusion2) after validation. As shown in Figure 2, the main components include MUX, XOR gate, inverters, and AND gate. In the implementation of the RRO PUF, the primitive MUXF7 is chosen for the multiplexer, the primitive LUT1 is adopted for the inverter, and LUT2 is utilized for the AND gate. Eight delay units that include seven inverter delay units and one AND gate delay unit are included in the single RRO array. Each delay unit occupies one slice and two delay units can be implemented in one configurable logic block (CLB). Therefore, four CLBs are needed to implement one RRO PUF array. In order to make sure that all RROs are identically routed, they are created as hard macros to avoid the bias introduced in the placement and routing. The detailed design can be referred in authors’ previous work [29].

3. PUF-Based Mutual Authentication and Key Distribution

3.1. Network Model. Large-scale wireless sensor networks are usually deployed in a hierarchical clustered structure and contain heterogeneous nodes, such as a base station (BS), several sink nodes (SN), and a number of low-energy sensors. BS is assumed to be resourceful and global trusted. It manages the entire network and stores all gathered information by sensor nodes. Sink node is assumed to have higher hardware configurations than sensors, including memory, communication, and computation ability. A sink node acts as a gateway between sensor nodes and BS. Sensors are divided into nonoverlapping clusters; they collect data from surroundings and send raw data to the sink node. Sensor nodes are assumed to have a random linear movement pattern, while the BS and sink nodes are static like Han and Erfani schemes [15, 16]. Because of unpredictable position of mobile sensors, how to ensure a sink node to authenticate and distribute a pairwise key to every present cluster-member sensor is difficult.

In our network model, assume there are n sensors, named $S_{0,\dots,n-1}$, and m sink nodes, named $SN_{0,\dots,m-1}$. Each sensor node has a unique ID S_i and embeds a chip with a PUF structure, denoted as P_{S_i} . Before network deployment, all the nodes are divided into m deployment groups (DGs), denoted as $\{DG_i\}_{i=0,\dots,m-1}$. In each DG, there is 1 SN and

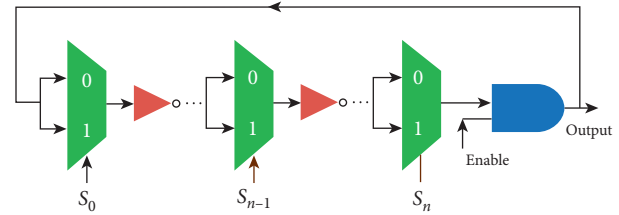


FIGURE 1: RRO PUF structure.

$d = n/m$ sensors, and the SN is called the “Home-SN” of these d sensors. Nodes in a DG will be thrown into the destination area together, so as to form a cluster. Figure 3 gives an example with 3 DGs and 9 sensors.

3.2. Initialization and Network Deployment. Before network deployment, for each sensor S_i , take a random challenge number c_{S_i} as the input of PUF P_{S_i} and get the output response r_{S_i} ; prestore the PUF CRP (c_{S_i}, r_{S_i}) to the Home-SN of S_i by indexing with the sensor ID S_i . For example, in Figure 3, in DG0, take the sink node SN0 as the Home-SN of sensors S_0, S_1 , and S_2 . Generate a CRP for each sensor as (c_{S_0}, r_{S_0}) , (c_{S_1}, r_{S_1}) , and (c_{S_2}, r_{S_2}) and save them into the memory of SN0.

After network deployment, the sink node launches the cluster forming process (not discussed in this paper, please refer to [33]), which divides all sensor nodes into clusters with no cross coverage. Each cluster includes a sink node, which is called the “cluster head” (CH), and n/m sensors, which are called the “cluster members” (CM). Nodes in the same DG form a cluster with very high probability since they are thrown close to each other. It shows an ideal deployment example in Figure 4.

In order to ensure the secure intracluster communication, a sink node needs to authenticate and distributes a pairwise key to every cluster-member sensor. In a short period after network deployment, assume sensors are static. It is easy for the sink node to run the authentication and key distribution according to the challenge-response mechanism based on PUF CRP. However, after some working time, a sensor moves into another cluster’s region (as shown in Figure 5), in which the sink node does not share the PUF CRP of the mobile sensor. In this situation, the sink node in the present cluster, called the “Present-SN,” should authenticate the mobile sensor via the help of the Home-SN. In the following section, we will describe our approach by two subschemes for static sensors and mobile sensors, respectively.

The differences between these two subschemes mainly happened in the following aspects: (1) there were two entities in static subscheme: Home-SN and the sensor; there were three entities in mobile subscheme: Home-SN, Present-SN, and the sensor; (2) in the static subscheme, the (Present also Home) SN generated the session key with the sensor; in the mobile subscheme, the Home-SN generated the session key between the Present-SN and the sensor; (3) in the static subscheme, the (Present also Home) SN authenticated the sensor directly; in the mobile subscheme, the Home SN helped the Present-SN to authenticate the sensor.

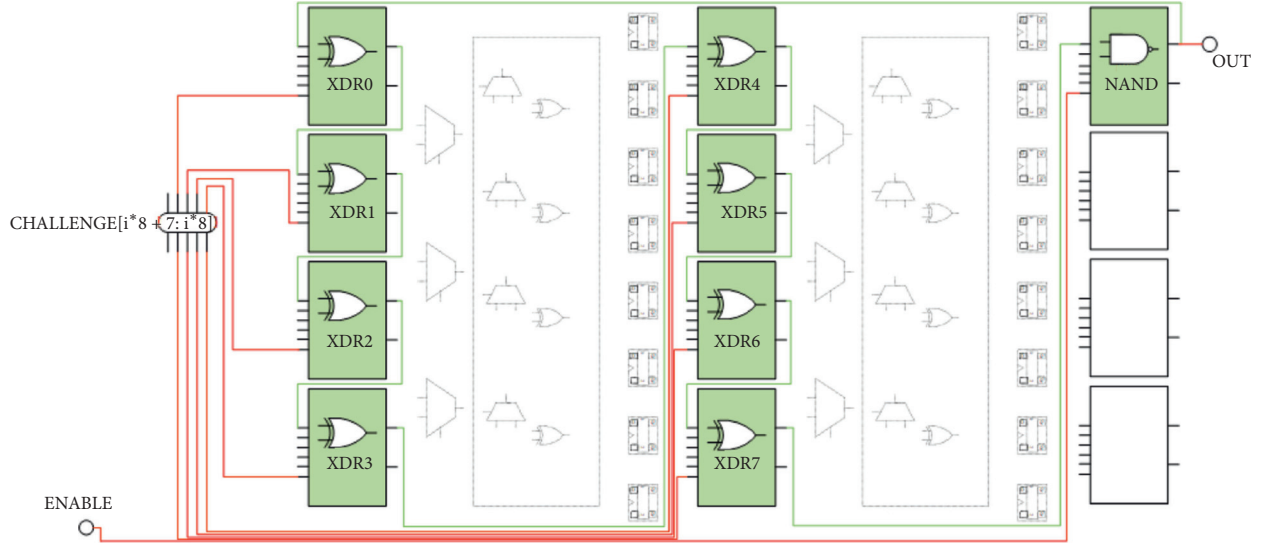


FIGURE 2: Implementation of an RRO in a CLB.

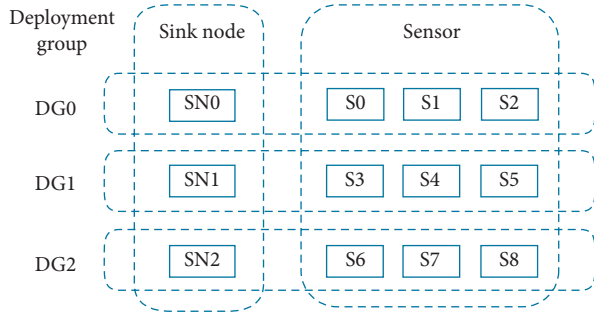


FIGURE 3: An example of the deployment model with 3 sink nodes and 9 sensors.

3.3. *Static Sensors Subscheme.* The approach of a sink node SN_0 authenticating and distributing a pairwise key to a static sensor S_0 is described as shown in Figure 6.

- (1) After network deployment and clustering process, in the cluster C_0 , the sink node SN_0 detects a sensor S_0 in its cluster. SN_0 reads a PUF CRP in its memory: (c_{S_0}, r_{S_0}) by indexing of id_{S_0} .
- (2) SN_0 computes a temporary key key_{SN_0} :

$$key_{SN_0} = H(\|r_{S_0} \text{timestamp}1), \quad (2)$$

where H is a hash function.

SN_0 generates a session key $key_{SN_0-S_0}$ and encrypts it by key_{SN_0} to get cipher1:

$$\text{cipher1} = E(key_{SN_0}, key_{SN_0-S_0}). \quad (3)$$

E is symmetric encryption (e.g., AES). Then, SN_0 encrypts c_{S_0} by using $key_{SN_0-S_0}$:

$$\text{cipher2} = E(key_{SN_0-S_0}, \|c_{S_0} \text{timestamp}1). \quad (4)$$

Then, SN_0 generates a secret random number $\text{nonce}1$ and encrypts it by using $key_{SN_0-S_0}$:

$$\text{cipher3} = E(key_{SN_0-S_0}, \text{nonce}1). \quad (5)$$

SN_0 sends the challenge c_{S_0} , cipher1, cipher2, and cipher3 to S_0 :

$$SN_0 \rightarrow S_0: \|c_{S_0} \| \text{cipher1} \| \text{cipher2} \| \text{cipher3} \text{timestamp}1. \quad (6)$$

- (3) After receiving the message, the sensor S_0 firstly inputs c_{S_0} into the PUF structure P_{S_0} , which is embedded during the initialization phase, and gets the output response r_0 :

$$r_0 = P_{S_0}(c_{S_0}). \quad (7)$$

S_0 computes a temporary key, key_{S_0} :

$$key_{S_0} = H(\|r_0 \text{timestamp}1). \quad (8)$$

Then, S_0 decrypts the cipher1 to get the pairwise key, $key_{S_0-SN_0}$:

$$\begin{aligned} \text{plain1} &= D(key_{S_0}, \text{cipher1}) \\ &= D(key_{S_0}, E(key_{SN_0}, key_{SN_0-S_0})) = key_{S_0-SN_0}. \end{aligned} \quad (9)$$

The function D is the decryption operation of E .

S_0 decrypts cipher2 by using $key_{S_0-SN_0}$ and gets plain2:

$$\begin{aligned} \text{plain2} &= D(key_{S_0-SN_0}, \text{cipher2}) \\ &= D(key_{S_0-SN_0}, E(key_{SN_0-S_0}, \|c_{S_0} \text{timestamp}1)). \end{aligned} \quad (10)$$

The sensor S_0 checks if the equation $\text{plain2} = \{c_{S_0} \| \text{timestamp}1\}$ is correct.

If not, S_0 deduces that the sink node SN_0 is not its valid Home-SN, since it does not share a correct PUF

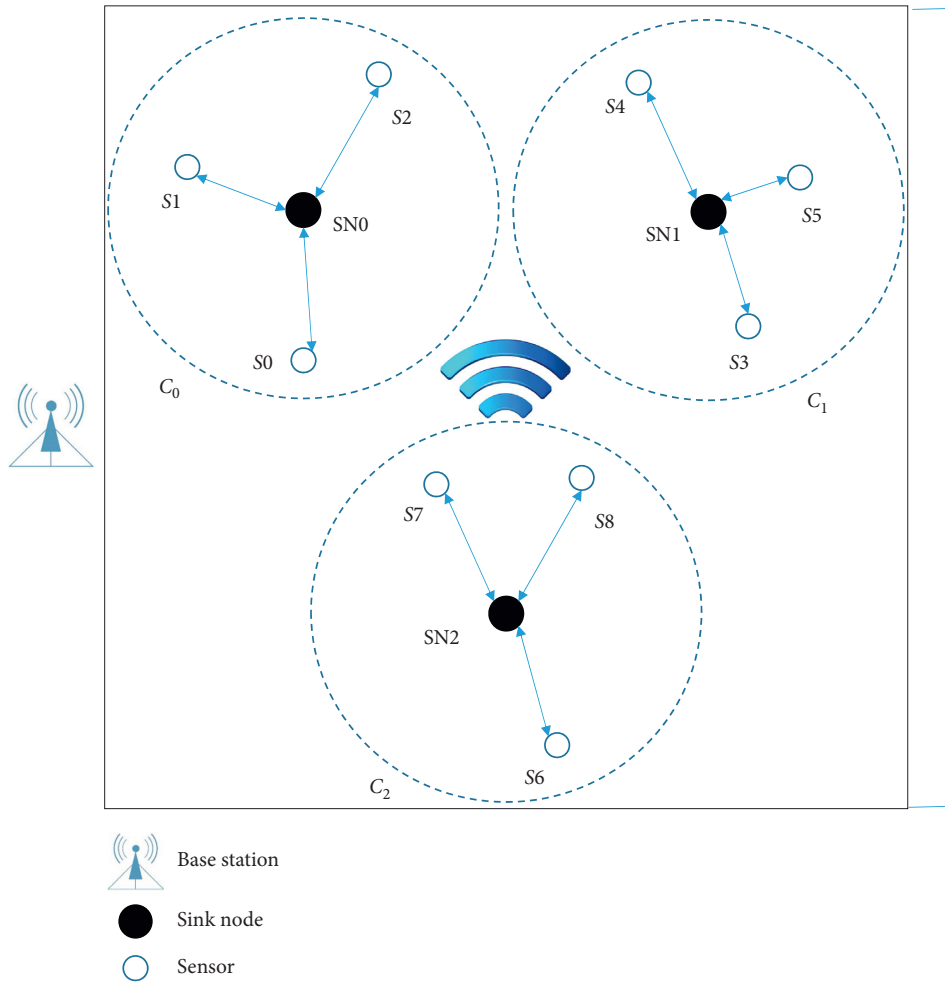


FIGURE 4: The network deployment.

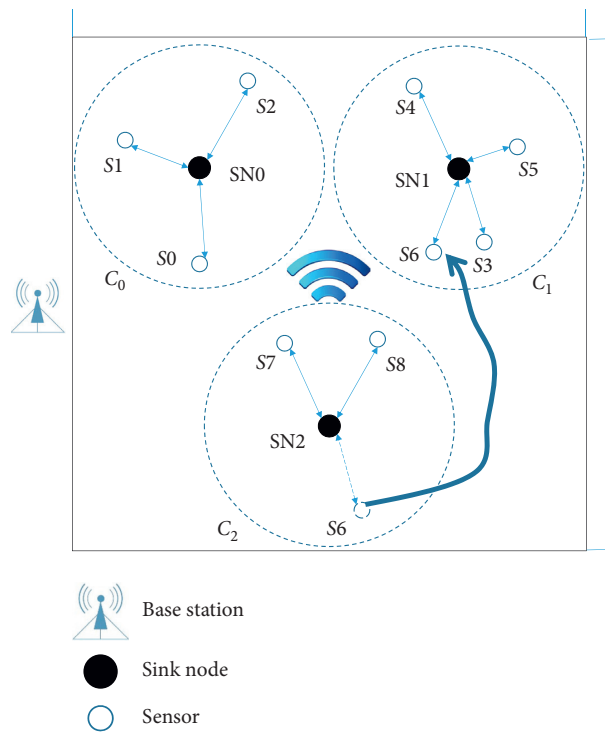


FIGURE 5: S_6 moves from C_2 into C_1 .

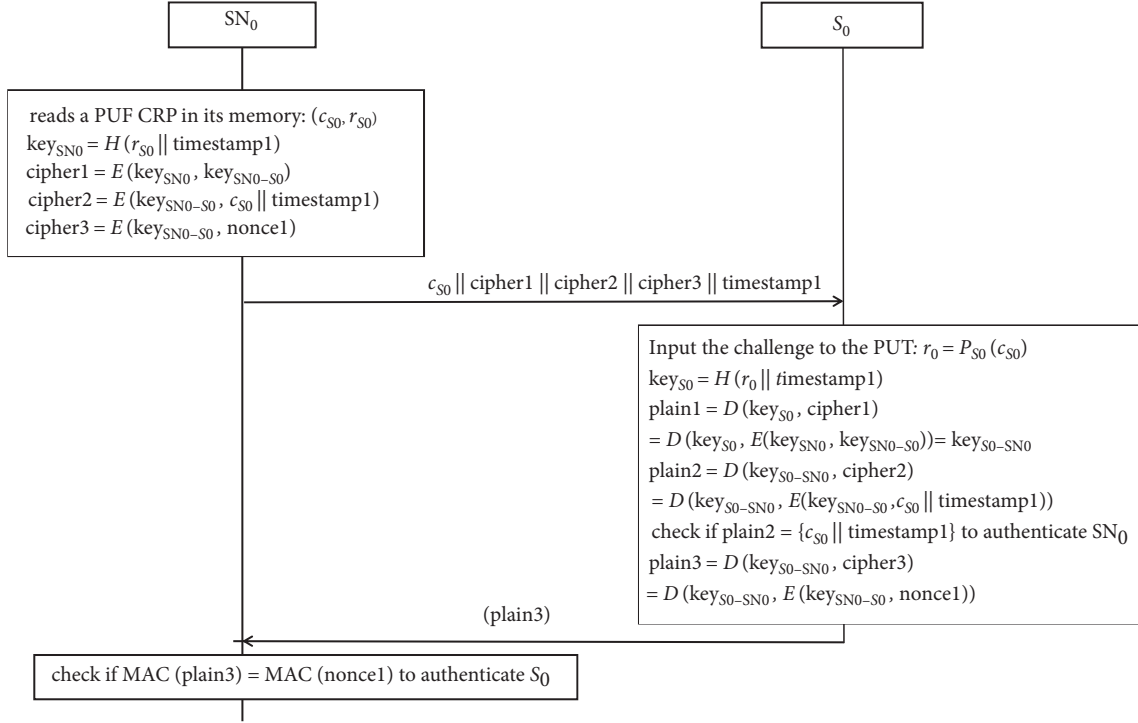


FIGURE 6: Authentication and key distribution between SN₀ and static sensor S₀.

CRP of SN₀ (c_{S0}, r_{S0}). The SN₀ fails the authentication by S₀ and the scheme quits.

If correct, S₀ infers that key_{S0-SN0} = key_{SN0-S0}; then key_{S0} equals key_{SN0}, and r₀ equals r_{S0}. This means the sink node SN₀ indeed shares a CRP (c_{S0}, r_{S0}) of the PUF P_{S0} and passes the authentication by S₀.

S₀ decrypts the cipher3 by using key_{S0-SN0} and gets plain3:

$$\begin{aligned} \text{plain3} &= D(\text{key}_{S0-SN0}, \text{cipher3}) \\ &= D(\text{key}_{S0-SN0}, E(\text{key}_{SN0-S0}, \text{nonce1})). \end{aligned} \quad (11)$$

S₀ constructs and sends a message authentication code (MAC) to the SN₀:

$$S_0 \longrightarrow SN_0: \text{MAC}(\text{plain3}). \quad (12)$$

- (4) SN₀ checks if the equation MAC(plain3) = MAC(nonce1) is correct.

If correct, SN₀ infers that the S₀ carried out a correct nonce1 by computing the correct pairwise key, key_{S0-SN0}, which is derived by the correct response r_{S0} of PUF P_{S0}. Thus, the sensor S₀ passes the authentication by SN₀.

If not, SN₀ deduces that the sensor is not a valid S₀ as it declares, since it cannot output a correct response of r_{S0} so as to compute a correct key_{S0-SN0}. S₀ fails the authentication and quits.

From now on, an intracluster pairwise key key_{S0-SN0} = (key_{SN0-S0}) is established and utilized to encrypt the communications between S₀ and SN₀.

The mutual authentication is implemented by PUF CRP and the intracluster communication security is assured. Besides, the process is safe from the replay attack because the temporary key is derived involving the timestamps.

3.4. Mobile Sensors Subscheme. The network is dynamic during the working time. As shown in Figure 5, the sensor S₆ moves from the cluster C₂, where it is thrown on, into the cluster region of C₁. Therefore, the Home-SN of S₆ is SN₂ and the Present-SN is SN₁. However, the SN₁ does not share the PUF CRP of S₆, and it should implement the authentication and key distribution via the help of SN₂. The subscheme is described as shown in Figure 7.

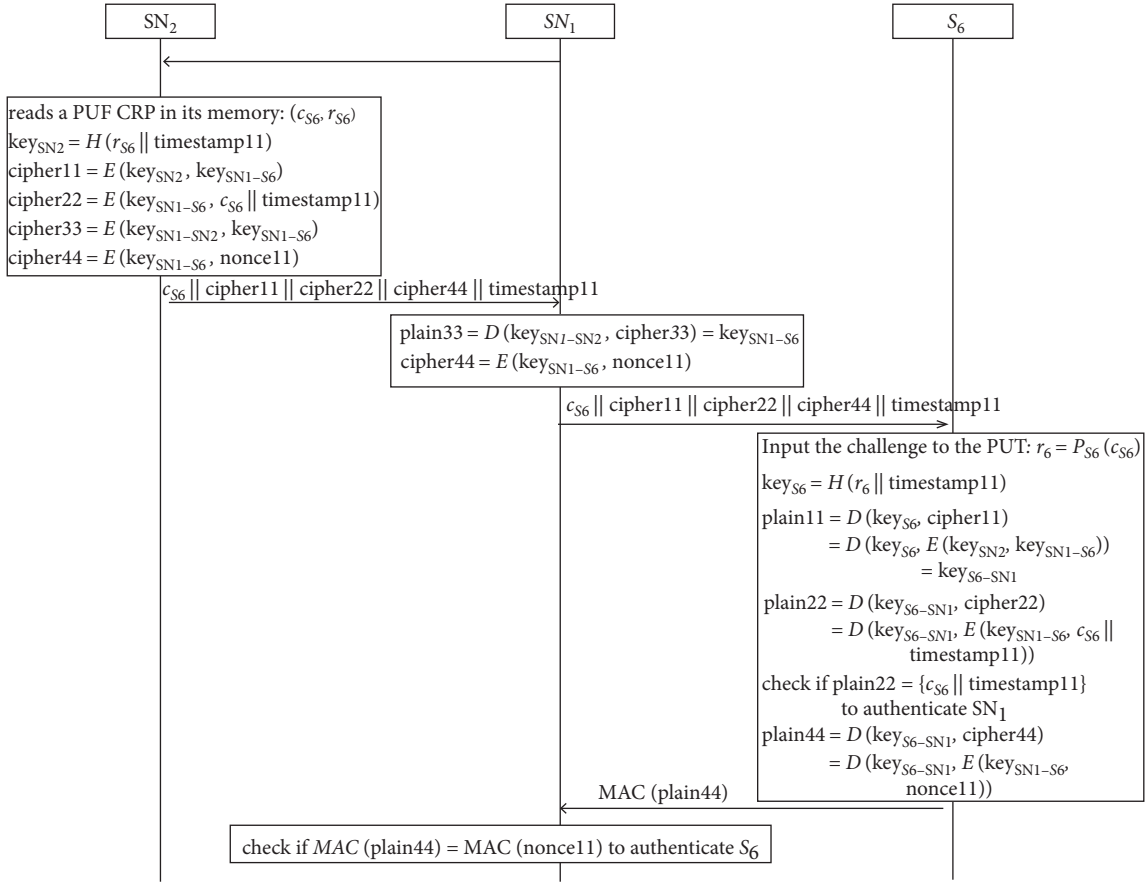
- (1) The sink node SN₁ broadcasts the id of sensor S₆ to request help. This is a round of intercluster communication.
- (2) The sink node SN₂ reads a PUF CRP in its memory: (c_{S6}, r_{S6}) by indexing of id_{S6}. SN₂ computes a temporary key, key_{SN2}:

$$\text{key}_{SN2} = H(\|r_{S6} \text{timestamp1}\|). \quad (13)$$

SN₂ generates a session key between SN₁ and S₆, key_{SN1-S6}, and encrypts it by key_{SN2}:

$$\text{cipher11} = E(\text{key}_{SN2}, \text{key}_{SN1-S6}). \quad (14)$$

Then, SN₂ encrypts the c_{S6} by using key_{SN1-S6} to get cipher22:

FIGURE 7: Authentication and key distribution between SN₁ and mobile sensor S₆.

$$\text{cipher22} = E(\text{key}_{SN_1-S_6}, c_{S_6} \parallel \text{timestamp11}). \quad (15)$$

SN₂ encrypts the key $\text{key}_{SN_1-S_6}$ by an intercluster key, $\text{key}_{SN_1-SN_2}$, shared between SN₁ and SN₂:

$$\text{cipher33} = E(\text{key}_{SN_1-SN_2}, \text{key}_{SN_1-S_6}). \quad (16)$$

SN₂ sends the challenge c_{S_6} , cipher11, cipher22, and cipher33 to SN₁:

$$SN_2 \longrightarrow SN_1: \|c_{S_6} \parallel \text{cipher11} \parallel \text{cipher22} \parallel \text{cipher33} \parallel \text{timestamp11}. \quad (17)$$

(3) SN₁ decrypts the cipher33 to get the session key, $\text{key}_{SN_1-S_6}$:

$$\text{plain33} = D(\text{key}_{SN_1-SN_2}, \text{cipher33}) = \text{key}_{SN_1-S_6}. \quad (18)$$

Then, SN₁ generates a secret random number nonce11 and encrypts it by using $\text{key}_{SN_1-S_6}$:

$$\text{cipher44} = E(\text{key}_{SN_1-S_6}, \text{nonce11}). \quad (19)$$

SN₁ sends the challenge c_{S_6} , cipher11, cipher22, and cipher44 to the sensor S₆:

$$SN_1 \longrightarrow S_6: \|c_{S_6} \parallel \text{cipher11} \parallel \text{cipher22} \parallel \text{cipher44} \parallel \text{timestamp11}. \quad (20)$$

(4) After receiving the message, the sensor S₆ firstly inputs c_{S_6} into the PUF structure P_{S_6} , which is embedded during the initialization phase, and gets the output response r_6 :

$$r_6 = P_{S_6}(c_{S_6}). \quad (21)$$

S₆ computes a temporary key, key_{S_6} :

$$\text{key}_{S_6} = H(\|r_6 \parallel \text{timestamp11}). \quad (22)$$

Then, S₆ decrypts cipher11 to get the pairwise key, $\text{key}_{S_6-SN_1}$:

$$\begin{aligned} \text{plain11} &= D(\text{key}_{S_6}, \text{cipher11}) \\ &= D(\text{key}_{S_6}, E(\text{key}_{SN_2}, \text{key}_{SN_1-S_6})) = \text{key}_{S_6-SN_1}. \end{aligned} \quad (23)$$

S₆ decrypts cipher22 by using $\text{key}_{S_6-SN_1}$ and gets plain22:

$$\begin{aligned} \text{plain22} &= D(\text{key}_{S_6-SN_1}, \text{cipher22}) \\ &= D(\text{key}_{S_6-SN_1}, E(\text{key}_{SN_1-S_6}, c_{S_6} \parallel \text{timestamp11})). \end{aligned} \quad (24)$$

The sensor S_6 checks if the equation $\text{plain22} = \{c_{S_6} \parallel \text{timestamp11}\}$ is correct.

If not, S_6 deduces that the cipher11 and cipher22 are not generated from its valid Home-SN or not forwarded from a trusted Present-SN. The SN_1 fails the authentication by S_6 and quits.

If correct, S_6 infers that the Present-SN SN_1 is trusted by SN_2 and passed the authentication.

S_6 decrypts cipher44 by using $\text{key}_{S_6-SN_1}$ and gets plain44:

$$\begin{aligned} \text{plain44} &= D(\text{key}_{S_6-SN_1}, \text{cipher44}) \\ &= D(\text{key}_{S_6-SN_1}, E(\text{key}_{SN_1-S_6}, \text{nonce11})). \end{aligned} \quad (25)$$

S_0 constructs and sends a message authentication code (MAC) to the SN_1 :

$$S_6 \longrightarrow SN_1: \text{MAC}(\text{plain44}). \quad (26)$$

- (5) SN_1 checks if the equation $\text{MAC}(\text{plain44}) = \text{MAC}(\text{nonce11})$ is correct:

If correct, SN_1 infers that the S_6 carried out a correct nonce11 by computing the correct pairwise key, $\text{key}_{S_6-SN_1}$, which is derived by the correct response r_{S_6} of PUF P_{S_6} . Thus, the sensor S_6 passes the authentication by SN_1 .

If not, SN_1 deduces that the sensor is not a valid S_6 as it declares, since it cannot output a correct response of r_{S_6} to compute a correct $\text{key}_{S_6-SN_1}$. S_6 fails the authentication and quits.

4. Simulation, Analysis and Comparisons

We present the security and performance evaluation of the proposed scheme through simulation experiments and analysis. We provide extensive simulations to verify the performance metrics such as secure connectivity, resilience against node capture, memory consumption, and communication overhead. We compare the proposed approach with other key management schemes. In the simulation, we assume 10000 sensor nodes, and 100 sink nodes are randomly distributed in a 1000×1000 m field. Each sensor node has a fixed speed ranging from 1 to 10 m/s. The radio range of each sensor node is considered as 50 m.

4.1. Mutual Authentication. The basic idea of the authentication of our approach is the challenge-response mechanism based on the PUF CRP. In both subschemes, mutual authentication between the sink node and the (static or mobile) sensor is assured. Furthermore, the scheme quits before key distribution process if the authentication failed,

that is, an unauthenticated sensor cannot participate the whole communication network. Compared with the PKI method, the PUF-based authentication speeds up and reduces the storage requirement.

In some proposed PUF authentication schemes [21, 22], the challenge and response are always sent in plaintext. If attackers catch an entire PUF CRP, they are able to launch the replay attack and man-in-the-middle attack. In order to resist the replay attack, a strong PUF is usually employed to provide a plenty of CRPs and each of them is only used once. Then, different CRPs of a PUF are openly exposed in a dynamic network where a mobile node needs frequent authentication with new neighbors. This PUF structure is vulnerable to the modelling attack that tries to guess and predict the response value related to a certain challenge.

In our scheme, the PUF response is not transmitted in plain but converted into an encryption key by hashing with a timestamp. A node succeeds the authentication if it decrypts and carries out a correct plaintext. This is a kind of symmetric authentication [34] combined with the PUF challenge-response mechanism. In order to prevent the replay attack, a timestamp has been used. The fact that the PUF response is not transmitted in plain effectively resists the modelling attack on PUF.

4.2. Overheads. We mainly consider the energy consumption in terms of storage, communication, and computation overheads. We mainly consider the following assumptions: MAC size is considered as 4 bytes, 4 bytes for time stamp, random nonce as 16 bytes, 32 bytes for key size, and 32 bytes for challenge/response of a PUF. We also consider 2 bytes for the node ids. The ciphertext has the same length with the key.

4.2.1. Key Storage. In our approach, during the initialization phase, each sensor is not predistributed with any key in its memory, while each sink node is predistributed with n/m PUF CRPs. A PUF structure is embedded in a sensor (as a hardware) during the initialization phase (therefore, the storage overhead is not discussed in this paper). After the key distribution, the sensor stores 1 intracluster session key established with the sink node, while the sink node stores one intracluster session key for each cluster-member sensor. All the intermediate data generated in the key distribution process is deleted to release the storage space. Therefore, the storage overhead of a sensor is 32 bytes and that of a sink node is $(32 + 32 \times 2 + 2)n/m = 98n/m$ bytes.

Du et al. proposed an AP scheme [10], which is a pure random key predistribution scheme. The main idea is to preload only a small number of keys (denoted as l) in low-ended sensors, while preloading a relatively large number of keys (denoted as $M \gg l$) in each high-ended sink nodes. Any two nodes cannot establish a secure link if they do not share a common pairwise keys. Therefore, nodes need to store more keys to increase the probability of sharing common keys, which is defined as the secure connectivity. As analyzed in Erfani's scheme [16], the sensor memory is partitioned into two parts: store α predistributed keys in the first part and β postdeployment keys in the second part. Each pair of

neighboring nodes establish a common predistributed or postdeployment key to secure the communication. Erfani's scheme claimed that each sink node stores only 1 key; BS stores a key table, which contained some information about sensor nodes' keys. In addition, BS is aware of sink nodes' keys.

Table 1 compares the amount of memory required for storing keys in the proposed scheme and other two solutions. The key storage in sink node of our scheme is higher than Erfani's scheme, but the storage of sensor is much lower than both Erfani's and AP schemes. Therefore, our scheme is efficient for resource limited sensor nodes, and this performance also brings an advantage of better resilience against node capture attack.

4.2.2. Communication Overhead. In this paper, the communication overhead is measured by the message size and transmission rounds but does not consider the message overhead consisting of a protocol ID, a message ID, a checksum, and the headers and footers of the low-level network layers.

We analyze the communication overhead for static and mobile subschemes, respectively.

In the static subscheme, to establish an intracluster pairwise key, the sensor sends only 1 MAC packet with 4 bytes, while the sink node sends 1 packet with 132 bytes.

In the mobile subscheme, to establish an intracluster pairwise key, the sensor sends only 1 MAC packet with 4 bytes, while the Home-SN sends 1 packet with 132 bytes and the Present-SN sends 2 packets with 2 bytes and 132 bytes.

Compared with the random key predistribution schemes like the AP, nodes do not need key construction or authentication but try to find a common key by sending the key indexes or encrypted challenges. The transmitted message size is linearly related to the size of the keyring. However, if two neighboring nodes do not share a common key, they must send further messages to ≥ 2 hops intermediate nodes.

4.2.3. Computation Overhead. The most computation overhead is related to cryptography and authentication operations, and the PUF computation especially for sensors. As shown in Table 2, to establish an intracluster pairwise key, the number of encryption or decryption operations in each sensor is 3 and 3 or 5 in a sink node. All these schemes use light weight cryptography methods. The computation overhead is higher than the random key predistribution scheme AP but still acceptable for both sensors and sink nodes.

4.3. Secure Connectivity. The security connectivity of a network is defined as the probability that two entities can establish a session key to secure the communications. Since this paper mainly proposes an approach for intracluster authentication and key distribution, we define the conception of "intracluster secure connectivity" as the probability that a sink node can establish a pairwise key with a cluster-member (static or mobile) sensor.

TABLE 1: Comparison of storage overhead in different schemes (bytes).

	Our scheme	Erfani's	AP
Sensor	32	$32(\alpha + \beta)$	$32l$
Sink	$98n/m$	32	$32M$
BS	—	$32[n(\alpha + \beta) + m]$	—

TABLE 2: Comparison of computation overhead.

	Our scheme	AP
Cryptography in sensor	3	NA
Cryptography in sink node	Static: 3 Mobile: 5	NA
PUF	1	NA

This scheme is a kind of deterministic key distribution model, in which any sensor node can successfully establish a session key with no matter the Home-SN or the Present-SN. Therefore, the intracluster security connectivity is 100% in this scheme, which is a remarkable improvement compared with the probabilistic key distribution schemes [9, 10, 12].

The random schemes, like AP scheme, must increase the amount of key storage to achieve high security connectivity. Figure 8 shows the secure connectivity versus the key pool size P in the AP. There are four solid curves in Figure 8, from bottom to top, corresponding parameters $[l, M]$ of [5, 125], [10, 250], [15, 375], and [20, 500], respectively. It is observed that the probability of sharing key increases when the number of preloaded keys increases. For the same parameters $[l, M]$, the probability of sharing key decreases as the key pool size becomes large. In Figure 9, we also plot the secure connectivity for different numbers of preloaded keys in the AP and our scheme. As analyzed in the above section, the storage overhead of the sink node in our scheme is $98n/m \approx 10000$ bytes, almost 300 32bytes-keys. It is worth emphasizing that the key storage of sensor nodes in our proposal is 0, which is significantly lower than that of AP scheme, but the connectivity is significantly higher than that of AP scheme. The Erfani's scheme is also claimed of providing full secure connectivity in [16], however there is a trade-off between α and β in balancing the storage, connectivity, and resilience.

4.4. Resilience Against Node Capture. Sensor networks are usually deployed in an unattended environment, and attackers illegally obtain the secret information of nodes by capturing nodes and other physical attacks. Resilience against node capture is defined as the probability $F(x)$ that the attacker can obtain the key in the uncaptured node directly or indirectly according to a certain number of captured nodes x :

$$F(x) = \frac{\text{number of compromised links between uncaptured nodes}}{\text{number of uncompromised links}}. \quad (27)$$

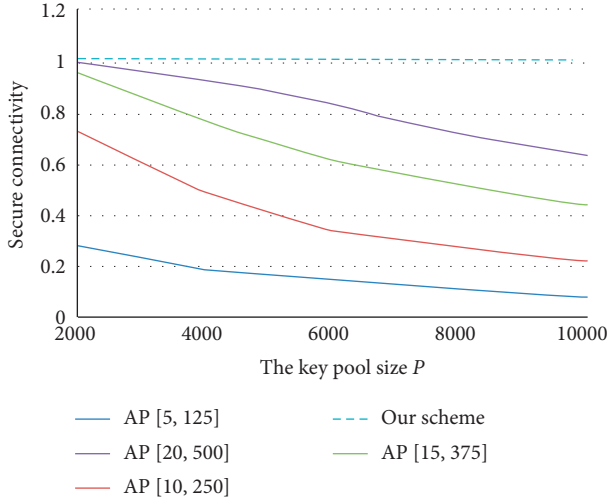


FIGURE 8: Secure connectivity versus the key pool size P .

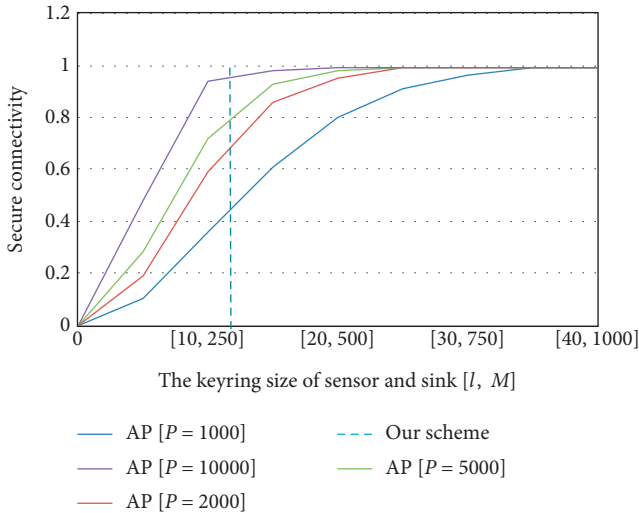


FIGURE 9: Secure connectivity versus keyring size.

4.4.1. Resilience against the Sensor Capture. Different from the traditional random key predistribution schemes [4, 5, 7], in this scheme, the sensor node does not prestore any keys or other key materials, which not only reduces the storage cost of the sensor but also improves the resilience against the sensor capture because the attacker cannot obtain any key that belong to a safe node despite capturing a sensor physically. Therefore, our proposal has perfect resilience against the sensor capture; that is,

$$F(x_S) = 0, \quad (28)$$

where x_S represents the number of captured sensor nodes.

4.4.2. Resilience against the Sink Node Capture. The sink node acts as the cluster head, which maintains the intra-cluster secure communication with the cluster members and also the intercluster secure communication with other cluster heads externally. Each sink node is prestore with a

number of CRPs in the initialization phase and uses the CRPs to authenticate and distribute pairwise keys with its cluster-member sensors.

The physical capturing of a sink node breaks up both the internal and external cluster communication of it. The dismissed cluster members (sensors) become isolated nodes and may join other clusters. By repeating the authentication and key distribution process, the dismissed sensor obtains a new session key with its new cluster head. There is not any key that belong to a safe node that will be exposed by a physical captured sink node. Therefore, our proposal has perfect resilience against the sink node capture; that is,

$$F(x_{SN}) = 0, \quad (29)$$

where x_{SN} represents the number of captured sink nodes.

4.4.3. Resilience against Selective Node Capture. Huang et al. [35] pointed out that, in many key management schemes, the selective node capture causes more damage to the network. In the selective node capture attacks, attackers attempt to capture nodes that may reveal more valid and fresh information about uncaptured nodes. In our proposed scheme, an adversary cannot figure out which sink node owns the CRP of a certain sensor, because all CRPs are randomly and safely selected from the CRP pool. Therefore, unless the adversary compromises all the sink nodes, it cannot choose a certain sink node to capture to maximize the uncompromised keys.

4.4.4. Simulation Results. The AP scheme [10] proposed by Du et al. is a pure random key predistribution scheme in cluster sensor networks, with the advantage in saving nodes' communication and computation overheads. But it is hard to balance the tradeoff between the security connectivity and security. Boujelben et al. [12] improved the AP by combining the Blom matrix in terms of the resilience against node capture but require quantity of storage overhead for matrix parameters. Erfani's scheme [16] is a combination of the key pre-distribution and post-deployment key management scheme. When a sensor is captured, all pre-distributed and postdeployment keys of the node are compromised. But since the postdeployment key is not selected from the key pool, the compromise of such key does not affect the security of other communications, whereas compromising the pre-distributed keys of a sensor node will make other communication links insecure, because such keys are selected from the key pool and might be common with some sensors. Erfani's scheme provides better resilience against node capture attack than the AP, and the resilience of sensor network depends on the number of pre-distributed keys α and key pool size P .

We will compare our scheme with these schemes by simulation experiments. The size of key pool in AP, Boujelben's, and Erfani's schemes is $P = 10000$. Similar to the experiments environment in [16], the keyring size is 100 in Erfani's scheme.

As shown in Figures 10 and 11, the experimental results prove that, in the random key predistribution schemes, the resilience against node capture gets worse and worse with the number of captured nodes increasing, because the nodes store a large number of keys. In Boujelben et al. scheme, the nodes store matrixes instead of keys, so the resilience against node capture is better than that in the AP scheme, but the storage cost is λ times that of AP (λ is the matrix parameter).

In our scheme, the sensor node does not store any key, and the sink node stores the CRPs rather than the key as well, so perfect resilience against node capture is provided.

4.5. PUF Security. In this paper, PUF is the core of the authentication and key distribution. The security of the PUF is crucially important. The main threats to some PUF-based schemes [36] include man-in-the-middle attack, replay attack, and the modelling attack to the PUF, because the PUF CRPs are transmitted in plain form. A PUF is considered failed when the adversaries can guess more than 75% bits of the response to a challenge after obtaining enough amount of CRPs of a given PUF. In our proposal, the response, generated by a PUF on a sensor on-the-fly, is not sent to the sink node directly but is utilized as an encryption key to encrypt the challenge. Such design can successfully protect the PUF from cloning attack, modelling attack, and side channel attacks, including electromagnetic analysis attack and differential fault attack. The eavesdropping is invalid, since all the transmitted messages are encrypted with symmetric algorithm (e.g., AES), the attackers cannot get any plain information about responses or keys. The scheme can withstand the man-in-the-middle attack and tamper attack, since the encrypted response protects its integrity in the wireless communications.

In the replay attack, an attacker resends an old message, which has been sent for key generation request. In the proposed approach, timestamp has been used in generating the temporary key to prevent the replay attack. Besides, the session key is randomly generated between the sink node and sensor and will not be the same as the a priori key. An attacker can continuously resend an old message to consume the energy of sensor nodes; however, these messages will be discarded.

Table 3 shows the comprehensive comparison results among different authentication and key distribution schemes for sensor networks proposed in recent years. Unlike the key predistribution schemes, for example, AP [10], our scheme is perfectly resilient against node capture attacks, because a sensor does not prestore any keys that might secure other sensors' communications. PUF CRPs provide a type of authentication by a challenge-response mechanism, but Chatterjee's scheme [23] does not guarantee mutual authentication between two parties. In addition, PUFs provide another type of security guarantee implied by

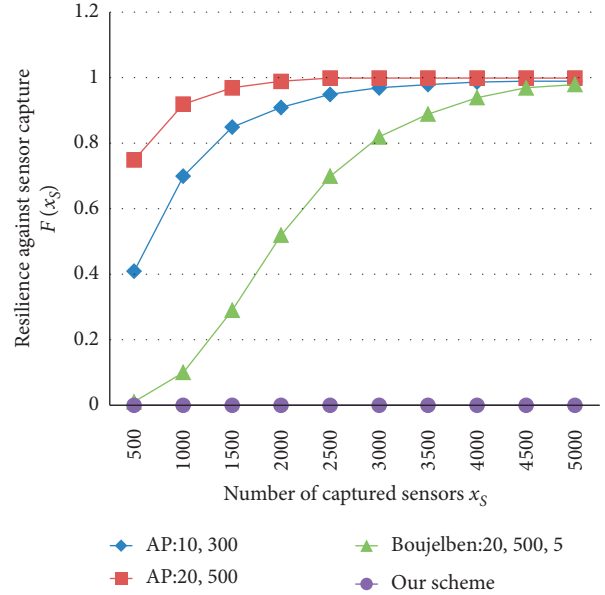


FIGURE 10: Resilience against the sensor capture.

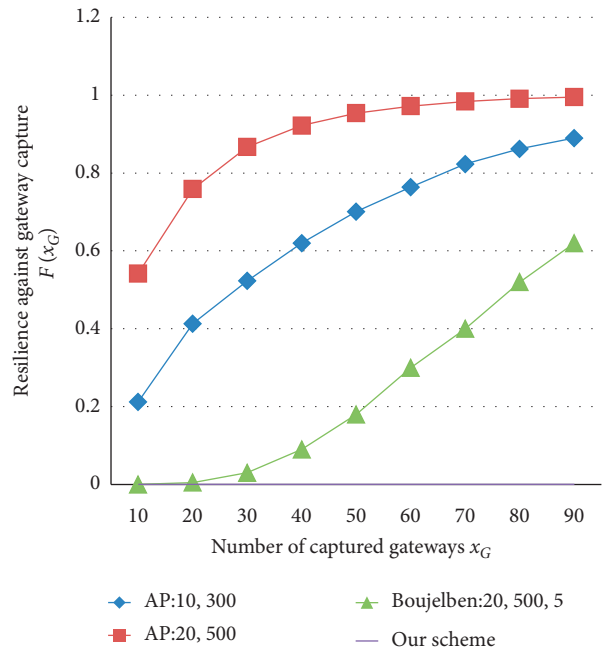


FIGURE 11: Resilience against the sink node capture.

their unclonability and tamper evidence. Such property is only available to PUF-based solutions. However, PUF CRPs are sent as plaintext in [23, 25], which make them vulnerable to impersonation attack, but we avoid this in our scheme by encrypting the response of the CRPs. Also, in [14, 23, 25, 27], they used public key algorithm that consumed more computation overhead than the AP [10] and our proposal.

TABLE 3: Comparisons of different key distribution schemes.

Property	Ours	AP [10]	Lee and Kim [14]	Erfani et al. [16]	Chatterjee et al. [25]	Li et al. [27]
Public key encryption	No	No	Yes	No	Yes	Yes
Key redistribution	No	Yes	No	Yes	No	No
Perfect resilience against node capture	Yes	No	—	No	—	—
PUF-based	Yes	No	No	No	Yes	Yes
Mutual authentication	Yes	No	Yes	Yes	No	Yes
Resistant to modelling attacks	Yes	—	—	—	No	No
Resistant to eavesdropping attacks	Yes	—	—	—	Yes	Yes
Resistant to collusion attacks	Yes	—	—	Yes	—	—

—: not applicable.

5. Conclusions

In a dynamic sensor network, how to ensure two communicating (static or mobile) nodes authenticate and share a pairwise key is difficult because the sensors' mobility pattern or track is unknown. In this paper, we propose a mutual-authenticated key distribution scheme for the intracluster communication. In order to reduce the storage overhead and the key exposure risk of low-end sensors, we employ a CRO Physical Unclonable Function (PUF) in the mutual-authentication process, which has the lightweight, unclonability, and unpredictability advantages. Compared with the classical PUF challenge-response authentication mechanism in some literatures, the PUF response is not transmitted in plain forms so as to resist the modelling attacks on PUFs. We also demonstrate that the proposed scheme improves the secure connectivity and other performances by analysis and experiments.

Data Availability

The data are available at <https://www.zhangqiaokeyan.com/patent-detail/06120103885959.html>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (under Grant 61902163), the Research Startup Foundation of Jinling Institute of Technology (under Grant JIT-B-201639), and the Key Program of National Key Research and Development Project "Cybersecurity" (under Grant 2017YFB0802800).

References

- [1] D. Carman, P. Kruus, and B. Matt, *Constraints and approaches for distributed sensor network security (final)*, pp. 1–139, NAI Labs Technical Report, NAI Labs, MD, USA, 2000.
- [2] Y. Ren, Y. Leng, J. Qi et al., "Multiple cloud storage mechanism based on blockchain in smart homes," *Future Generation Computer Systems*, vol. 115, pp. 304–313, 2021.
- [3] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.
- [4] G. Liu, Q. Yang, and H. Wang, "Trust assessment in online social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, pp. 994–1007, 2018.
- [5] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 3, p. 1, 2021.
- [6] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and F. Liming, "Secure keyword search and data sharing mechanism for cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 4, p. 1, 2020.
- [7] C. Y. Chen and H. C. Chao, "A survey of key distribution in wireless sensor networks," *Security and Communication Networks*, vol. 7, 2014.
- [8] D. Farooq and M. Gull, "A survey about applications, issues and challenges of sensor network," *International Journal of Computer Applications*, vol. 180, no. 19, pp. 41–46, 2018.
- [9] L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks," in *Proceedings of the ACM Conference on Computer and Communication Security*, pp. 41–47, Washington, DC, USA, November 2002.
- [10] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 24–34, 2007.
- [11] R. Blom, "An optimal class of symmetric key generation systems," *Advances in Cryptology-Eurocrypt*, vol. 84, pp. 335–338, 1984.
- [12] M. Boujelben, O. Cheikhrouhou, M. Abid, and H. Youssef, "Establishing pairwise keys in heterogeneous two-tiered wireless sensor networks," in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications Athens*, pp. 18–23, Athens, Greece, 2009.
- [13] K. Benamar, F. Mohammed, and M. Abdellah, "Architecture aware key management scheme for wireless sensor networks," *International Journal of Information Technology & Computer Science*, vol. 4, no. 12, pp. 50–59, 2012.
- [14] S. Lee and K. Kim, "Key renewal scheme with sensor authentication under clustered wireless sensor networks," *Electronics Letters*, vol. 51, no. 4, pp. 368–369, 2015.
- [15] K. Han, K. Kim, and T. Shon, "Untraceable mobile node authentication in WSN," *Sensors*, vol. 10, no. 5, pp. 4410–4429, 2010.
- [16] S. H. Erfani, H. H. S. Javadi, and A. M. Rahmani, "A dynamic key management scheme for dynamic wireless sensor networks," *Security and Communication Networks*, vol. 8, no. 6, pp. 1040–1049, 2015.
- [17] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in

- DWSNs,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
- [18] J. Delvaux, R. Peeters, D. Gu, and I. Verbauwhede, “A survey on lightweight entity authentication with strong PUFs,” *ACM Computing Surveys*, vol. 48, no. 2, pp. 1–42, 2015.
- [19] A. R. Sadeghi and D. Naccache, “Towards hardware-intrinsic security,” *Information Security & Cryptography*, vol. 364, no. 1849, pp. 3215–3230, 2010.
- [20] B. Gassend, D. E. Clarke, and M. V. Dijk, “Silicon physical random Functions,” in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 18–22, Washington, DC, USA, November 2002.
- [21] P. Tuyls, G. J. Schrijen, and B. Skoric, “Read-proof hardware from protective coatings,” in *Proceedings of the 8th International Workshop of Cryptographic Hardware and Embedded Systems—CHES 2006*, Yokohama, Japan, October 2006.
- [22] C. Brzuska, M. Fischlin, H. Schröder, and S. Katzenbeisser, “Physically uncloneable functions in the universal composition framework,” *Advances in Cryptology—CRYPTO 2011*, vol. 6841, pp. 51–70, 2011.
- [23] A. Allam, “FPGA-based authenticated key exchange scheme utilizing PUF and CSI for wireless networks,” in *Proceedings of the IEEE International Conference on System of Systems Engineering (SoSE 2015)*, pp. 170–175, Monterey, CA, USA, 2015.
- [24] R. Bahrapour and R. E. Atani, “A novel key management protocol for wireless sensor networks based on PUFs,” *International Journal of Future Generation Communication & Networking*, vol. 6, no. 2, pp. 93–106, 2013.
- [25] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, “A PUF-based secure communication protocol for IoT,” *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 3, pp. 1–25, 2017.
- [26] A. Braeken, “PUF based authentication protocol for IoT,” *Symmetry*, vol. 10, no. 8, pp. 1–15, 2018.
- [27] S. S. Li, Y. C. Huang, and B. Yu, “A PUF-based low cost secure communication scheme for IoT,” *Acta Electronica Sinica*, vol. 47, no. 04, pp. 46–51, 2019.
- [28] Z. Zhang, Y. Liu, Q. Zuo, L. Harn, S. Qiu, and Y. Cheng, “PUF-based key distribution in wireless sensor networks,” *Computers, Materials & Continua*, vol. 64, no. 2, pp. 1261–1280, 2020.
- [29] Y. Cui, C. Wang, and W. Liu, “Low-cost configurable ring oscillator PUF with improved uniqueness,” in *Proceedings of the IEEE International Symposium on Circuits & Systems*, pp. 558–561, IEEE, Baltimore, MD, USA, 2016.
- [30] L. Zhang, C. H. Wang, W. Q. Liu, M. O’Neill, and F. Lombardi, “XOR Gate Based Low-Cost Configurable RO PUF,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, Baltimore, MD, USA, 2017.
- [31] Y. Cui, C. Gu, C. Wang, M. O’Neill, and W. Liu, “Ultra-lightweight and reconfigurable tristate inverter based physical uncloneable function design,” *IEEE Access*, vol. 6, pp. 28478–28487, 2018.
- [32] A. Maiti and P. Schaumont, “Improved ring oscillator PUF: an FPGA-friendly secure primitive,” *Journal of Cryptology*, vol. 24, no. 2, pp. 375–397, 2011.
- [33] Y. Mohamed, Y. Moustafa, and A. Khaled, “Energy-aware management for cluster-based sensor networks,” *Computer Networks*, vol. 43, no. 5, pp. 649–668, 2003.
- [34] S. Malhotra and M. C. Trivedi, “Symmetric key based authentication mechanism for secure communication in MANETs,” in *Intelligent Communication and Computational Technologies* Springer, Singapore, 2018.
- [35] D. Huang, M. Mehta, D. Medhi, and L. Harn, “Location-aware key management scheme for wireless sensor networks categories and subject descriptors,” in *Proceedings of the Second ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 29–42, Washington, DC, USA, October 2004.
- [36] M. N. Aman, K. C. Chua, and B. Sikdar, “Position paper: physical uncloneable functions for IoT security,” in *Proceedings of the ACM international workshop*, pp. 10–13, ACM, Seoul, Republic of Korea, 2016.

Research Article

A Black-Box Attack Method against Machine-Learning-Based Anomaly Network Flow Detection Models

Sensen Guo , Jinxiong Zhao , Xiaoyu Li, Junhong Duan, Dejun Mu, and Xiao Jing

School of Cybersecurity, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

Correspondence should be addressed to Sensen Guo; guosensen@mail.nwpu.edu.cn

Received 15 January 2021; Revised 10 February 2021; Accepted 5 March 2021; Published 24 April 2021

Academic Editor: Qing Yang

Copyright © 2021 Sensen Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, machine learning has made tremendous progress in the fields of computer vision, natural language processing, and cybersecurity; however, we cannot ignore that machine learning models are vulnerable to adversarial examples, with some minor malicious input modifications, while appearing unmodified to human observers, the outputs of machine learning-based model can be misled easily. Likewise, attackers can bypass machine-learning-based security defenses model to attack systems in real time by generating adversarial examples. In this paper, we propose a black-box attack method against machine-learning-based anomaly network flow detection algorithms. Our attack strategy consists in training another model to substitute for the target machine learning model. Based on the overall understanding of the substitute model and the migration of the adversarial examples, we use the substitute model to craft adversarial examples. The experiment has shown that our method can attack the target model effectively. We attack several kinds of network flow detection models, which are based on different kinds of machine learning methods, and we find that the adversarial examples crafted by our method can bypass the detection of the target model with high probability.

1. Introduction

Along with the rapid development of computer technology and communication technology, the computer network is acting a more and more important role in information society nowadays, and it has already become an essential part of people's lives. Meanwhile, the rapid development of the Internet also brings about people many security problems, and how to protect the transmission of secret information on the network effectively has become a concern.

With the development of computer technology, especially the improvement of calculating speed, transferring speed, and memory capacity, machine learning (ML), especially deep learning (DL), has developed very fast and has been widely used in many fields, such as natural language processing (NLP) [1], the Internet of things (IoT) [2, 3], computer vision (CV) [4], and time series prediction [5, 6]. In recent years, many scholars have also tried to use machine learning algorithm to solve network security detection problems.

Pervez et al. [7] proposed a filtering algorithm, which is based on a Support Vector Machine (SVM) classifier to

identify malicious network intrusion on the NSL-KDD intrusion detection database; their method achieves very high classification accuracy in the training set, but the performance in the test set is not ideal.

Experimented with a wide variety of attacks and different k values, Rao et al. [8] used Indexed Partial Distance Search k -Nearest Neighbor (IKPDS) to recognize attacks. They tested their method with 12,597 samples that were randomly selected from the NSL-KDD dataset, resulting in 99.6% accuracy in their experiment.

Azad et al. [9] proposed an intrusion detection method based on the genetic algorithm and a C4.5 decision tree; they trained their model on the KDD Cup 99 dataset and got 99.89% accuracy rate and a 0.11% FAR.

Deep Belief Network (DBN) is also used by many scholars in intrusion detection, by training on 40% NSL-KDD database. Alom et al. [10] proposed a Deep Belief Network (DBN)-based intrusion detection model through a series of experiments. In their experiment, their DBN intrusion detection model achieved 97.5% accuracy after 50 iterations, and it can identify unknown attacks effectively.

Yin et al. [11] proposed the intrusion detection (RNN-IDS) model based on a cyclic neural network. They used the NSL-KDD database to evaluate the performance of their model in multi-classification and binary classification; they also tested the influence of different learning rates and the number of neurons on the performance of their model. In the binary classification experiment, the training and test accuracy of their model achieved 99.81% and 83.28%, respectively, and in the multi-classification experiment, the training and test accuracy achieved 99.53% and 81.29%, respectively.

By taking network flow data as images, Wang et al. [12] proposed an abnormal traffic classification method based on convolutional neural network (CNN); in their study, they conduct experiments in two scenarios with three types of classifiers, and their final average accuracy achieves 99.41%. Besides, many other machine-learning-based applications in cybersecurity are also introduced in [13].

Although the abovementioned developments represent great strides in many fields, machine learning has its inner shortages. Szegedy et al. [14] found that machine learning, especially deep learning, is vulnerable to adversarial examples. A machine learning (ML) or deep learning (DL) model can easily be fooled by adding some well-designed noise to the inputs. Since Szegedy et al. first discovered adversarial examples for deep learning in 2013, the academic and security communities have also realized that even the most advanced machine learning algorithms can easily be fooled by the adversarial examples, which are carefully crafted by the attackers. This will make it difficult for the machine-learning-based model to play its due role in practical applications.

The main contribution of this paper includes the following:

- (i) An untargeted black-box adversarial example generation method for the machine-learning-based abnormal network flow detector is proposed in this paper.
- (ii) The differences in the method of generating adversarial example between the field of computer vision and intrusion detection system are discussed in this paper.
- (iii) The key points about the generate adversarial example against anomaly network flow detection are discussed in this paper.

The main notations and symbols used in this paper are listed in Table 1.

The rest of this paper is organized as follows. In Section 2, the work related to adversarial examples generate method is reviewed. Section 3 explains the key point of adversarial example generate method in the field of IDS. Section 4 details our black-box attack method toward the machine-learning-based network traffic detector. Section 5 introduces methods and the specific steps of our black-box attack method. Section 6 is the experimental results and analysis. Section 7 concludes this paper.

2. Related Work

The current adversarial example generation algorithms for machine learning are mainly concentrated in the field of computer vision. Szegedy et al. [15] first introduced the concept of adversarial examples for deep neural networks in 2014. They introduced a method named L-BFGS to generate adversarial examples, and it can be expressed as

$$\begin{aligned} \min_{x^{\text{adv}}} c\|r\| + J_{\theta}(x^{\text{adv}}, I^{\text{adv}}) \\ \text{s.t. } x^{\text{adv}} \in [0, 1], \end{aligned} \quad (1)$$

where c is a constant, calculated the by the line-searching method; $J(\cdot)$ is the lost function; and r is the perturbation added to the original picture. The author opined that the perturbation added to the input layer will accumulate in the process of forwarding the propagation of the neural network until it becomes large enough to cross the classification boundary.

While L-BFGS Attack uses the method of linear search to find the optimal value, it is impractical and time-consuming. Goodfellow et al. [16] proposed a fast method named the Fast Gradient Sign Method (FGSM) to generate adversarial examples in 2014; they performed only one step gradient update along with the sign of gradient at each pixel, and their method can be expressed as

$$x^{\text{adv}} = x + \varepsilon \text{sign}(\nabla_x J(\theta, x, y)), \quad (2)$$

where x^{adv} is the adversarial example, x is the original data, and ε is the magnitude of the perturbation.

Kurakin et al. [17] proposed their method called Basic Iterative Method (BIM), which is the straightforward extension of FGSM by applying it multiple times with a small step size:

$$\begin{aligned} x_0 &= x, \\ x_{n+1} &= \text{Clip}_{x,\varepsilon}\{x_n + \alpha \text{sign}(\nabla_x J(x_n, y_{\text{true}}))\}, \end{aligned} \quad (3)$$

where $\text{Clip}_{x,\varepsilon}(A)$ denotes element-wise clipping A , with $A_{i,j}$ clipped to the range $[x_{i,j} - \varepsilon, x_{i,j} + \varepsilon]$.

To further attack a specific class, they chose the least-likely class of the prediction and tried to maximize the cross-entropy loss. This method is referred to as the Iterative Least-Likely Class method [18]:

$$\begin{aligned} x_0 &= x, \\ x_{n+1} &= \text{Clip}_{x,\varepsilon}\{x_n - \alpha \text{sign}(\nabla_x J(x_n, y_{LL}))\}. \end{aligned} \quad (4)$$

Using this method, they fooled the neural network with a crafted adversarial example image taken from a camera successfully.

The algorithms to generate adversarial examples introduced above are all based on white-box attacks. Among the black-box attack methods, Papernot N et al. proposed a method based on a substitute model; their strategy was to train a local substitute model, which shares the same decision boundary with the target model. The dataset used for training the substitute model is generated by the attacker and

TABLE 1: Notations and terminology used in this paper.

Notations and symbols	Description
\mathbf{x}	The original data
l	The class that is labeled by the machine learning model
\mathbf{x}^{adv}	The adversarial example
l^{adv}	The label of the adversarial examples
$J(\cdot)$	The loss function
η	The noise that is added to the original data
θ	The parameters of the machine learning model
$T(\cdot)$	The target machine learning model or deep learning model
$ST(\cdot)$	The substitute model
D	The constraint vector of the perturbation
ξ	The influence coefficients vector of the features
α	The step size of the perturbation in the single iteration

labeled by the target model. Adversarial examples are crafted using the substitute parameters, which are known to them. The adversarial examples generated by their method can not only fool the substitute model but also the target model, because both models have similar decision boundaries [19]. Beyond this, there are many other methods for generating adversarial examples, such as zeroth order optimization (ZOO) [20], one-pixel attack [21], natural GAN [22], natural evolution-strategy-based attack [23], boundary attack [24], and so on, and they have made great progress in the field of black-box adversarial example generate research. Besides, more research can be seen in [25, 26].

In the field of cybersecurity, Hu and Tan [27] performed a detailed analysis of the robustness of machine-learning-based malware algorithms. They proposed two pretense approaches under which malware can pretend to be benign and fool the detection algorithms. Grosse K et al. [28] also expanded the method that used in the field of computer version to attack Android malware detection models; on the DREBIN dataset, they achieved misclassification rates of up to 69%.

Anderson et al. [29] designed the DeepDGA, which is an extension of GAN. They tried to pseudo-randomly produce domain names that are difficult for modern DGA classifiers to detect. Their technique generates domains on a character-by-character basis and greatly exceeds the stealth of typical DGA techniques.

Using the NSL-KDD database, Yang K et al. [30] had tried to mimic the adversarial attacks against the deep neural network (DNN) model applied for NIDS in the real world, and they evaluate three different algorithms (attack based on substitute model, ZOO, and GAN) in launching adversarial attacks in the black-box model. In their work, the accuracy, precision, recall, and fscore of the target DNN model are significantly decreased under the black-box attack.

Training on the KDD Cup 99 dataset, Lin Z et al. [31] proposed IDSGAN, an improved framework of GAN against the intrusion detection system. In their study, the feasibility of the model is demonstrated to attack many detection systems with different attacks and excellent results are achieved; however, currently, the training of GAN is still unstable, and it has problems such as convergence failure and model collapse.

Although the main purpose of the adversarial attack by the adversarial example is to evade detection of the machine-learning-algorithm-based IDS system, the premise is that the adversarial examples crafted by the attacker should retain the attack function of the network behavior. Yang K et al. [30] retained the attack function by constraining the perturbations to the original attack traffic. Lin Z et al. [31] did it by keeping the functional features of each attack unchanged, but they did not further study how to limit the perturbations to make the adversarial examples conform to the physical characteristics of network traffic without distortion.

3. Adversarial Examples in the Field of IDS

Taking the classification problem as an example, generate adversarial example is usually to solve the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{x}'} \quad & J(f(\mathbf{x}^{\text{adv}}), l^{\text{adv}}) \\ \text{s.t.} \quad & \begin{cases} \|\eta\|_p \leq \varepsilon, \\ f(\mathbf{x}) = l, \\ l \neq l^{\text{adv}}, \end{cases} \end{aligned} \quad (5)$$

where $J(\cdot)$ is the loss function, $f(\cdot)$ is the target classification model, \mathbf{x} is the original data, \mathbf{x}^{adv} is the adversarial example, $l = f(\mathbf{x})$, $l^{\text{adv}} = f(\mathbf{x}^{\text{adv}})$, and η is the distance between the adversarial example \mathbf{x}^{adv} and the original data \mathbf{x} .

As shown in Figure 1, similar to the field of computer vision, in the field of IDS, the process of adversarial example generation is to add a subtle perturbation noise to the original malicious attack traffic data, so that the attacker can successfully bypass the detection of machine learning algorithm to carry out a malicious attack on the target model. De Lucas et al. [32] introduced many key points of adversarial example for traffic data; here, we focus on two key differences of adversarial example between the field of IDS and computer vision:

- (i) The direction of the noise η
- (ii) The static of the noise η

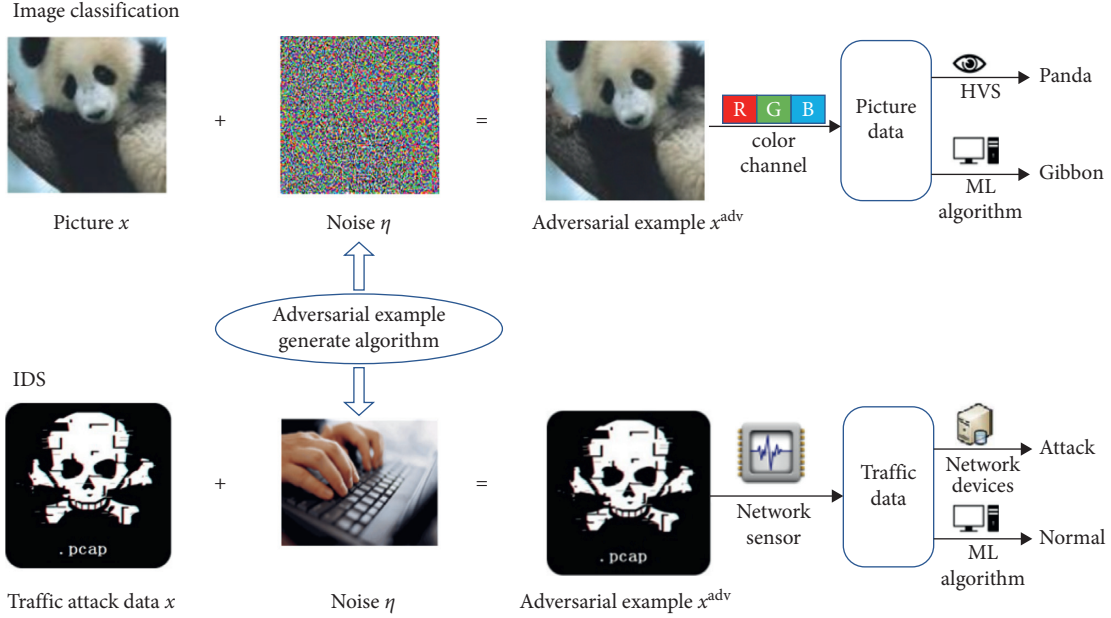


FIGURE 1: The differences of adversarial example generate process between IDS and computer vision.

3.1. The Direction of the Noise. In general, the process of generating adversarial example is to add appropriate amount of perturbation along with the direction of its gradient. The attacker can deceive the target model successfully by making the adversarial example cross the decision boundary of the target model; however, there is a key point that we must make sure the attack function is not lost while we add the noise to the original data.

In the field of computer vision, a picture file is composed of many pixels; each pixel is composed of three numbers, corresponding to three colors, namely, red, green, and blue, and each pixel shares the same attribute. However, in the field of intrusion detection, traffic connection data consist of an indefinite number of packets, and each packet comprises a lot of information, such as the five-tuple (the source IP, the source port, the destination IP, the destination port, and the protocol type), the packet header, and the payloads. Based on this, a variety of features, such as the protocol type, the load length, duration, the maximum message length, the minimum message length, the average message length, and so on, can be extracted for the input of machine learning model. However, unlike the picture file, each feature of the traffic connection represents different physical meanings, and some features are related to others (for example, the minimum and maximum packet length will affect the average length of the packets). Besides, a small change in the number of pixel color values has little impact on the overall picture, while for the traffic connection data, the modification of some key features may lose critical information and weaken the attack ability of the original malicious behavior, therefore, in the process of traffic adversarial examples generating, the direction of the noise that is added to the original data must be strictly controlled.

3.2. The Static of the Noise. As shown in Figure 1, in the field of computer vision, the adversarial example is still a panda in the human vision system (HVS), but after the image is converted into a digital signal on the three color channels of red, green and blue, it can successfully mislead the machine learning-based model to classify the panda as a gibbon. In order to make the adversarial example x^{adv} visually approximately the same as the original picture x , the p norm ($\|\eta\|_p \leq \epsilon$) constraint is usually introduced during the generation of the adversarial example.

However, in the field of IDS, this condition is not suitable. Whether the adversarial example is similar to the original traffic and will cause an exception alarm is not ascertained through visual observation of the traffic data directly, but the network monitoring device, besides most of the machine-learning-based abnormal traffic identification methods, often extracts traffic characteristics, such as protocol type, packet length, and duration of information from traffic data, and then identify malicious behaviors based on these statistical characteristics. Normally, these statistical features correspond to different physical meanings; therefore, when calculating the distance between the adversarial example and the original sample, different statistical features should be based on different influence coefficients ξ . For example, the length of traffic packet change from 500 bytes to 510 bytes does not affect the overall traffic information, but if the protocol type changes from TCP to UDP, it means two completely different traffic data. Therefore, the constraint condition of the noise that is added to the traffic data should be described as

$$\sum_{i=1}^{i=n} \|\xi_i \cdot \eta_i\|_p \leq \epsilon, \quad (6)$$

where n is the number of traffic data features.

4. Black-Box Attack Method

In the black-box attack scenario, the attacker has no information about the structure and parameters of the target model, and the only capability of the attacker is to input the chosen data to the target model and observe results labeled by the target model. Therefore, the current mainstream method of generating adversarial examples is mainly based on the migration of the adversarial examples. As long as both models A and B are trained under similar tasks, the adversarial examples that affect one model tend to affect the other, even if the two models have completely different structure and parameters. Therefore, the attacker only needs to launch attacks on the substitute model in the white-box method and transfer the adversarial examples generate from the substitute model to the target model.

Based on the information of the structure and parameters of the substitute model, the attacker can use any white-box method to craft adversarial examples. Due to the migration of the adversarial examples, the adversarial examples that are effective for the substitute model will also be misclassified by the target model with high probability. Therefore, the black-box adversarial example generation mainly includes two processes:

- (i) *Substitute Model Training*: Based on the same training task and similar database with the target model, we train a substitute model ST that shares the similar decision boundary with the target model
- (ii) *Adversarial Example Generation*: The attacker uses the substitute model ST to craft adversarial examples and then checks whether the adversarial examples will be misclassified by the target model

The black-box attack on the target model is achieved through a white-box attack on the substitute model. In our paper, the white-box method that we used to create abnormal network flow adversarial example \mathbf{x}^{adv} is the extension of BIM [17] and can be expressed as follows:

$$\begin{aligned} \mathbf{x}_0^{\text{adv}} &= \mathbf{x}, \\ \mathbf{x}_{N+1}^{\text{adv}} &= \mathbf{x}_N^{\text{adv}} + D \cdot \left(\alpha O(\nabla_{\mathbf{x}_N} \mathbf{J}(\theta, \mathbf{x}_N, y)) \right), \end{aligned} \quad (7)$$

where \mathbf{x} is the original network flow data, α is the step size, N is the number of iterations, and D is the constraint vector. The constraint vector is used to limit \mathbf{x}^{adv} always change in an allowed direction as the physical constraint of noise. $O(\mathbf{x})$ is a normalization function, which is used to convert the gradient vector into a vector with all values between $[-1, 1]$, and it can be expressed as follows:

$$O(\mathbf{x}) = \begin{cases} \frac{x_i}{\max(\mathbf{x})}, & x_i > 0, \\ 0, & x_i = 0, \\ \frac{x_i}{|\min(\mathbf{x})|}, & x_i < 0. \end{cases} \quad (8)$$

5. Generate Abnormal Network Flow Adversarial Example

In our work, we tried to bypass the machine-learning-based abnormal network flow classifier by adding small but intentionally worst-case perturbations to data from the dataset. To achieve this, we assume that we know nothing about the structure, type, and parameters of the target model, and we can only make a limited number of query accesses to the target model.

In our paper, we first train a substitute model that has similar decision boundaries with the target classifier; then, based on the migration of the adversarial examples, we used the white-box generation method mentioned above and the substitute model to craft adversarial examples. The black-box abnormal network flow adversarial example generate process proposed in this paper is shown in Figure 2, and the main process includes the following parts:

5.1. Dataset. As shown in Figure 2, the dataset S_0 is used for training the target model and generating adversarial examples. We chose the KDD cup 99 and the CSE-CIC-IDS2018 as the datasets in our experiment. The KDD cup 99 is 9 weeks of network connection data collected from a simulated US Air Force LAN and is divided into labeled training data and unlabeled test data. In this dataset, each connection is described by 41 characteristics; among them, there are the basic characteristics of TCP connections (9 types in total), the content characteristics of TCP connections (13 types), the statistical characteristics of time-based network flow (9 types), and the host-based network flow statistics (10 types in total). As shown in Table 2, the dataset contains four attack types, there are Dos, Probing, R2l, and U2r, in the 10% subset of KDD99, DOS attacks accounted for the largest proportion of abnormal attacks, up to 98%. U2r type attacks are the least, only 22. Due to the small amount of U2r and R2L in the training set, both of them are traffic content-based attacks; therefore, in our experiment, these two types of attacks are put in one group. To balance the number of each group, we extracted 1,000 attacks from each group.

Table 2 shows the types of network attacks contained in the KDD99 dataset. The last column is the number of the attacks in the 10% dataset.

In recent years, the IDS2018 has been widely used in the research of network security. The IDS2018 is a diverse and comprehensive benchmark dataset in the field of intrusion detection, and it includes and captures network traffic and system logs of each machine, along with 80 features extracted from the captured traffic, and includes seven different attack scenarios: Heartbleed, Brute-force, Botnet, Web attacks, DoS, DDoS, and infiltration of the network from inside. In our experiment, we summarized all the attack types into: Bot, Dos, Brute, and Infiltration.

5.2. Sampling Algorithm. In the case of the black-box attack, querying the target model too many times can easily attract the attention of defenders; therefore, reducing the query

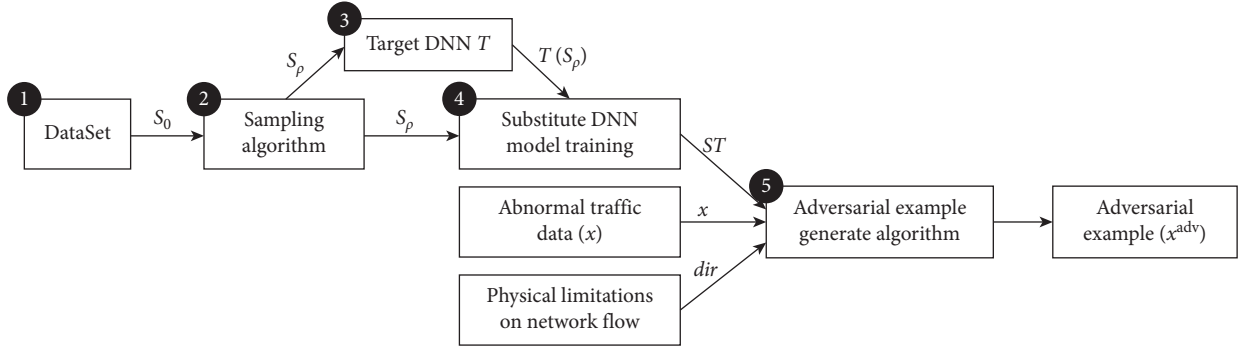


FIGURE 2: Abnormal network flow adversarial example generate process. We (1) choose an initial network flow training set S_0 and (2) generate subdataset S_ρ by sampling algorithm, then (3) label the S_ρ by the target DNN model and (4) train the substitute DNN model ST , finally (5) craft the adversarial example \mathbf{x}^{adv} with the substitute DNN model ST , the abnormal network flow data \mathbf{x} , and the physical limitations on network flow dir .

TABLE 2: Type of malicious attacks in KDD Cup 99.

No.	Types of attack	Attacks in dataset	Quantity
1	DOS	Back, land, Neptune, pod, smurf, teardrop	391458
2	R2L	Ftpwrite, guesspasswd, imap, multihop, phf, spy, warezclient, warezmaster	1126
3	Probing	Ipsweep, nmap, portsweep, Satan	4107
4	U2R	Bufferoverflow, loadmodule, perl, rootkit	22

times to the target model as much as possible can not only improve the efficiency of black-box attacks but also is the key constraint as to whether the black-box attack method can really be implemented in the real network environment.

In this paper, similar to Papernot N et al. [33], we use the reservoir sampling algorithm to reduce the times of query to the target model. The reservoir sampling is a random sampling algorithm, the purpose of which is to select K samples from the set S that contains N items, where N is a large or unknown number. As shown in Algorithm 1, the first K samples of the set S are initially taken as the sampling result, and then go through the other samples in set S . When the i -th sample is taken, the selection strategy is to generate a random number r in the range $[0, K + i - 1]$. If r is less than K , replace the r -th sample in the sampling result set $R[K]$ to the i -th sample in the dataset S . If r is greater than or equal to K , continue the iteration. After iteration through all the data, return these K samples. This algorithm makes the probability of all samples in the set selected to be equal under the premise of only accessing the data stream once. Using the reservoir sampling algorithm can greatly reduce the number of queries to the target model and improve the training efficiency of the substitute model.

As shown in step 2 of Figure 2, the original dataset is S_0 , and the subset S_ρ is obtained after the sampling algorithm. Using S_ρ as a training set, query the target model to label it as a training set, and that can be used to train the substitute model with similar decision boundaries on the limited number of the dataset.

5.3. Substitute Model Training. In the process of generating adversarial examples, we use the gradient information of the substitute model as the direction to craft adversarial

example. It is required that the substitute model should have a similar decision boundary as the target model. From the point of the black-box attacker, we know nothing about the structure and parameters of the target model. However, since we can query the target model, we can estimate the approximate information of the input layer and output layer of the target model by observing the input and output of the target model; then, we can design the structure of the substitute model.

The research of Papernot N et al. [19] showed that the substitute model and the target model only need to go through a similar training process during the generation of the adversarial examples, and it is not necessary to have the same network structure and parameters. In this paper, we choose Multi-Layer Perceptron (MLP) network as the structure of the substitute model. The number of neurons of the input layer corresponds to the number of features in the traffic data, and the number of neurons of the output layer corresponds to the number of attacks of the traffic data. As shown in Figure 2, the substitute is training on the dataset S_ρ , which is the subset of the original dataset S_0 and is labeled by the target model.

5.4. Generate Adversarial Example. In the field of computer vision, the process of adversarial example generate is to find the appropriate perturbation η to satisfy the following conditions:

$$F(\mathbf{x} + \eta) \neq F(\mathbf{x}), \quad (9)$$

$$\|\eta\|_p \leq \epsilon,$$

where $F(\cdot)$ is the target model, \mathbf{x} is the input picture, and η is the perturbation added to the original picture when the

Input: $S[N]$, K , where S is the sample set, N is the sample size, and K is the number of samples
Output: $R[K]$, where R is the set of sampling results

- (1) set $R(K) \leftarrow X(K)$
- (2) **for** $i \in [K, N - 1]$ **do**
- (3) $r \leftarrow$ random integer between $[0, K + i - 1]$
- (4) **if** $r < k$ **then**
- (5) $R[r] \leftarrow S[i]$
- (6) **end if**
- (7) **end for**
- (8) **return** $R[K]$

ALGORITHM 1: Reservoir sampling algorithm.

p – norm of the perturbation η is less than ϵ ; it means that the perturbation is not perceptible to the human eye.

As mentioned above, we must ensure that the network flow adversarial examples remain in its attack function and has no key information lost, otherwise, it has no practical significance. In the field of computer vision, the modification of any pixel on the picture will not have a greater impact on the content of the picture. Therefore, whether the perturbation can be perceived by the human eye is mainly measured by calculating the p – norm of the noise η , but in the field of IDS, different characteristics have different effects on the overall network connection properties. The change of some features, such as the type of network connection, will cause the fundamental change of network connection, and changes in some features will cause the network connection information not to conform to the physical properties. Therefore, the adversarial example generation process in the network security field is subject to the following constraints:

- (i) Whether the magnitude of perturbation can be detectable is not decided by a person, but by the network devices
- (ii) Compared to the original connection, the adversarial example cannot lose the key information of the network connection, which determines that the direction of the perturbation η added to the original connection must be strictly restricted
- (iii) The adversarial example must retain the attack function of the original connection

To address the above issues, as described above, Lin Z et al. [31] try to keep the attack function by adding unmodified features to the model, which means only add perturbation to the features with less influence. However, this method only limits some features that cannot be modified but does not limit the direction of the perturbation added to the modifiable feature. This may distort the original connection information; for example, if the original connection contains 1000 bytes of data, and the adversarial example has only 990 bytes, it will cause 10 bytes data distortion compared to the original connection.

As shown below, in this paper, we address this issue by two measures, and this is the primary content of constraint vector D in equation (7).

- (i) Strictly limit the number of modifiable features in the process of adversarial examples generation. In this paper, for the features extracted from the network flow data, we only add perturbation to the noncritical features, such as the length of the packets, the duration of the connection, and the length of the package interval.
- (ii) For modifiable features, the direction of perturbation added to the original connection is strictly limited to avoid information distortion. For features that can be modified, we only add positive perturbation to the original data. For instance, for the length of packets, we only add perturbation in the direction in which the length of the packet grows.

We now describe the network flow adversarial example generate process outlined in Algorithm 2, which is as follows.

- (1) Initially, set the adversarial example \mathbf{x}^{adv} as the original connection input \mathbf{x} .
- (2) In the iterative process, first calculate the cross-entropy L of the original label of network flow information l and the label $\text{ST}(\mathbf{x}^{\text{adv}})$, and then calculate the gradient G of L at the sample \mathbf{x}^{adv} .
- (3) Calculate the perturbation η added in this iteration:

$$\eta = D \cdot (\alpha O(G)), \quad (10)$$

where $O(\cdot)$ is a normalization function (equation (8)). α is the step size of the sample moving along the gradient direction, the larger the α is, the greater the noise is added in a single iteration, and D is the constraint vector of the perturbation.

- (4) Add the perturbation η generated in the iteration to \mathbf{x}^{adv} . If the substitute model is successfully deceived or the noise generated in the current iteration is 0, stop the iteration process and return the adversarial example \mathbf{x}^{adv} .

6. Results

6.1. Target Models. To evaluate the capacity of our model comprehensively and deeply, we first trained several typical abnormal network flow classification models based on the

Input: $ST, T, \mathbf{x}, N, l, \alpha$, where ST is the substitute model, T is the target model, \mathbf{x} is the network flow data, N is the iteration steps, l is original label, and α is the move step

Output: traffic adversarial example \mathbf{x}^{adv}

```

(1) set  $\mathbf{x}^{\text{adv}} = \mathbf{x}$ 
(2) for  $i \in [0, N - 1]$  do
(3)   loss  $L = \text{cross\_entropy}(l, ST(\mathbf{x}^{\text{adv}}))$ 
(4)   gradient  $G = \nabla_{\mathbf{x}^{\text{adv}}} L$ 
(5)   perturbation  $\eta = D \cdot (\alpha O(G))$ , where  $O(\cdot)$  is shown in equation (8)
(6)   set  $\mathbf{x}^{\text{adv}} = \mathbf{x}^{\text{adv}} + \eta$ 
(7)   if  $ST(\mathbf{x}^{\text{adv}}) \neq l$  and  $T(\mathbf{x}^{\text{adv}}) \neq l$  then
(8)     break
(9)   end if
(10)  if  $\mathbf{x}^{\text{adv}} = \mathbf{x}$  then
(11)    break
(12)  end if
(13) end for
(14) return  $\mathbf{x}^{\text{adv}}$ 

```

ALGORITHM 2: Network flow adversarial example generate algorithm.

10% subset of the KDD99 dataset and the IDS2018 dataset, respectively. The adopted algorithms of the black-box IDS in the experiments include Convolutional Neural Networks (CNN), Support Vector Machines (SVM), k-Nearest Neighbor (KNN), Multilayer Perceptron (MLP), and the Residual Network (Resnet).

To verify the validity of the network follow adversarial example, we randomly select 1000 samples from the data of various abnormal attacks and label it with the target model. Based on these attack data, we use the method proposed in this paper to generate adversarial examples and query the classification results of the target model for these adversarial examples. Then, we use the recall rate to evaluate the effectiveness of the adversarial examples, the lower the recall rate, the more effective the adversarial examples are.

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (11)$$

where TP is the number of instances that were correctly classified and FN is the number of instances that are misclassified by the model.

6.2. Attack Based on White-Box. To verify the effectiveness of the attack method proposed in this paper, we carried out a white-box attack experiment with our method in this section. In the experiment, we chose CNN as the target model. Firstly, we used the KDD Cup 99 dataset to train a CNN-based malicious traffic detection model. Then we randomly selected 1000 records from the Dos attacks, the U2r&R2l attacks, the Probing attacks, and the Normal network connections, respectively. Finally, we used the method proposed in this paper and the 4000 records extracted from the original dataset to craft adversarial examples, we use the target CNN model to label the original data and the adversarial examples, and the confusion matrix is shown in Figure 3.

As shown in Figure 3(a), for the original dataset, the CNN-based malicious traffic detection model can make accurate classifications for different types of attacks, with an accuracy rate of about 98%, which can well complete the detection of network flow. However, for the generated adversarial examples, as we can see from Figure 3(b), the target CNN model has the highest detection accuracy for different types of network traffic and only 27.2% for a four-class detection model, which is completely unusable. The method of adversarial example generation proposed in this paper can significantly reduce the classification accuracy of the machine-learning-based network abnormal traffic detection model. For Dos attacks, about 74% of the attack connections can successfully bypass the detection of the target model, and for other types of abnormal traffic connections, the effect is similar.

During the experiment, we also adjusted the step size of the perturbation in the single iteration, and the results are shown in Figure 4. When the step size is set to 1, the mean recall rate of the network flow detector is about 33%. With the increase of the step size, when the step is above 3, the recall rate of the model stabilized at 25% or so, and increase in the step size has little effect on the success rate of adversarial example generation.

6.3. Attack Based on Black-Box

6.3.1. The Substitute Model. As mentioned above, we choose the Multi-Layer Perceptron (MLP) network as the structure of the substitute model. The fact that the substitute model and the target model have similar decision boundaries is a key point for the success rate of our method. Here, we use SCR to evaluate the similarity of decision boundaries between the substitute model and the target model, as shown below. The higher the SCR value, the more similar the decision boundaries of the substitute model and the target model.

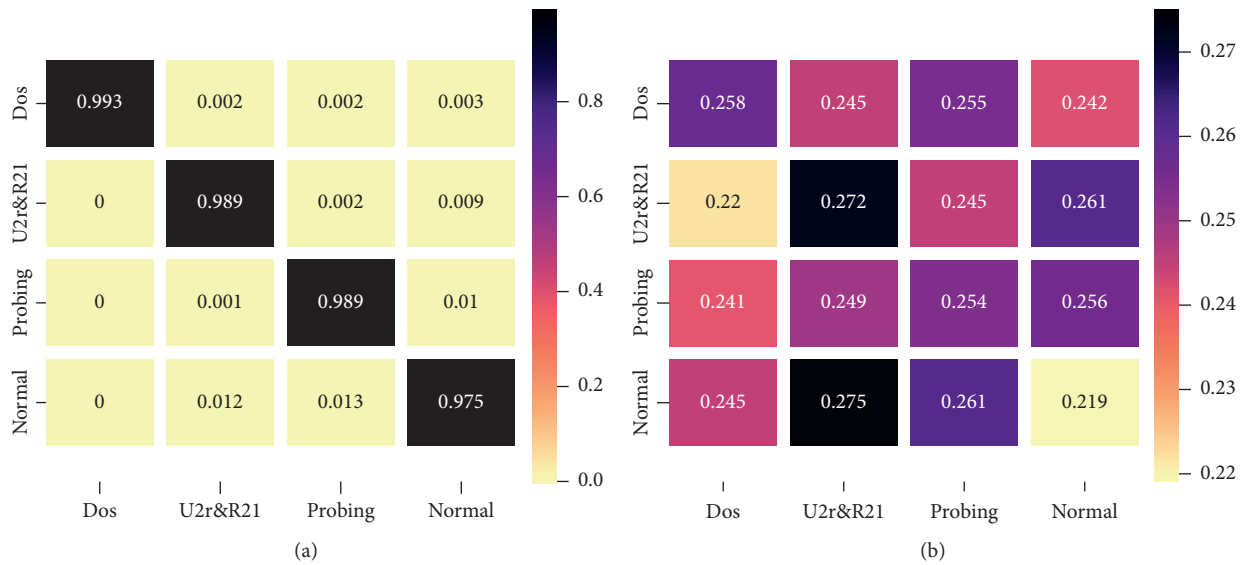


FIGURE 3: The confusion matrix of the target model. (a) The confusion matrix of the original traffic data. (b) The confusion matrix of the adversarial examples.

$$\text{SCR} = \frac{\text{Number of same classified by the substitute model and target model}}{\text{Total number of the special attacks}} \quad (12)$$

For different target models, the SCR values of the substitute model for different types of attacks are shown in Figure 5. In the dataset used for substitute model training, normal network connections and Dos attacks account for a higher proportion, and its SCR values are all around 99%. On the contrary, Probing, U2l, and R2l account for a relatively low percentage in the dataset, and their SCR values are relatively low. For the U2r & R2l group with the lowest SCR, the minimum SCR value is 50% and the maximum is 70%. However, since the number is very small in the whole dataset, it has little effect on the total SCR value. In general, the substitute model still has a high similarity decision boundary with the target model.

6.3.2. Model Attack. Based on the types of black-box malicious network flow detection models that we had trained, and the method we used in this paper, for the KDD99 dataset, the attack results are shown in Figure 6. The lower Recall of the adversarial examples under various attacks reflect the great capacity of the adversarial attack in the experiments.

As shown in Figure 6(a), the mean Recall of these malicious traffic detection models is 91.8%, which means that all of these models can very well identify malicious attacks.

As shown in Figure 6(b), the average Recall of DoS under all detection algorithms is 19.8%. The results show the excellent performance of our black-box attack method in DoS. Preferably, for the case of MLP, more than 94.2% of the adversarial DoS network flow examples can evade the detection of the IDS model in each test.

For the case of the Probing, the average Recall is 32.7%. Although KNN shows better robustness, there are still a large number of malicious attacks that evade detection of the target model. On average, about 67.3% of the Probing network flow adversarial examples can bypass the detection of the target model.

And, in the worst case of U2R & E2L, the average Recall of U2R & E2L under all detection algorithms is 42.5%, which means that about 57.5% of the U2R & E2L network flow adversarial examples can evade the detection of the target model in average.

For the IDS2018 dataset, as mentioned above, we summarized all the attack types into: Bot, Dos, Brute, and Infiltration. Based on this, we trained three kinds of malicious traffic detection models: the MLP, CNN, and ResNet. The Recall of these malicious traffic detection models is shown in Figure 7(a), as we can see that all of these models can very well identify malicious attacks with a mean Recall reach of 90%. Similarly, we randomly choose 1000 samples from each malicious attack and label them with the target model. Then, we generate adversarial examples with our method, and the results are shown in Figure 7(b). For the MLP-based detection model, an average of 72.2% of malicious traffic data can successfully bypass the detection of the target model. Among them, the Dos attacks with high success rate can successfully deceive the target model with 87% probability, and the Bot attacks with low success rate also have 52.5% probability. For CNN and the ResNet-based detection model, an average of 70% and 71% of malicious traffic attack can successfully bypass the detection of the target model, respectively, and among them, 99.9% of bot attacks can successfully bypass the detection of the target

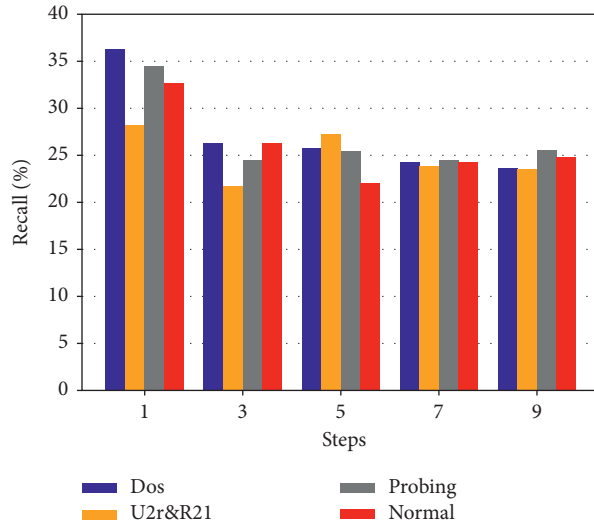


FIGURE 4: The impact of step size on adversarial attack.

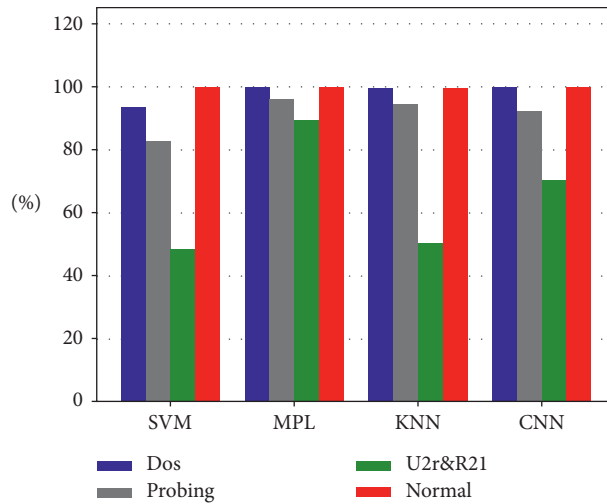


FIGURE 5: The SCR values of the substitute model.

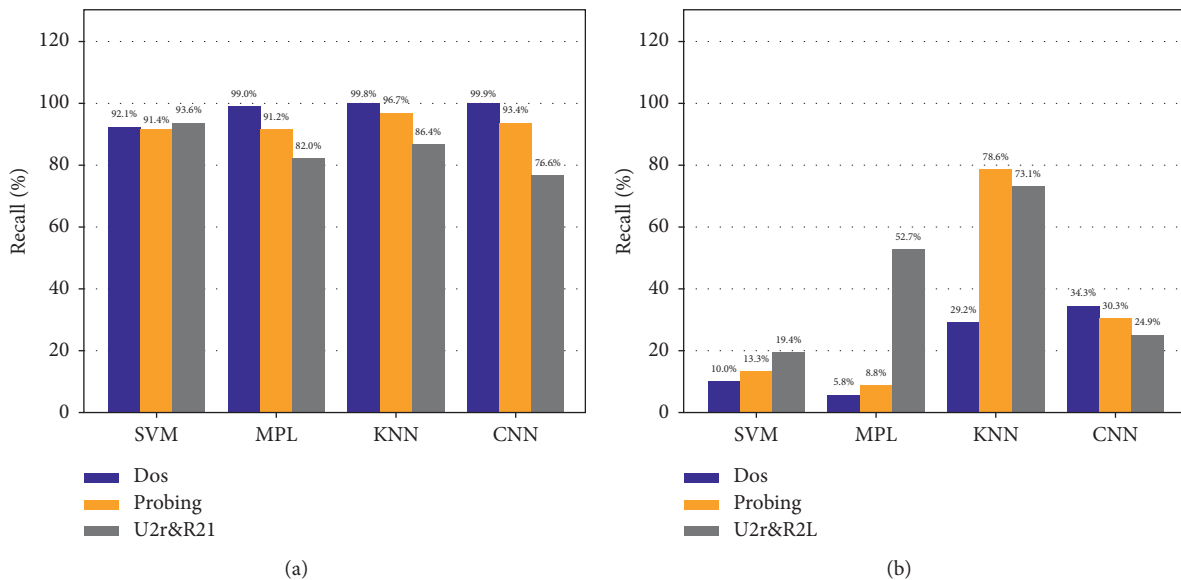


FIGURE 6: The recall rate of the KDD99 dataset and its adversarial examples. (a) The recall rate of the original network flow data. (b) The recall rate of the network flow adversarial example.

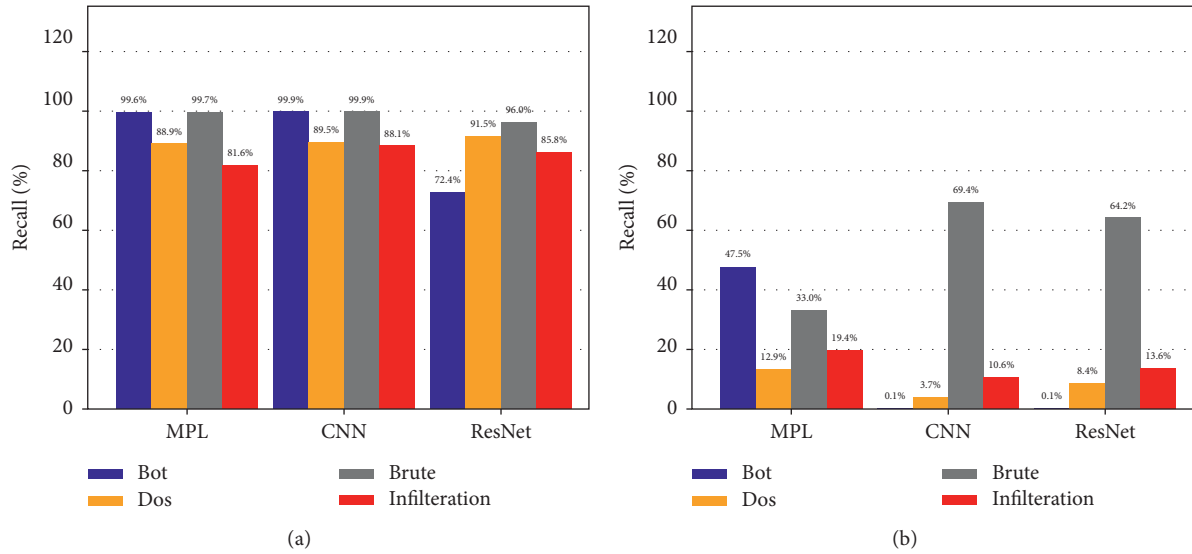


FIGURE 7: The recall rate of the CSE-CIC-IDS2018 dataset and its adversarial examples. (a) The recall rate of the original traffic data. (b) The recall rate of the traffic adversarial example.

model. The adversarial examples show weak attacks against Brute attacks, but more than 30% of the traffic data successfully bypass the detection of the target model.

6.3.3. *Effect of Sampling Rate on Black-Box Attack.* In this paper, we launch attacks on the substitute model in the white-box model and then apply the adversarial example to the target model. The success rate of this method mainly depends on the similarity of the gradient information and decision boundary between the substitute model and the target model. As shown in step 2 of Figure 2, the substitute model is trained on S_p which is the subset of S_0 , and the sampling rate is the proportion of S_p in S_0 . The larger the sampling rate, the closer S_p is to S_0 , and the more likely it is that the substitute model and the target model will have similar decision boundaries. Based on this, we test on the Kdd99 dataset, and take CNN as the black-box IDS model. Different sampling rates are used in the sampling algorithm to generate network flow adversarial examples, the result of which is shown in Figure 8:

As shown in Figure 8, when the sampling rate is set to 10%, the adversarial examples for Dos can largely bypass the detection of the target model. However, the performance of the other two types is poor because DOS occupies a large proportion in the dataset. When the sampling rate is small, the proportion of probing and U2r&R2l in the sub-dataset used for training the substitute model will be smaller, and the substitute model cannot have very similar decision boundaries with the target model. When the sampling rate is more than 30%, the mean probability of the adversarial examples of various attacks escaping the detection of the target model does not change much. Therefore, our method can generate the network flow adversarial example effectively, even if the capacity of the dataset used for training the substitute model is relatively small.

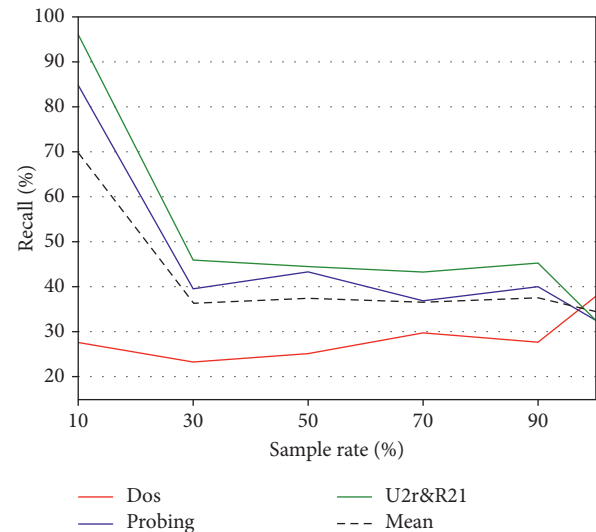


FIGURE 8: The effect of sampling rate on black-box attack.

6.3.4. *Effect of Step Size on Black-Box Attack.* As described in step 3 of Figure 2, in the process of generating abnormal network flow adversarial examples, the amount of perturbation added to the original network flow data in a single iteration depends on the gradient and the step size α . A proper step size α can quickly generate effective adversarial examples. Based on this, we test on the Kdd99 dataset, take CNN as the black-box IDS model, and then use different step sizes α to craft abnormal network flow adversarial examples, the results of which are shown in Figure 9.

As shown in Figure 9, in the process of generating abnormal network flow adversarial example, take probing as an example. When the step size changes from 1 to 17, the recall rate decreases from 85% to nearly 30%. When α is 5 or 9, the average Recall is going to be the lowest, about 33%, which means more than 67% abnormal network flow examples can bypass the

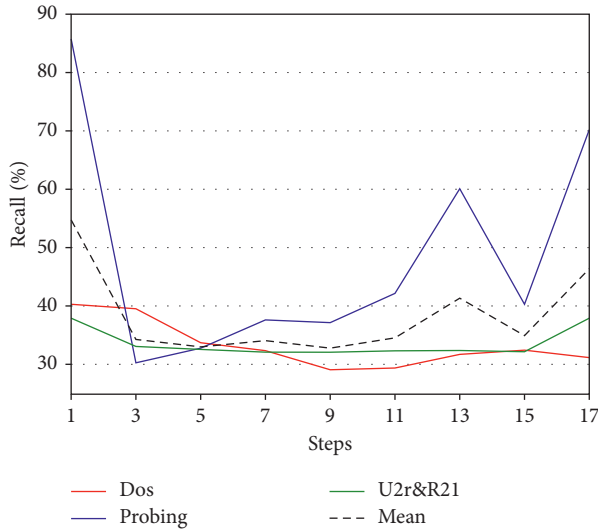


FIGURE 9: The effect of step size on black-box attack.

detection of the target machine-learning-based model. So it can be seen that an appropriate step size has a big influence on the success rate of adversarial example generation.

7. Conclusion

In this paper, we made a detailed comparison of the adversarial example generation technology between the field of computer vision and IDS, and we analyzed the key points and corresponding solutions for making adversarial examples in the field of IDS. Firstly, we train a substitute model with a similar decision boundary with the target model on the KDD99 dataset and the CSE-CIC-IDS2018 dataset, and then extend the BIM algorithm to craft adversarial examples with the structure and parameters of the substitute model. Finally, we check whether the adversarial examples can bypass the detection of the target model or not. Experiments show that our method can effectively generate network flow adversarial examples that can be applied to the real world and can successfully fool most of the machine-learning-based detection models.

In the future, we will further focus on the research of adversarial example technology in the field of cybersecurity. The research will concentrate on two aspects: first, we will directly apply the algorithm to real network traffic packets; Secondly, we will study the more complex malicious attack adversarial example technology based on multi-sensor data on network devices.

Data Availability

The dataset used in our paper can be made available at <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html> and <https://www.unb.ca/cic/datasets/ids-2018.html>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by National Key R&D Program of China (Grant No. 2020AAA0107700), in part by the Natural Science Basic Research Plan in Shaanxi Province of China (Grant No. 2020JQ-214), in part by the State Grid Gansu Electric Power Company Science and Technology Projects (Grant 52272219100Q), and in part by the Natural Science Foundation of Jiangsu Higher Education Institutions of China (Project no. 17KJB413001).

References

- [1] D. Hu, "An introductory survey on attention mechanisms in NLP problems," in *Proceedings of the SAI Intelligent Systems Conference*, pp. 432–448, London, UK, September 2019.
- [2] M. S. Mahdavejad, M. Rezvan, M. Barekatin, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for Internet of Things data analysis: a survey," *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, 2018.
- [3] J. Xiong, M. Zhao, M. Z. A. Bhuiyan et al., "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2019.
- [4] L. Schmarje et al., "A survey on semi-, self- and unsupervised techniques in image classification," 2020, <https://arxiv.org/abs/2002.08721>.
- [5] Y. Li, S. Wang, Y. Ma et al., "Popularity prediction on vacation rental websites," *Neurocomputing*, vol. 412, 2020.
- [6] Y. Li, S. Wang, T. Yang, Q. Pan, and J. Tang, "Price recommendation on vacation rental websites," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 399–407, Westin Galleria Houston, TX, USA, April 2017.
- [7] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," in *Proceedings of the 8th international conference on software, knowledge, information management and applications (SKIMA 2014)*, pp. 1–6, New York, NY, USA, December 2014.
- [8] B. B. Rao and K. Swathi, "Fast kNN classifiers for network intrusion detection system," *Indian Journal of Science and Technology*, vol. 10, no. 14, pp. 1–10, 2017.
- [9] C. Azad, V. K. Jha, and V. Kumar Jha, "Genetic algorithm to solve the problem of small disjunct in the decision tree based intrusion detection system," *International Journal of Computer Network and Information Security*, vol. 7, no. 8, p. 56, 2015.
- [10] M. Z. Alom, V. R. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *Proceedings of the NAECON 2015-IEEE National Aerospace and Electronics Conference*, Dayton, OH, USA, June 2015.
- [11] Y. Chuan-Long, Z. Yue-Fei, F. Jin-Long et al., "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 99, p. 1, 2017.
- [12] W. Wang, M. Zhu, X. Zeng et al., "Malware traffic classification using convolutional neural network for representation learning," in *Proceedings of the 2017 International Conference on Information Networking (ICOIN)*, pp. 712–717, Da Nang, Vietnam, January 2017.
- [13] Y. Xin, L. Kong, Z. Liu et al., "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, p. 1, 2018.

- [14] J. H. Davis and J. R. Cogdell, *Calibration Program for the 16-Foot Antenna*, Electrical & Computer Engineering Research Laboratories, Austin, TX, USA, 1987.
- [15] C. Szegedy, W. Zaremba, I. Sutskever et al., "Intriguing properties of neural networks," 2013, <https://arxiv.org/abs/1312.6199>.
- [16] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *EnCase Computer Forensics*, vol. 6, 2014.
- [17] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proceedings of the International Conference on Learning Representations*, Toulon, France, April 2017.
- [18] A. Kurakin, I. Goodfellow, S. Bengio et al., "Adversarial examples in the physical world," in *Proceedings of the international conference on learning representations*, Toulon, France, April 2017.
- [19] N. Papernot, P. Mcdaniel, I. Goodfellow et al., "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM*, Singapore, November 2017.
- [20] P. Chen, H. Zhang, Y. Sharma et al., "ZOO: zeroth Order Optimization Based Black-Box Attacks to Deep Neural Networks without Training Substitute Models," *arXiv: Machine Learning*, vol. 7, pp. 15–26, 2017.
- [21] J. Su, D. V. Vargas, and S. Kouichi, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, 2017.
- [22] Z. Zhao, D. Dua, and S. Singh, "Generating natural adversarial examples," 2017, <https://arxiv.org/abs/1710.11342>.
- [23] Y. Li, L. Li, L. Wang et al., "Nattack: learning the distributions of adversarial examples for an improved black-box attack on deep neural networks," in *Proceedings of the International Conference on Machine Learning. PMLR*, pp. 3866–3876, Sydney, Australia., June 2019.
- [24] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: reliable attacks against black-box machine learning models," 2017, <https://arxiv.org/abs/1712.04248>.
- [25] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.
- [26] X. Liu, L. Xie, Y. Wang et al., "Privacy and security issues in deep learning: a survey," *IEEE Access*, vol. 10, 2020.
- [27] W. Hu and Y. Tan, "The robustness of machine learning based malware detection algorithms," in *Proceedings of the IEEE 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1435–1441, Anchorage, AK, USA, April 2017.
- [28] K. Grosse, N. Papernot, P. Manoharan et al., "Adversarial examples for malware detection," in *European symposium on research in computer security*, pp. 62–79, Springer, Cham, Switzerland, 2017.
- [29] H. S. Anderson, J. Woodbridge, and B. Filar, "DeepDGA: Adversarially-tuned domain generation and detection," 2016.
- [30] K. Yang, J. Liu, C. Zhang et al., "Adversarial examples against the deep learning based network intrusion detection systems," in *Proceedings of the Military Communications Conference*, pp. 559–564, Los Angeles, CA, USA, October 2018.
- [31] Z. Lin, Y. Shi, and Z. Xue, "IDSGAN: generative adversarial networks for attack generation against intrusion detection," 2018, <https://arxiv.org/abs/1809.02077>.
- [32] M. J. De Lucia and C. Cotton, "Adversarial machine learning for cyber security," *Journal of Information Systems Applied Research*, vol. 12, no. 1, p. 26, 2019.
- [33] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," 2016, <https://arxiv.org/abs/1605.07277>.

Research Article

Security Analysis of a Lightweight Identity-Based Two-Party Authenticated Key Agreement Protocol for IIoT Environments

Yuting Li ^{1,2}, Qingfeng Cheng ^{1,2} and Wenbo Shi³

¹State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

²Strategic Support Force Information Engineering University, Zhengzhou 450001, China

³School of Computer and Communication Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China

Correspondence should be addressed to Qingfeng Cheng; qingfengc2008@sina.com

Received 8 January 2021; Revised 2 February 2021; Accepted 22 February 2021; Published 28 February 2021

Academic Editor: Jinbo Xiong

Copyright © 2021 Yuting Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Internet of Things brings convenience to the social life, at the same time, putting forward higher requirements for the security of data transmission and storage. Security incidents based on industrial Internet of Things have occurred frequently recently, which should be given full consideration. The identity-based authenticated key agreement protocol can solve these security threats to a certain extent. Recently, a lightweight identity-based authenticated key agreement protocol for Industrial Internet of Things, called ID-2PAKA protocol, was claimed to achieve secure authentication and meet security properties. In this paper, we show that the ID-2PAKA protocol is insecure in identity authentication and cannot resisting ephemeral key compromise impersonation attack.

1. Introduction

The application field of the Internet of Things is very extensive, especially in the industry [1]. As increasingly more devices such as sensors are connected together [2], related industries are getting closer and integrated with the Industrial Internet of Things (IIoT). IIoT can be regarded as a high degree of integration of industrial automation systems and IoT systems. With the explosive growth of industrial information, the large amount of data generated in the industrial production is a challenge for IIoT. How to effectively process, analyze, and record these data, and extract the results of guiding suggestions for industrial production, is the core difficulty of IIoT [3].

The system architecture of IIoT is shown in Figure 1. The perception layer is composed of widely deployed physical devices (such as sensors, actuators, manufacturing equipment, facility utilities, and other industrial manufacturing and automation related objects) and is responsible for real-time collection of industrial environment and production resource data. The network layer makes short-distance access and long-distance transmission of perception data a reality, while the data processing layer is for fully mining and

utilizing the aggregated perception data. The application layer is composed of various industrial applications, including smart factories and smart supply chains. These intelligent industrial applications utilize numerous sensors and actuators to achieve real-time monitoring, precise control, and effective management.

With attendant, incidents based on IIoT security have occurred frequently recently. For intruders, attacks on IIoT systems can attract more attention or get more than attacks on IoT systems in other industries. Attackers have adopted a variety of intrusion methods, such as the leakage of industrial key data, and the illegal hijacking and manipulation of interconnected terminals [4]. The IIoT relies on modern and mature industrial automation systems and integrates a large number of technologies and applications from the fields of communications and computers. The wide application of the IoT puts forward more strict security requirements for data transmission and storage. Therefore, some traditional network attack methods are also suitable for IIoT systems. A large number of attacks have occurred in the past few years. Exposing the various hidden dangers of IIoT in terms of information security is a major obstacle to the rising trend of IoT.

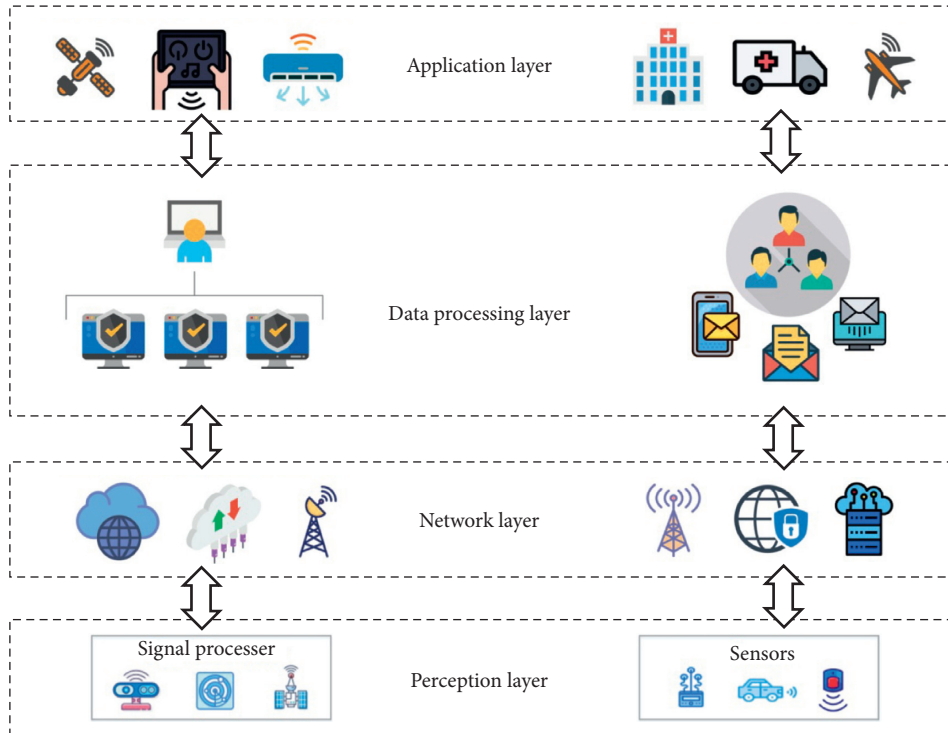


FIGURE 1: IIoT system architecture.

Specifically, the security threats faced by IIoT can be divided into two categories, namely, the hidden dangers of the internal structure of IIoT and the hidden dangers of external network attacks. Among them, attacks against external networks have the characteristics of wide coverage, multiple levels, and diverse attack methods. The solutions to these security problems usually use a mixture of computing, encryption, image processing, and identity authentication.

Applying cryptography to network communication can solve these security threats to a certain extent. Cryptography realizes the encryption, decryption, user identity authentication, key agreement, and privacy protection of important information through strict mathematical theories. It is one of the important means to protect communication security. The key agreement protocol is an important branch of cryptography, which refers to the rule that two or more parties in communication negotiate a symmetric encryption key on a common channel before formal communication. The key agreement protocol determines the security of the symmetric encryption key and thus determines the information security of the communication participants. Therefore, the study of session key agreement protocol can strengthen the security of the network to a certain extent, and it is of great significance to the protection of personal privacy and commercial interests.

Traditional key agreement protocols use certificates to authenticate the participants of the protocol, which are easy to be forged and tampered with. Therefore, the traditional session key agreement protocol still has certain deficiencies in security. The identity-based authenticated key agreement (ID-AKA) protocol integrates identity authentication into the key agreement process, avoiding the use of digital

certificates and improving the security of the key agreement protocol [5, 6]. According to whether bilinear pairing is used in the ID-AKA protocol, it can be divided into the ID-AKA protocol based on bilinear pairing and the ID-AKA protocol without bilinear pairing. Although the ID-AKA protocol without bilinear pairing has an advantage over the ID-AKA protocol based on bilinear pairing in terms of computational efficiency, the ID-AKA protocol without bilinear pairing is not satisfactory in terms of security [7]. Bilinear pairing operation is a computationally intensive operation, so ID-AKA protocol based on bilinear pairing has obvious shortcomings in computational efficiency. This affects the comprehensive performance of the ID-AKA protocol based on bilinear pairs and also seriously affects its practical application range [8].

In this paper, we analyze the ID-2PAKA protocol for IIoT environments from [9] in terms of a security perspective and discover some insecure threats. When the protocol is analyzed, it is insecure in terms of identity authentication. Moreover, there were some threats in resistance to ephemeral key compromise impersonation attack.

The organization of this paper is arranged as follows. Related works are firstly introduced in Section 2. Then, we briefly review the ID-2PAKA protocol in Section 3. Furthermore, Section 4 points out the weaknesses of the ID-2PAKA protocol. Conclusion will be given in Section 5.

2. Related Work

In recent years, cyberattacks against industrial IoT systems have emerged one after another, showing a continuous upward trend. The security issues of industrial IoT systems

have attracted great attention in the information security industry.

In view of the security issues of the IoT, a large number of security mechanisms have been proposed [10, 11], especially the wireless sensor network as an important supporting technology of IoT. In [12], in response to the vulnerability of wireless sensor network nodes and limited resources, Zhou and Xiong propose a lightweight smart card-based wireless sensor network user authentication scheme, which is based on random values as temporary keys. Through the request-response handshake mechanism to ensure the two-way authentication between the user and the gateway node, this solution avoids the problem of asynchrony between the smart card and the gateway node. The literature [13] presents a two-factor authentication protocol that provides a powerful authentication and session key establishment process. The protocol resists the threat of multiple users logging in with the same identity. The authentication process does not require public key operations, and it uses a cryptographic hash function to achieve higher efficiency.

The literature [14] proposes a new method adapted to resource-constrained wireless sensor networks. Only legitimate users can access node resources, and illegal users are denied access. The solution is based on ID technology and elliptic curve cryptosystem (ECC), which provides mutual authentication and key agreement processes between users and nodes. In [15], Liu et al. analyze the wireless sensor network in the perception layer of the IoT and propose an identity authentication scheme for the wireless sensor network. The scheme uses ECC, protecting the data confidentiality and integrity of the perception layer of the IoT. However, this scheme only protects the data security of the perception layer of the IoT system and does not protect the IoT terminal devices at the perception layer.

At present, many key agreement protocols for the IoT environment pay more attention to lightweight requirements [16, 17]. In 2016, Farash et al. [18] improved the key agreement protocol based on heterogeneous sensor network proposed by Turkanovic. The improved version can strengthen the security level. Srinivas et al. [19] proposed a chaotic mapping-based key agreement protocol for IIoT environment. However, the author uses a weaker model to prove the protocol; thus, there is still room for further improvement in the security of the protocol.

In addition to the traditional key agreement protocol, some other methods have also been introduced into the field of IIoT security protection. Recently, Xiong et al. [20] combined data encryption with game theory, designing a personalized privacy protection framework. The advantage is to find a reasonable balance between retaining quality of crowdsensing services and privacy. Besides, in order to solve the key management problem of dynamic wireless sensor networks in IIoT, Tian et al. [21] presented a key management scheme based on blockchain. This scheme used stake blockchain to replace the base station to implement key management, avoiding the security threats of untrusted base stations. The summary of literature studies is given in Table 1.

3. Review of ID-2PAKA Protocol

A brief introduction of ID-2PAKA protocol will be given in this section. It consists of three phases: setup phase, private-key generation phase, and session key agreement phase. The notations and the corresponding meanings used in ID-2PAKA protocol are shown in Table 2.

There are three entities participating in ID-2PAKA protocol: the initiator P_1 , the responder P_2 , and the PKG. Among them, the PKG is only responsible for generating the identity-based private key of P_i ($i = 1, 2$). Other details can be depicted in the following subsections.

3.1. Setup Phase. In setup phase, the PKG generates the system parameters according to the security parameter k :

- (1) With a given security parameter k , the PKG chooses a prime number q greater than 2^k , then generates an additive cyclic group G_1 , and a multiplicative group G_2 of order q . The generator of G_1 is P .
- (2) The PKG chooses a bilinear map $e: G_1 \times G_1 \rightarrow G_2$.
- (3) The PKG chooses two one-way hash functions H_i ($i = 1, 2$): $\{0, 1\}^* \rightarrow \{0, 1\}^q$.
- (4) The PKG randomly chooses a master private key $s_0 \in Z_q^*$ and computes the master public key $P_0 = s_0P$.
- (5) The system parameters are set as $\{q, G_1, G_2, P, e, H_1, H_2, P_0\}$, public to all entities.

3.2. Private-Key Generation Phase. In this phase, the identity-based private keys and the corresponding public keys of P_i ($i = 1, 2$) are generated by the PKG. The main details are shown in Figure 2:

- (1) P_i ($i = 1, 2$) submits the identity ID_i ($i = 1, 2$) to the PKG.
- (2) The PKG first authenticates the legality of ID_i ($i = 1, 2$), then computes the public key $q_i = H_1(ID_i)$ and the identity-based private key $Pr_i = (s/(s + q_i))$.

3.3. Session Key Agreement Phase. This phase is executed between the initiator P_1 and the responder P_2 . The details are described in Figure 3:

- (1) The initiator P_1 chooses a random number $r_1 \in Z_q^*$, then computes $\psi_1 = r_1P$ and $\sigma_1 = r_1Pr_1$. Then, P_1 sends the tuple $\{\psi_1, \sigma_1\}$ to the responder.
- (2) After receiving $\{\psi_1, \sigma_1\}$ from P_1 , the responder P_2 chooses a random number $r_2 \in Z_q^*$, then computes $\psi_2 = r_2P$ and $\sigma_2 = r_2Pr_2$. Finally, P_2 sends the tuple $\{\psi_2, \sigma_2\}$ to P_1 .
- (3) After receiving response of P_2 , P_1 first verifies whether the equation $e(\sigma_2, P_0 + q_2P) = e(\psi_2, P_0)$ holds, where $q_2 = H_1(ID_2)$. If verified, P_1 computes $X = r_1\psi_2$ and sets the session key as $sk_1 = H_2(ID_1 \| ID_2 \| \psi_1 \| \psi_2 \| X)$.

TABLE 1: The summary of literature studies.

Literature studies	Description	Application
[10, 11]	Security mechanisms	For wireless sensor networks
[12]	A lightweight smart card-based authentication scheme	For wireless sensor networks
[13]	A two-factor authentication protocol	For wireless sensor networks
[14]	Uses ID technology and elliptic curve cryptosystem	For resource-constrained wireless sensor networks
[15]	Protects the data security	For the perception layer
[16, 17]	Key agreement protocols	For lightweight IoT environment
[18]	An improved key agreement protocol	For heterogeneous sensor network
[19]	Uses chaotic mapping	For IIoT environment
[20]	Combines data encryption with game theory	For privacy protection in IIoT
[21]	Uses stake blockchain	For dynamic wireless sensor networks

TABLE 2: The notations.

Notations	Meanings
k	Security parameter
G_1	An additive cyclic group
G_2	A multiplicative group
q	The prime order of G_1 and G_2
P	The generator of G_1
s_0	The master private key
P_0	The master public key
$H_i (i = 1, 2)$	The secure hash functions
$P_i (i = 1, 2)$	The users

(4) In the same way, P_2 first verifies whether the equation $e(\sigma_1, P_0 + q_1P) = e(\psi_1, P_0)$ holds, where $q_1 = H_1(\text{ID}_1)$. If verified, P_2 computes $X = r_2\psi_1$ and sets the session key as $sk_1 = H_2(\text{ID}_1 \parallel \text{ID}_2 \parallel \psi_1 \parallel \psi_2 \parallel X)$.

Remark. The consistency of the computation is verified as

$$\begin{aligned}
e(\sigma_1, P_0 + q_1P) &= e\left(r_1 \frac{s}{s + q_1} P, (s + q_1)P\right), \\
&= e(r_1 s P, P), \\
&= e(r_1 P, s P), \\
&= e(\psi_1, P_0), \\
e(\sigma_2, P_0 + q_2P) &= e\left(r_2 \frac{s}{s + q_2} P, (s + q_2)P\right), \\
&= e(r_2 s P, P), \\
&= e(r_1 P, s P), \\
&= e(\psi_1, P_0).
\end{aligned} \tag{1}$$

4. Security Analysis of ID-2PAKA Protocol

There are some security vulnerabilities in the proposed ID-2PAKA protocol that cannot be ignored, which will be introduced in detail in this subsection. The security analysis of ID-2PAKA protocol in this paper is based on the theory of eCK model, which is mainly composed of

Ephemeral Key Compromise Impersonation Attack and Secure Authentication.

In the idea of eCK model, we can consider the security of the scheme from the perspective of leaking any two keys, except for leaking the long-term private key and temporary private key of a communicating party at the same time. The security analysis of ID-2PAKA protocol is given as follows.

4.1. Ephemeral Key Compromise Impersonation Attack.

After analysis, when the ephemeral keys r_1 and r_2 of both communicating parties are leaked, the adversary \mathcal{A} can recover the corresponding session key according to the leaked messages. Thus, ID-2PAKA protocol cannot resist ephemeral key compromise impersonation attack. The details are described in the following.

In the case that r_1, r_2 are known to \mathcal{A} and $\{q, G_1, G_2, P, e, H_1, H_2, P_0\}$ are public to all entities, so that \mathcal{A} can compute $\psi_1 = r_1P, \psi_2 = r_2P$ and $X = r_1r_2P$. The session key is computed as $sk_1 = H_2(\text{ID}_1 \parallel \text{ID}_2 \parallel \psi_1 \parallel \psi_2 \parallel X)$. In this way, the adversary can easily compute the vital session key without having to do any modification or insertion operations.

4.2. Secure Authentication.

In addition to the ephemeral key compromise impersonation attack, the ID-2PAKA protocol is also insecure in terms of identity authentication. The verification of either party to the other is based on the equation $e(\sigma_1, P_0 + q_1P) = e(r_1(S/(s + q_1))P, (s + q_1)P)$. However, the equation is essentially established by relying on the ephemeral key r_1 . The processes of disguising P_1 and P_2 and completing the session key agreement phase are described below.

If \mathcal{A} pretends to be P_1 , she first chooses $r'_1 \in Z_q^*$, then computes $\psi'_1 = r'_1P_0 + r'_1q_1P$ and $\sigma'_1 = r'_1P_0$, finally sends the tuple $\{\psi'_1, \sigma'_1\}$ to the responder. The responder P_2 verifies the equation $e(\sigma'_1, P_0 + q_1P) = e(\psi'_1, P_0)$. The correctness is as follows:

$$\begin{aligned}
e(\sigma'_1, P_0 + q_1P) &= e(r'_1P_0, P_0 + q_1P), \\
&= e(r'_1sP, (s + q_1)P), \\
&= e(r'_1(s + q_1)P, sP), \\
&= e(\psi'_1, P_0).
\end{aligned} \tag{2}$$

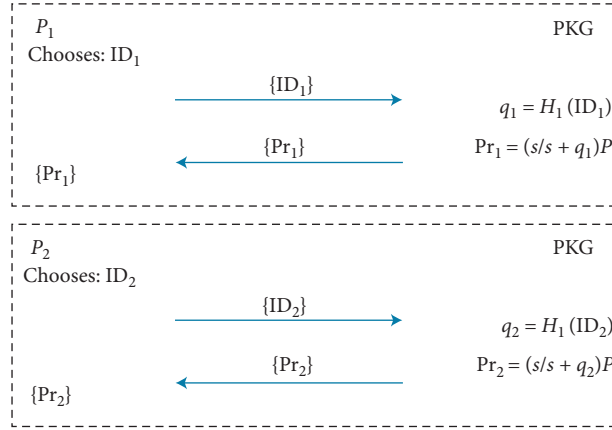


FIGURE 2: Private key generation phase.

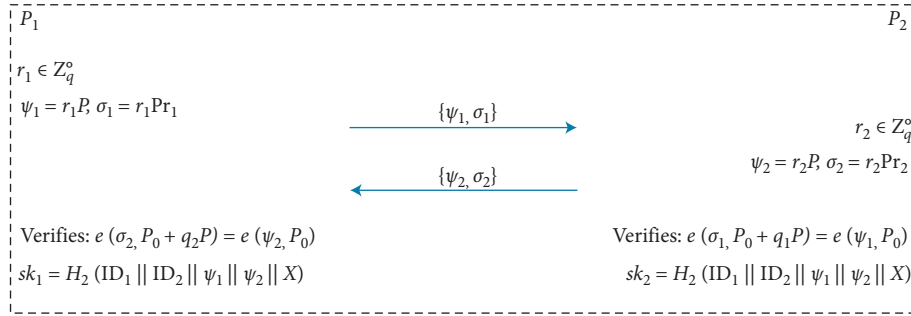


FIGURE 3: Session key agreement phase.

In the same way, \mathcal{A} can pretend to be P_2 . First, \mathcal{A} chooses $r'_2 \in Z_q^*$, then computes $\psi'_2 = r'_2 P_0 + r'_2 q_2 P$ and $\sigma'_2 = r'_2 P_0$, finally sends the tuple $\{\psi'_2, \sigma'_2\}$ to the initiator. The initiator P_1 verifies the equation $e(\sigma'_2, P_0 + q_2 P) = e(\psi'_2, P_0)$. The correctness is as follows:

$$\begin{aligned}
 e(\sigma'_2, P_0 + q_2 P) &= e(r'_2 P_0, P_0 + q_2 P), \\
 &= e(r'_2 s P, (s + q_2) P), \\
 &= e(r'_2 (s + q_2) P, s P), \\
 &= e(\psi'_2, P_0).
 \end{aligned} \tag{3}$$

5. Conclusions

Secure communication is a vital point in IIoT environment, which should be given full consideration. There are many ID-AKA protocols for IIoT environments suffer from a variety of attacks. ID-AKA protocols based on bilinear pairing have advantage in terms of security. In this paper, we analyze the ID-2PAKA protocol, which is a lightweight identity-based authenticated key agreement protocol for industrial Internet of Things proposed by Gupta et al. recently. The analysis results show that the ID-2PAKA protocol cannot obtain the secure identity authentication or resist ephemeral key compromise impersonation attack. The main reason for this situation is that there are some security flaws in the misuse of ephemeral key and long-term private key.

Data Availability

No data were used to support this study.

Conflicts of Interest

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

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (grant nos. 61872449 and 62072093).

References

- [1] D. Kiel, C. Arnold, and K.-I. Voigt, "The influence of the Industrial Internet of Things on business models of established manufacturing companies—a business level perspective," *Technovation*, vol. 68, pp. 4–19, 2017.
- [2] A. Nauman, Y. A. Qadri, M. Amjad, Y. B. Zikria, M. K. Afzal, and S. W. Kim, "Multimedia Internet of Things: a comprehensive survey," *IEEE Access*, vol. 8, pp. 8202–8250, 2020.
- [3] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of industrial Internet of Things security: requirements and fog computing opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2489–2520, 2020.

- [4] R. Ande, B. Adebisi, M. Hammoudeh, and J. Saleem, "Internet of Things: evolution and technologies from a security perspective," *Sustainable Cities and Society*, vol. 54, p. 101728, 2020.
- [5] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology-CRYPTO 1984*, pp. 47–53, Springer, Berlin, Germany, 1984.
- [6] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proceedings of the Annual International Cryptology Conference*, pp. 213–229, Santa Barbara, CA, USA, August 2001.
- [7] S. Kumari, M. Karuppiyah, A. K. Das, X. Li, F. Wu, and N. Kumar, "A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers," *The Journal of Supercomputing*, vol. 74, no. 12, pp. 6428–6453, 2018.
- [8] A. Karati, S. H. Islam, M. Karuppiyah et al., "Provably secure and lightweight certificateless signature scheme for IIoT environments," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3701–3711, 2018.
- [9] D. S. Gupta, S. H. Islam, M. S. Obaidat, P. Vijayakumar, N. Kumar, and Y. Park, "A provably secure and lightweight identity-based two-party authenticated key agreement protocol for IIoT environments," *IEEE Systems Journal*, pp. 1–10, 2020.
- [10] X. Miao, P. Fan, and D. Mu, "The study on wireless sensor networks security access scheme," in *Proceedings of the 2009 3rd International Conference on Teaching and Computational Science (WTCS 2009)*, pp. 233–241, Shenzhen, China, December 2009.
- [11] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.
- [12] X. Zhou and Y. Xiong, "An efficient and lightweight user authentication scheme for wireless sensor networks," in *Information Computing and Applications*, pp. 266–273, Springer, Berlin, Germany, 2011.
- [13] K. S. Arikumar and K. Thirumoorthy, "Improved user authentication in wireless sensor networks," in *Proceedings of the 2011 International Conference on Emerging Trends in Electrical and Computer Technology*, pp. 1–15, Nagercoil, India, March 2011.
- [14] A. Mnif, O. Cheikhrouhou, and M. B. Jemaa, "An ID-based user authentication scheme for wireless sensor networks using ECC," in *Proceedings of the 2011 International Conference on Microelectronics (ICM)*, pp. 1–9, Hammamet, Tunisia, December 2011.
- [15] J. Liu, Y. Xiao, and C. P. Chen, "Authentication and access control in the Internet of Things," in *Proceedings of the 2012 32nd International Conference on Distributed Computing Systems Workshops*, pp. 588–592, Macau, China, June 2012.
- [16] M. E. S. Saeed, Q.-Y. Liu, G. Tian, B. Gao, and F. Li, "AKA-IoTs: authenticated key agreement for Internet of Things," *Wireless Networks*, vol. 25, no. 6, pp. 3081–3101, 2019.
- [17] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Networks*, vol. 25, no. 8, pp. 4737–4750, 2019.
- [18] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Networks*, vol. 36, no. 1, pp. 152–176, 2016.
- [19] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, pp. 1133–1146, 2018.
- [20] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.
- [21] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.

Research Article

Exploring the Optimum Proactive Defense Strategy for the Power Systems from an Attack Perspective

Jinxiong Zhao , **Xun Zhang, Fuqiang Di, Sensen Guo, Xiaoyu Li, Xiao Jing, Panfei Huang, and Dejun Mu**

School of Cybersecurity, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

Correspondence should be addressed to Jinxiong Zhao; jxzhao1229@163.com

Received 30 December 2020; Revised 19 January 2021; Accepted 4 February 2021; Published 12 February 2021

Academic Editor: James Ying

Copyright © 2021 Jinxiong Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Proactive defense is one of the most promising approaches to enhance cyber-security in the power systems, while how to balance its costs and benefits has not been fully studied. This paper proposes a novel method to model cyber adversarial behaviors as attackers contending for the defenders' benefit based on the game theory. We firstly calculate the final benefit of the hackers and defenders in different states on the basis of the constructed models and then predict the possible attack behavior and evaluate the best defense strategy for the power systems. Based on a real power system subnet, we analyze 27 attack models with our method, and the result shows that the optimal strategy of the attacker is to launch a small-scale attack. Correspondingly, the optimal strategy of the defender is to conduct partial-defense.

1. Introduction

Energy is one of the most important forces to promote the development of industry in the entire society. The energy systems, the only channels for energy transmission, are responsible for the stable transmission of energy. As the primary branch of the energy systems, the power systems have been the focus to be assaulted in recent years [1]; in 2010, a power plant in Iran was attacked by the Stuxnet virus, which made the Iranian nuclear power plant lose its power generation capacity for a short time [2]; in 2014, the malicious software Black Energy invaded into USA power turbines during which USA power systems suffered a total of no less than 79 hacker attacks; in 2015, Ukrainian power systems were attacked by a malicious code, which caused a large-scale blackout; in 2016, a great many computers of the power systems, attacked in Israel by hackers, were in a suspended state; in 2019, many major hydropower stations in Venezuela were under cyber-attack, which occurred in more than half of the regions with a large-scale power outage for more than 6 days.

The fundamental reason that energy systems such as the power systems can be frequently attacked successfully is that the protection strategy of each system is passive and static and it does not have an autoimmune function [3]. For such prominent problems, a lot of researches on proactive defense have been conducted in the industry areas and the related works include moving target defense, mimic defense, and end-to-end hopping [3, 4]. The abovementioned proactive defense technologies have made considerable progress in theory, but the disadvantage is that it needs to take a huge cost to build a system with the above defense attributes, which is often unbearable. For the sake of solving this problem, many scholars have applied game theory to network security defense, but so far, there have been fewer reports on game theory that can be employed to solve security problems in the real power production systems.

In this paper, we firstly introduce some related typical works in the field of active and proactive defense. Secondly, a single and dual game model between the hackers and defenders of the power systems, based on the introduction of game theory, is constructed. Then, the established attack and

defense model is verified by the production environment attack data from a real power systems subnet. Finally, the model that matches the real production environment is used to predict the hostile attack strategy in the next month and the key defense points of the power systems under the current situation are given.

The significant contribution of this article is to provide a qualitative method for evaluating external attacks for the power systems, which is as follows: the idea with the revenue of the power systems being robbed is proposed for the first time; taking a subnet in a real power systems as an example, three single and dual attack and defense models are discussed in detail, respectively; the theoretical model in line with the real production environment was established and verified by the attack data in actual production; the best revenues of the attack and defense sides are calculated separately, and the best defense modes for the power systems when facing different attack scales are given. Overall, the main contributions of our works are as follows:

- (1) Compared with the mimic defense and moving target defense methods in the current industry, we propose a relatively low-cost proactive defense method based on game theory.
- (2) By calculating the best benefits of the attacker and defender in the game, we can predict the most likely attack behavior and provide more targeted defense strategy for the power systems.
- (3) We have evaluated the benefits of the three combined attack strategies that are closer to the actual attack situations for both the attacker and defender and verified it with actual attack data.

2. Related Work

Currently, there are many related reports on active defense on active network defense, mainly including moving target defense (MTD), mimic security defense (MSD), end information hopping (EIH), game theory defense technologies (GTD), and information theory approach (ITA) [5, 6]. Deformation networks, adaptive computer networks, self-cleaning networks, and open-flow random host conversion technologies are widely reported in MTD, and the essence of them is to make it difficult for an attacker to accurately grasp the information of the target systems by proactively changing its relevant configuration within a certain time interval. The advantage of this theory is that it can improve the security of the systems attacked by forcing the attacker to continuously increase attack cost, but the pain points inside are needing a huge available configuration space to support the operation of this defense technology. The concept of MSD is fairly similar to that of MTD. Both technologies enable defenders to realize rapid migration in a diverse environment, thereby augmenting the difficulty degree of the hackers. Compared with MTD, MSD has more heterogeneous redundant architectures with the disadvantages that require huge investment cost, while MTD provides limited heterogeneous redundant architectures, which can be regarded as a special case of MTD. EIH, composed of

early warning, collaborative control, information management, and task switching function modules, mainly protects the two communication parties by changing crucial information such as the protocol, address, and port between the two communication ends. GTD is a network defense technology based on game theory, which belongs to the theory of beforehand decision analysis and has been used in the field of network security for many years. Although the abovementioned methods have a certain effect on the actual environment, the common problem is that these designs are expensive.

Game theory is an ideal solution to the problem of high cost in proactive defense methods. Many scholars have done a lot of research on it. Radha et al. proposed a game theory optimization routing framework for wireless networks, which provided a solution for the realization of low-energy routing [7]. Zhu and Basar explored the game mechanism of the optimal cross-layer flexible control system to enhance the robustness and security of the cyber-physical system [8]. Zhao et al. studied the game theory model based on the distribution market and solved the problem of the coordinated operation of multiple microgrids [9]. Rass and Zhu analyzed the defense-in-depth strategy of advanced persistent threats and proposed a method to deal with the threats [10]. Chen et al. used dynamic game theory to design a network protection and recovery system for infrastructure to ensure reliable service provision [11]. Miao et al. established a zero-sum mixed state random game model to solve different types of attacks on cyber-physical systems [12].

The abovementioned method based on game theory does solve many relevant network security problems. However, it neither calculates the value of offense and defense benefits nor does it use real-world attack data of power systems to verify its theoretical model. Therefore, the problem solved in this paper is how to use game theory to model real industrial control systems and how to qualitatively give the best protection strategy for power systems.

3. Preliminaries

3.1. Bayesian Game Theory. Bayesian game is also called incomplete information game, which means that at least one player among multiple players is not completely clear about the revenues or revenue functions of the remaining players. In this article, incomplete information means that the defender on the power systems does not know the method and purpose of the hackers. Similarly, the hackers are not fully aware of the power systems. Thus, we need to introduce the Bayesian game model to analyze the possible behaviors and revenues of both parties. Bayesian game is not repeated here because the existing literature is very detailed about it [13, 14].

3.2. Attack and Defense Model Construction of the Power Systems. Many factors that affect the safe operation of the power systems and the main targets that most likely to be attacked are focused on, including the host, network, and management. Here, we primarily build the single and dual

models of the power systems among which the single models include host, network, and management model and the dual models include host and network, host and management, and network and management model.

Under normal circumstances, a considerable revenue, recorded as the total revenue s , can be obtained by defenders from their assets inside the power systems. In consideration of the assets characteristics with wide coverage and multifaceted feature, there will always exist potential security vulnerabilities and this part, denoted as l , is defined as the inherent loss. The hackers frequently utilize various vulnerabilities in an attempt to reduce the defender's revenue. At this time, the hackers' benefit and cost are proportional to the attack size. When the defense side detects that the malicious forces from outside are attempting to damage the power system, it will consume a certain cost and adopt corresponding defense strategies to intercept. Once the power system is severely damaged, it needs at a significant cost to repair it. In this paper, the deliberate attack scale from the outside world is divided into three categories: large-scale, small-scale, and no-attack. The defense strategy in the power systems is divided into complete-defense, partial-defense, and no-defense. Judging from the data about the centralized attacks on the power systems organized by government departments every year, the main targets of the attack are the host, network, and management. The corresponding risk levels are 2, 3 and 2, respectively. According to the respective damage levels, specific values are assigned to the parameters in the models, as shown in Figure 1, and the specific meaning is listed in Table 1. Besides, the original benefits in the adversarial sides are increased by 10 to be convenient for processing data.

3.3. Single Attack-Defense Models

3.3.1. Host Attack-Defense Model. Considering that each attack/defense is a frequently organized and complicated process, all kinds of costs and revenues here are relative values and greater than zero. The expenditure cost from the hackers on large-scale, small-scale, and no-attack can be expressed as a matrix s_a , and the expenditure cost of the power systems on complete-, partial-, and no-defense can be expressed as a matrix s_d :

$$\begin{aligned} s_a &= \begin{bmatrix} -t \\ -(t - \Delta t) \\ 0 \end{bmatrix}, \\ s_d &= \begin{bmatrix} -p \\ -(p - \Delta p) \\ 0 \end{bmatrix}. \end{aligned} \quad (1)$$

There are three defense modes for the power systems to choose each time, and every defense mode may face any attack strategies from outside. In the following, the overall revenue matrix s_e obtained by the hackers is given in detail, among which the complete-, partial-, and no-defense in the power systems are taken as row vectors and the large-scale,

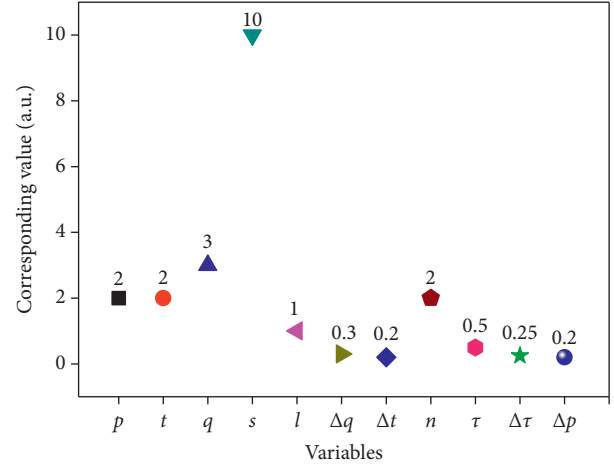


FIGURE 1: The value of different variables.

small-scale, and no-attack from the attackers are used as column vectors, respectively:

$$s_e = \begin{bmatrix} q & q + \Delta q & q + n^* \Delta q \\ \Delta q & q & q + \Delta q \\ 0 & 0 & 0 \end{bmatrix}. \quad (2)$$

At this time, according to the abovementioned discussion of various parameters in both sides, the total revenue of the hackers under different attack strategies corresponding to the different defense modes can be calculated, as shown in Figure 2. In the complete-defense mode, Figure 2(a) shows that the gained revenue by the hackers reaches maximum when they adopt large-scale strategy, with the result that the revenue in the power systems is forced to drop to the bottom, but the cost paid by the hackers is also huge. In partial-defense mode, Figure 2(b) reveals that the two sides reach the Nash equilibrium when the hackers use small-scale strategy. In no-defense mode, Figure 2(c) indicates that both sides also reach the Nash equilibrium when the hackers employ a small-scale strategy. It can be concluded that the probability of adopting small- and large-scale strategies for the hackers, respectively, is 2/3 and 1/3 corresponding to the three defense modes that the defender can choose. Therefore, the small-scale attack should be paid close attention to in-host attack and defense model.

3.3.2. Network Attack and Defense Model. The construction process of the network attack and defense model is the same as that of the host, and the cost of both sides is exactly the same as the matrix (1). Considering the openness and accessibility of the network, the extent of injury from an attacker via the network is slightly higher than that of the host, so its specific revenue is shown in the following matrix:

$$s'_e = \begin{bmatrix} 2q & 2q + \Delta q & 2q + n^* \Delta q \\ n^* \Delta q & 2q & 2q + \Delta q \\ 0 & 0 & 0 \end{bmatrix}. \quad (3)$$

TABLE 1: Summary of symbols and meaning.

Symbols	Meaning
t	The cost of a large-scale attack
$t - \Delta t$	The cost of a small-scale attack
$p - \Delta p$	The cost of a partial-defense
q	The benefit made by the attacker with launching a large-scale attack when the defender is in a state of complete-defense
$q + \Delta q$	The benefit made by the attacker with launching a large-scale attack when the defender is in a state of partial-defense
$q + n^* \Delta q$	The benefit made by the attacker with launching a large-scale attack when the defender is in a state of no-defense
q	The benefit made by the attacker with launching a small-scale attack when the defender is in a state of partial-defense
$q + \Delta q$	The benefit made by the attacker with launching a small-scale attack when the defender is in a state of no-defense
l	Inherent loss
s	The total benefit obtained by the defender using its own assets

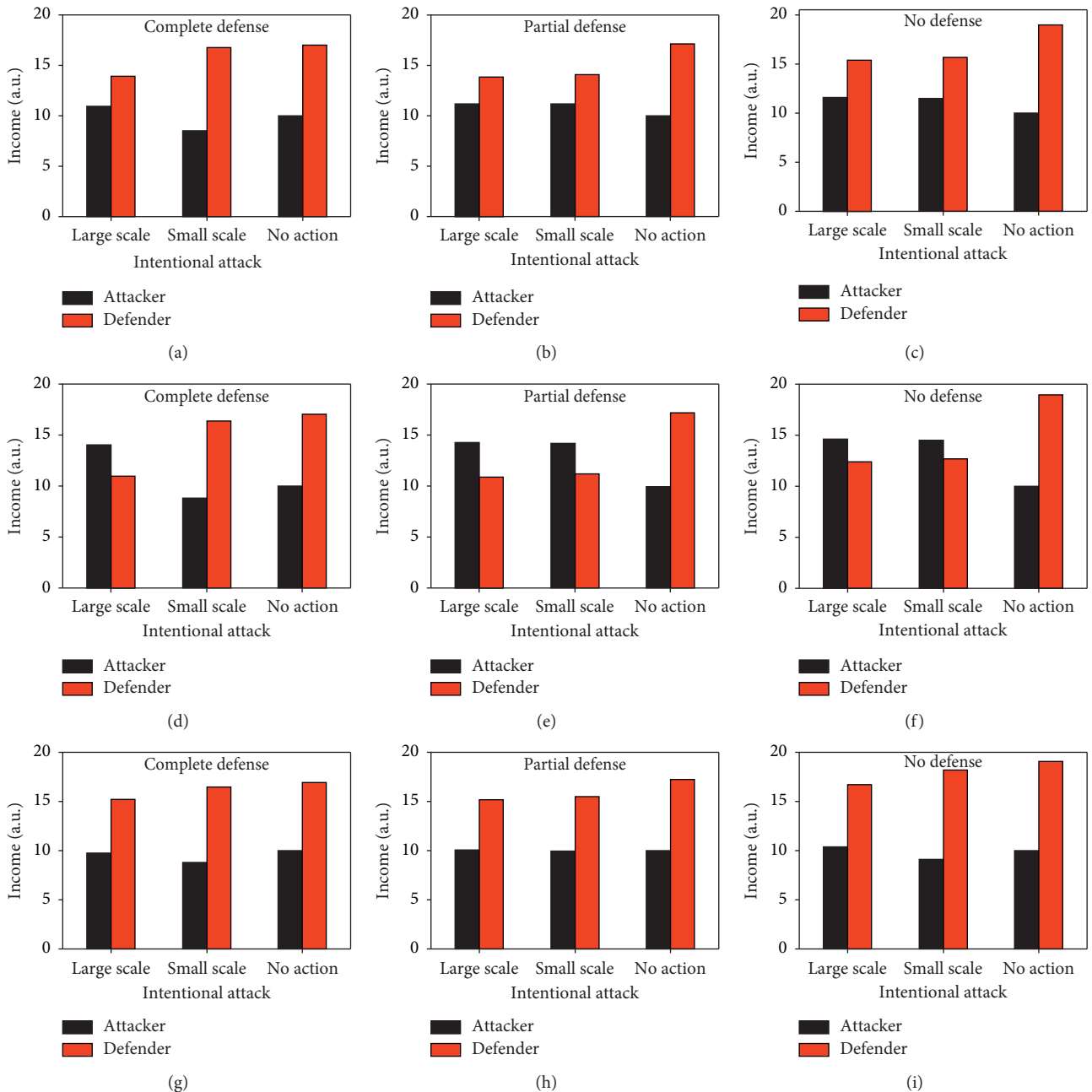


FIGURE 2: Under (a), (d), and (g) complete-defense mode, (b), (e), and (h) partial-defense mode, and (c), (f) and (i) no-defense mode, both sides benefit when the hackers launch different attack strategies.

Next, according to the cost of both sides and the revenue obtained by the hackers, we have calculated their total revenues in different states, as shown in Figure 2. Although the defender chooses complete-defense mode in Figure 2(d), the hackers still make very considerable gains by using large-scale strategy, owing to the network openness and its own various flaws such as protocol loopholes. Furthermore, the hacker revenue in Figure 2(d) is higher than that of the power systems and both sides have reached the Nash equilibrium, but considering the cost, this attack strategy will only be employed at a particular moment; in partial-defense mode, both sides reach the Nash equilibrium when the hackers adopt the small-scale strategy, and the revenue of the hackers is greater than that of the power systems, as shown in Figure 2(e); in no-defense mode, Figure 2(f) displays that the two sides also reach the Nash equilibrium when the hackers use small-scale strategy. At this moment, the hackers reach the optimal value in terms of costs and revenues.

In these three defense modes, the probability of small-scale strategy (2/3) is greater than the probability of large-scale strategy (1/3). It is worth noting that the conditions for the emergence of large-scale strategy are that the defender must select complete-defense mode and the hackers are willing to be at all costs.

3.3.3. Management Attack and Defense Model. The construction of the management attack and defense model is also similar to the network and the cost of both sides is identical with matrix (1). Different from the above-mentioned network attack and defense model, the degree of attack severity through management defects is lower than that of the host, and its specific revenue is presented in the following matrix:

$$s_e'' = \begin{bmatrix} q^{1/2} & q^{1/2} + \Delta q & q^{1/2} + n^* \Delta q \\ \Delta q^{1/2} & q^{1/2} & q^{1/2} + \Delta q \\ 0 & 0 & 0 \end{bmatrix}. \quad (4)$$

Then, based on the known costs and the gains obtained, the total revenues of the attack and defense parties in different states can be calculated, as shown in Figure 2. The data in Figure 2(g) indicate that the cost paid by the hackers is far higher than their revenues when large- and small-scale strategies are adopted. It can be inferred from the above that the best attack strategy of the hackers is to keep a static state and then both sides reach the Nash equilibrium with the defender being in complete-defense mode. In partial-defense mode, the hackers would try to remain stationary since Figure 2(h) indicates that the revenue of the hackers, although higher than in other cases, is still minimal with large-scale attacks. As a result, the power systems achieve the maximum revenue with a state of Nash equilibrium. Obviously, the situation in Figure 2(i) is similar to Figure 2(h).

In the management attack and defense model, it is fairly difficult for the hackers to find out a breakthrough point to invade the power systems because the various management measures on it are relatively complete. Therefore, the smartest choice for an attacker is to use a static observation strategy.

3.4. Dual Attack and Model

3.4.1. Host and Network Attack and Defense Model. In the general host and network policy configuration process, the security policy configuration of the host or network is usually completed first, and then the rest of the policy settings are completed in turn. The default security policy setting order in this part is host network. It is known that many security policies are often universal. Here, the security policies that have been set in the host and can be used in the network are recorded as cost savings. The following lists the cost savings matrix s_s of the host in complete-, partial-, and no-defense:

$$s_s = \begin{bmatrix} \tau \\ \tau - \Delta\tau \\ 0 \end{bmatrix}. \quad (5)$$

According to the cost saved by matrix (5), it can be calculated that the defense cost paid by the power systems in the case of complete-, partial-, and no-defense of the host is displayed in the matrix s_1 , s_2 , and s_3 :

$$\begin{aligned} s_1 &= \begin{bmatrix} -p + \tau \\ -(p - \Delta p) + \tau \\ \tau \end{bmatrix}, \\ s_2 &= \begin{bmatrix} -p + \tau - \Delta\tau \\ -(p - \Delta p) + \tau - \Delta\tau \\ \tau - \Delta\tau \end{bmatrix}, \\ s_3 &= \begin{bmatrix} -p \\ -(p - \Delta p) \\ 0 \end{bmatrix}, \end{aligned} \quad (6)$$

that the overall revenue required by various attack methods minus its cost is the ultimate revenue of the attack side. The final revenue on the power systems is that its total revenue subtracts the inherent loss, defense cost, and the plundered revenue. Thus, under the complete-, partial-, and no-defense of the host, the total revenue of the attack side can be gotten via the calculation process provided above, which is the average of the sum of the revenues of the corresponding host and network single model, as shown in the following matrix:

$$s_{hc}^I = \begin{bmatrix} \frac{3q}{2} & \frac{(3q + 2\Delta q)}{2} & \frac{(3q + 2n\Delta q)}{2} \\ \frac{(n + 1)\Delta q}{2} & \frac{3q}{2} & \frac{(3q + 3\Delta q)}{2} \\ 0 & 0 & 0 \end{bmatrix}. \quad (7)$$

Facing the three defense modes of the power systems, Figure 3 analyzes the final revenue that an attacker can achieve by using three different attack strategies. In the first place, we discuss the revenues of the attack and defense sides when the host is in complete-defense mode: with the network being complete-defense, the most likely outcome is

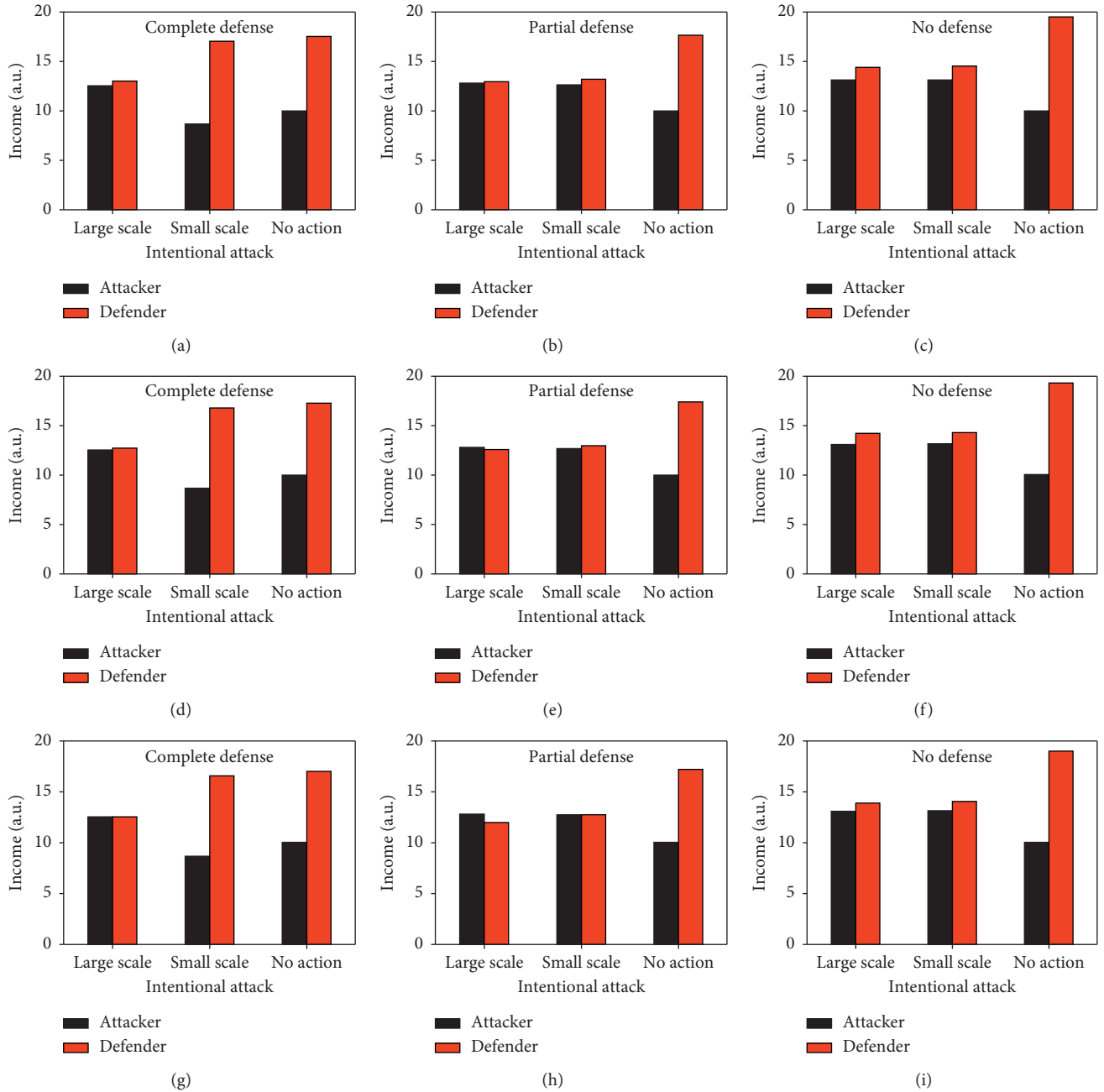


FIGURE 3: Under (a) Complete-defense of host and network, (b) complete-defense of host and partial-defense of network, (c) complete-defense of host and no-defense of network, (d) partial-defense of host and complete-defense of network, (e) partial-defense of host and network, (f) partial-defense of host and no-defense of network, (g) no-defense of host and complete-defense of network, (h) no-defense of host and partial-defense of network, and (i) no-defense of host and network, both sides benefit when the hackers launch different attack strategies.

that the attack side would adopt large-scale strategy to maximize its revenue, which is very close to the value of the power systems in Figure 3(a). In Figure 3(b), the revenue obtained by the hackers using small-scale strategy is almost equal to that obtained with large-scale strategy, and the two sides reach the Nash equilibrium; therefore, the possibility of suffering small-scale attack is the greatest in the case of network partial-defense; Figure 3(c) shows that the attack revenue is higher than that of the other two strategies when

small-scale strategy is employed by an attacker to destroy the power systems, which enable the two sides to reach the Nash equilibrium, it is why most of the outside invaders launch small-scale strategy when the network is in no-defense.

Afterwards, we discuss the revenue made by the attack and defense sides when the host is in partial-defense mode: as shown in Figure 3(d), small-scale strategy is the most unfavorable tactic for the hackers and its revenue is negative, while large-scale strategy makes the hackers' revenue

basically equal to that of the power systems and the both sides reach the Nash equilibrium. The data in Figure 3(d) indicates that the power systems are more likely to be subjected to large-scale attacks when the network is set as complete-defense mode; obviously, small-scale strategy makes the revenue of the attack and defense parties achieve the Nash equilibrium, as shown in Figure 3(e), which demonstrates that the most probable attack strategy is to launch a small-scale attack when the network is in partial-defense mode; the situation in Figure 3(f) is extremely similar to Figure 3(e).

Finally, we discuss the revenues of the attack and defense sides when the host is in no-defense mode: it can be seen from Figure 3(g) that small-scale strategy not only fails to break the defense of the power systems but also makes the attack side pay a huge price. On the contrary, large-scale strategy can maintain the revenues of both parties at a balanced point. Consequently, it must be alert to large-scale attack from the enemy when the network adopts complete-defense; the hackers utilize small-scale strategy to receive the same revenue as the power systems, and the two sides have reached the state of Nash equilibrium, as shown in Figure 3(h), when the network adopts partial-defense. At this time, we must pay more attention to the loss caused by small-scale strategy adopted by the attack side; when the network adopts no-defense, Figure 3(i) displays that small-scale strategy launched by the hackers has the largest gain and is also the most desirable strategy compared with the cost paid by large-scale strategy. Thus, small-scale strategy remains the focus of attention.

Based on the above discussion, this part lists nine possible combinations in host and network attack and defense model among which the probability that the hackers may adopt large- and small-scale strategies is 3/9 and 6/9, respectively. Therefore, it is critical to be aware of small-scale strategy implemented by the attack side for most of the time. In special circumstances, the possibility of an attacker launching a large-scale strategy is not ruled out.

3.4.2. Host and Management Attack and Defense Model. The possibility of an attacker carrying out a malicious attack on the host and management is also bound to exist. The main discussion here is to consider the establishment of host and management attack and defense model when the host security policy has been completed. In view of the low cross degree of the security strategy between the host and management, the cost saving matrix s'_s of the host during complete, partial-, and no-defense is as follows:

$$s'_s = \begin{bmatrix} \tau^{1/2} \\ \tau^{1/2} - \Delta\tau^{1/2} \\ 0 \end{bmatrix}. \quad (8)$$

Similarly, according to the cost saved in matrix (8), the defense costs paid by the side of the power systems in the above three cases (complete-, partial-, and no-defense) can be calculated, respectively, as the matrix s'_1 , s'_2 , and s'_3 as follows:

$$\begin{aligned} s'_1 &= \begin{bmatrix} -p + \tau^{1/2} \\ -(p - \Delta p) + \tau^{1/2} \\ \tau^{1/2} \end{bmatrix}, \\ s'_2 &= \begin{bmatrix} -p + \tau^{1/2} - \Delta\tau^{1/2} \\ -(p - \Delta p) + \tau^{1/2} - \Delta\tau^{1/2} \\ \tau^{1/2} - \Delta\tau^{1/2} \end{bmatrix}, \\ s'_3 &= \begin{bmatrix} -p \\ -(p - \Delta p) \\ 0 \end{bmatrix}. \end{aligned} \quad (9)$$

The calculation process of the total revenue of the hackers in this type of model is similar to matrix (7). The total revenue here is the average of the sum of the corresponding host and management single model, as shown in the following matrix:

$$s'_{hm} = \begin{bmatrix} \frac{q + q^{1/2}}{2} & \frac{q + q^{1/2} + \Delta q + \Delta q^{1/2}}{2} & \frac{(q + 2n\Delta q + q^{1/2})}{2} \\ \frac{(\Delta q + \Delta q^{1/2})}{2} & \frac{q + q^{1/2}}{2} & \frac{(q + 2\Delta q + q^{1/2})}{2} \\ 0 & 0 & 0 \end{bmatrix}. \quad (10)$$

Under the three defense modes on the power systems, Figure 4 analyzes in detail the final revenues made by the attack and defense sides when the hackers adopt three different attack strategies. For the host in complete-defense mode, the final revenue of the both sides is revealed in Figures 4(a) and 4(c), respectively: when the management is in complete-defense, the average revenue of the three different attack strategies of the hackers is 9.66, which is significantly

lower than the situation where the management is in partial- and no-defense, as described in Figure 4(a). At this time, the hacker's choice of large-scale strategy can effectively prevent the power systems from maximizing its revenue, but the own revenue performance of the attack side is fairly poor so that there are two possible options for the hackers: large-scale attack or no-attack; when the management is in partial-defense, small-scale strategy used by the attack side makes the

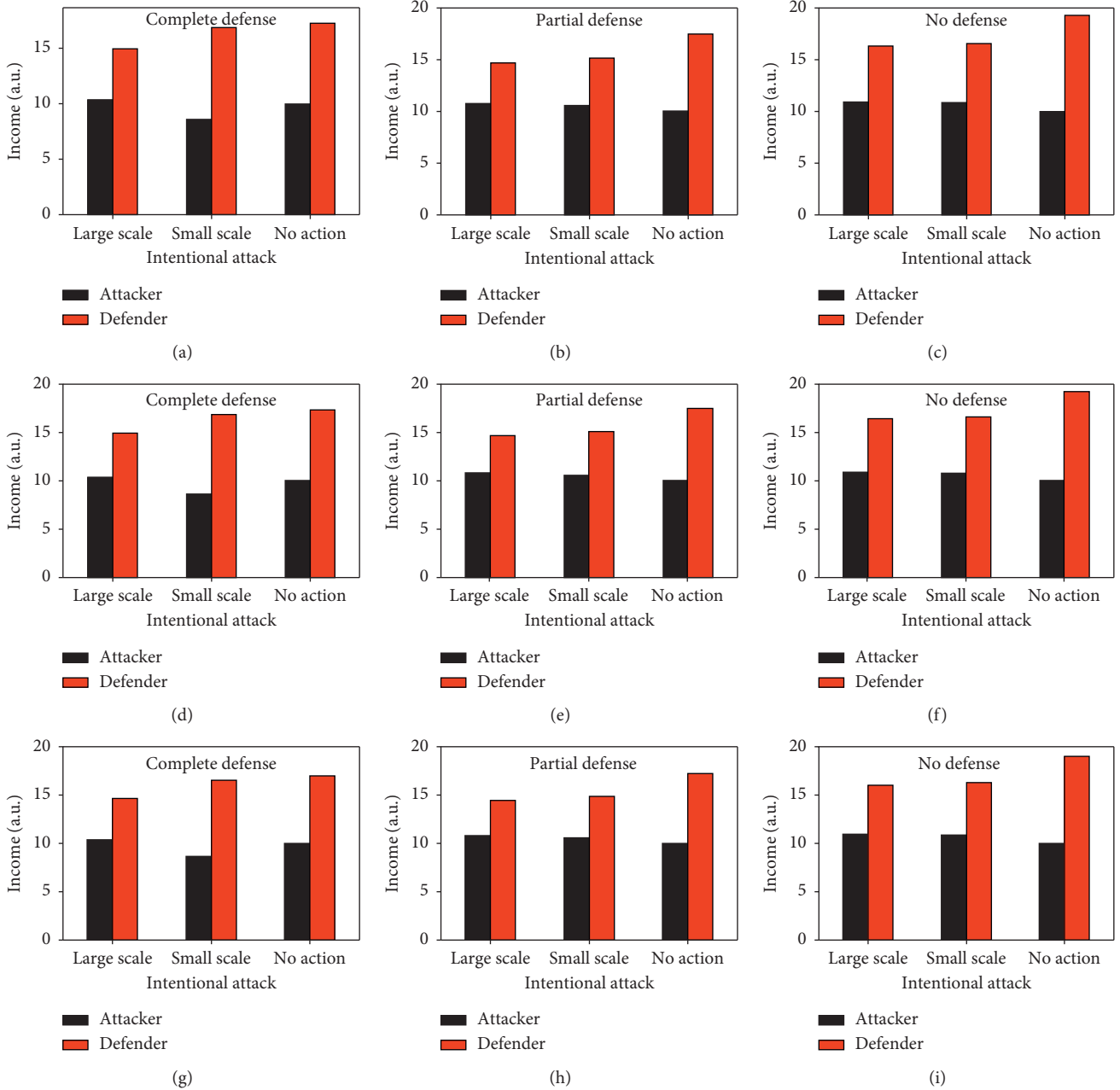


FIGURE 4: Under (a) complete-defense of host and management, (b) complete-defense of host and partial-defense of management, (c) complete-defense of host and no-defense of management, (d) partial-defense of host and complete-defense of management, (e) partial-defense of host and management, (f) partial-defense of host and no-defense of management, (g) no-defense of host and complete-defense of management, (h) no-defense of host and partial-defense of management, and (i) no-defense of host and management, both sides benefit when the hackers launch different attack strategies.

most significant contribution to its overall revenue, and the two sides have reached the Nash equilibrium. In this defense mode, the hackers are most likely to employ a small-scale strategy, as shown in Figure 4(b), as shown in Figure 4(c), the revenue of small-scale strategy is not only close to that of large-scale strategy but also the paid cost from small-scale strategy is much smaller than that of the large-scale strategy. From the perspective of maximizing revenue, the attack side will still choose a small-scale strategy when the management side is in no-defense.

For the host in partial-defense mode, the final revenues of both sides are displayed in Figures 4(d) and 4(f), respectively: in Figure 4(d), the revenue from utilizing small-scale strategy is about half that of the power systems, and it is the least of the three attack strategies. In view of that, the hackers have two options along with the management being in complete-defense: one is to be forced to adopt large-scale strategy when necessary to suppress the revenue of the power systems to reach maximum, and the other is to remain stationary to avoid its own loss; in Figure 4(e), the

defender achieves the biggest revenue when the hackers employ small-scale strategy, while the hackers get the maximum revenue with large-scale strategy. However, the maximum revenue obtained by the hackers through large-scale strategy is very close to the value obtained by small-scale strategy. To minimize cost and maximize revenue, the hackers have the highest probability of choosing small-scale strategy when the management side is in partial-defense; in Figure 4(f), the attack and defense parties arrive the Nash equilibrium when the hackers launch a small-scale strategy. Taking into account the respective costs and game issues, at this time, the revenues of both parties also reach each maximum value.

For the host in no-defense mode, the final revenues of the both sides are described in Figures 4(g) and 4(i), respectively: from Figure 4(g), it can be inferred that the revenue inside the power systems is obviously lower than that in Figures 4(a) and 4(d) in order when the management side is set as complete-defense mode; similarly, the revenue inside the power systems in Figure 4(h) is lower than Figures 4(b) and 4(e); it is also suitable for the revenue inside the power systems of Figure 4(i), lower than Figures 4(c) and 4(f). That manifests that the revenue of the power systems is more easily plundered by the hackers with the host being no-defense; the data in Figure 4(g) point that only when the attack side adopts large-scale strategy can the revenue of the power systems reach the maximization. When the management side is set as complete-defense, the hackers also have two options: launching large-scale attack when necessary to reach the Nash equilibrium or continuing to remain silent to avoid any losses; when the management side is in partial- and no-defense, as shown in Figures 4(h) and 4(i), both sides will reach the Nash equilibrium with small-scale strategy adopted by the hackers. Therefore, in the above three defense modes, the hackers are more likely to launch a small-scale strategy.

In summary, this part discusses nine possible combinations in host and management attack and defense model among which the occurrence probability of large-scale, small-scale, and no-attack strategy is 3/12, 6/12, and 3/12, respectively. So, it is critical to be aware of small-scale strategy implemented by the attack side most of the time. Under special conditions, the possibility of the attackers launching a large-scale and no-attack strategy is not excluded.

3.4.3. Network and Management Attack and Defense Model.

The third possible combination is network and management attack. Whenever a huge amount of cost is invested and a certain level of authorizations is still not available through the network path, the hackers will choose to use social engineering to seek management loopholes to breakthrough. When a certain authorization is obtained by means of management defects or a specific Trojan horse is implanted in a specific location, the hackers will successfully conduct the attack via the network path. The setting order of security policies here is the network management. Given that management vulnerabilities can often provide vital support for network attacks, the cost savings when the network is in complete-, partial-, and no-defense are denoted as $\tau/2$, $(\tau - \Delta\tau)/2$, and 0, respectively. In the light of the cost savings, the paid defense costs in the above three cases can be calculated, as exhibited, respectively, in matrix s'_1 , s'_2 , and s'_3 :

$$\begin{aligned}
 s''_1 &= \begin{bmatrix} -p + \frac{\tau}{2} \\ -(p - \Delta p) + \frac{\tau}{2} \\ \tau^{1/2} \end{bmatrix}, \\
 s''_2 &= \begin{bmatrix} -p + \frac{(\tau - \Delta\tau)}{2} \\ -(p - \Delta p) + \tau^{1/2} - \Delta\tau^{1/2} \\ \tau^{1/2} - \Delta\tau^{1/2} \end{bmatrix}, \\
 s''_3 &= \begin{bmatrix} -p \\ -(p - \Delta p) \\ 0 \end{bmatrix}.
 \end{aligned} \tag{11}$$

The calculation process of the total revenue of the hackers in this type of model is similar to matrix (9); the total revenue here is equal to the average of the sum of the corresponding single network and management model, as displayed in the following matrix:

$$s_{cm}^I = \begin{bmatrix} \frac{2q + q^{1/2}}{2} & \frac{2q + q^{1/2} + 2\Delta q}{2} & \frac{(2q + (n+1)\Delta q + q^{1/2})}{2} \\ \frac{(n\Delta q + \Delta q^{1/2})}{2} & \frac{2q + q^{1/2}}{2} & \frac{(2q + 2\Delta q + q^{1/2})}{2} \\ 0 & 0 & 0 \end{bmatrix}. \tag{12}$$

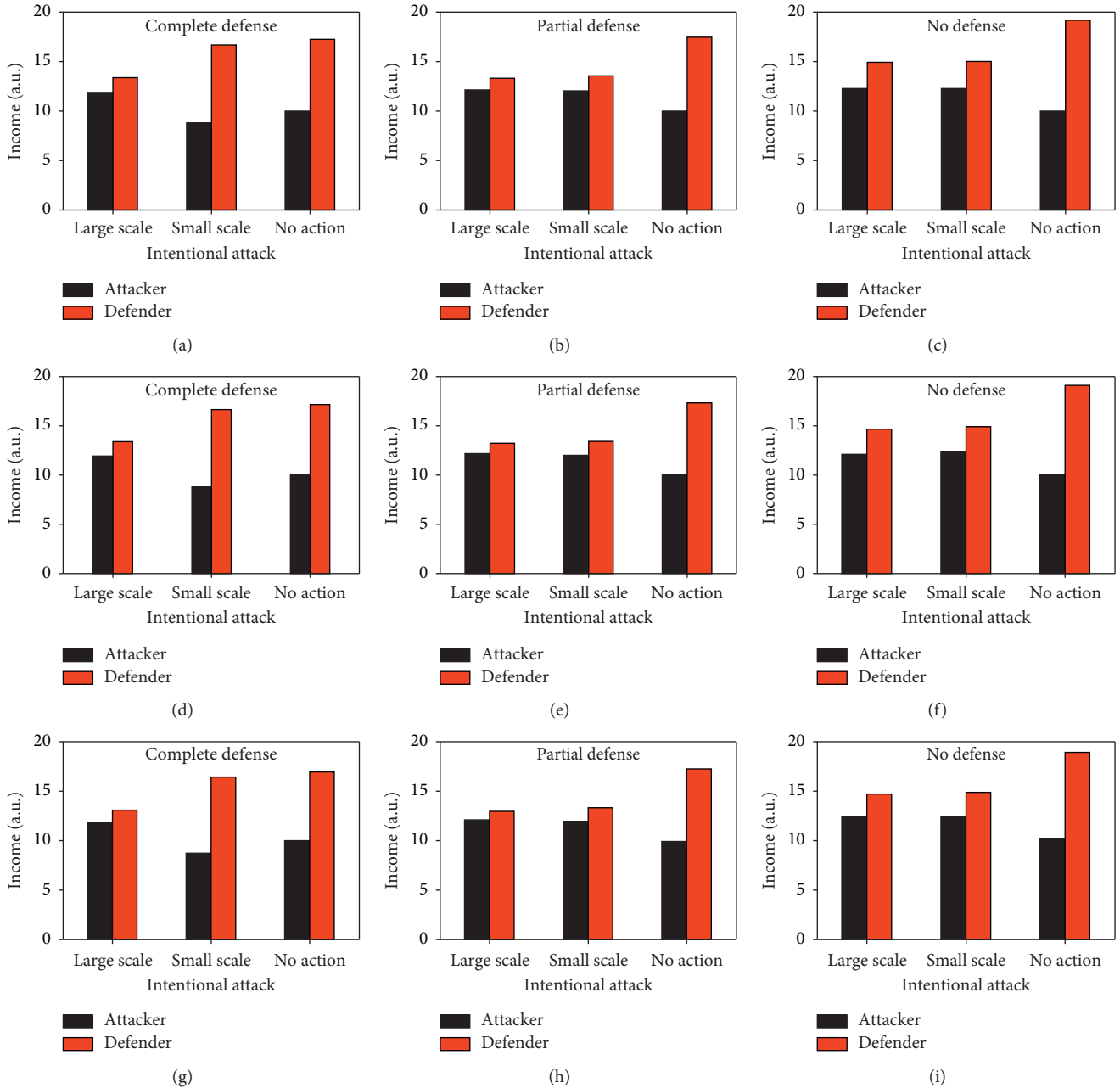


FIGURE 5: Under (a) complete-defense of network and management, (b) complete-defense of network and partial-defense of management, (c) complete-defense of network and no-defense of management, (d) partial-defense of network and complete-defense of management, (e) partial-defense of network and management, (f) partial-defense of network and no-defense of management, (g) no-defense of network and complete-defense of management, (h) no-defense of network and partial-defense of management, and (i) no-defense of network and management, both sides benefit when the hackers launch different attack strategies.

Next, we calculate the final revenues of the both sides that the revenue of the hackers acquired by various attack methods minus the attack cost is its ultimate value. That the total revenue on the power systems minus the inherent loss, defense cost, and plundered revenue is its final revenue. Under the three defense modes on the power systems, Figure 5 analyzes in detail the final revenues made by both sides when the hackers utilize three different attack strategies. For the network in complete-defense mode, the final revenues of the two sides are presented in

Figures 5(a) and 5(c), respectively: when the management side is set as complete-defense mode, Figure 5(a) indicates that the hackers can only obtain the maximum revenue by adopting large-scale strategy due to the rigorous defense on the power systems. At the same time, that strategy can effectively stop the power systems reaching its top value with the two sides being the Nash equilibrium, which demonstrates that large-scale strategy needs the most attention; when the management side is set as partial-defense mode, the two sides reach the Nash equilibrium

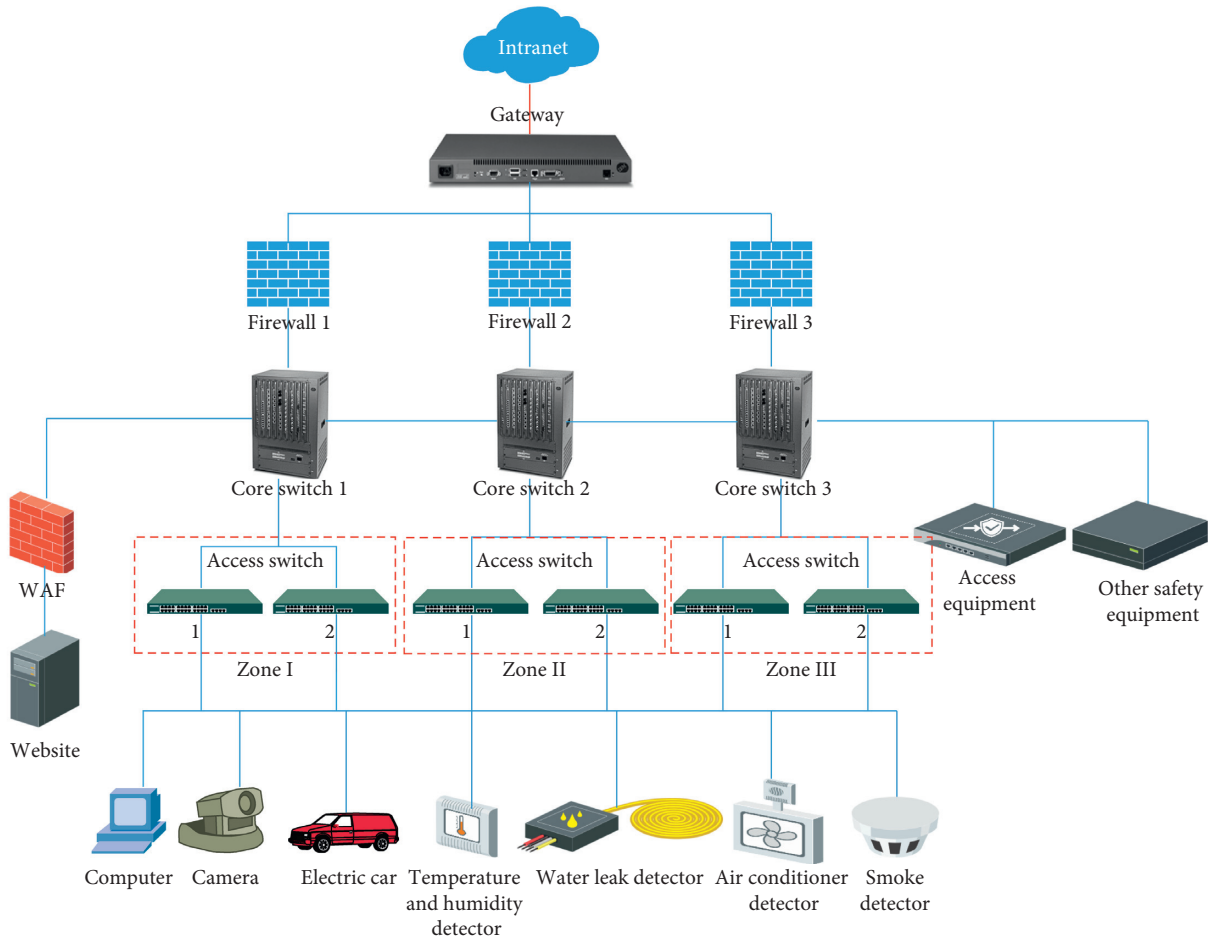


FIGURE 6: A certain production subnet in the power systems.

with the hackers using small-scale strategy, which expresses that the probability of small-scale attacks is the highest; The data in Figure 5(c) show that not only do the revenues of the two parties get the maximum value but both sides also reach a state of Nash equilibrium when the hackers launch small-scale attacks, which represents that small-scale strategy should be highly attached importance. For the network in partial-defense mode, the final revenues of the both sides are shown in Figures 5(d) and 5(f), respectively: when the management side is set as complete-defense, in order to suppress the emergence of the maximum gain on the power systems, the only option for the hackers is to employ large-scale strategy, as illustrated in Figure 5(d), so it is necessary to hinder large-scale attacks; when the management side is set as partial-defense, the attack side is more likely to seek small-scale attacks because under this situation, as presented in Figure 5(e), its output cost is fairly low and the revenue is only 0.1% lower than large-scale attack; when the management side is set as no-defense, Figure 5(f) indicates that the revenues of both sides have reached respective peaks and the state of Nash equilibrium with small-scale attacks by the hackers. For the network in no-defense mode, the final revenues of both sides are shown in Figures 5(g) and 5(i), respectively; when the management

side is set as complete-defense, the situation in Figure 5(g) is the same as that in Figure 5(d). At this time, the probability of the hackers being forced to adopt large-scale attack strategy is greater; the data trends in Figures 5(h) and 5(i) are respectively similar to those in Figures 5(e) and 5(f), and both sides have acquired each optimal value and reached the Nash equilibrium with small-scale strategy.

Based on the above discussion, here we list nine possible combinations in network and management attack and defense model among which the probability of the large- and small-scale strategy is $3/9$ and $6/9$, implying that the probability that the hackers will implement small-scale strategy is far greater than the probability of large-scale and no-attack strategy most of the time.

4. Experiment Analysis

First of all, the experimental part of this article is to verify the dual attack and defense model established previously to ensure that the constructed model can be applied to the power production systems. Then, we predict the best defense strategy on the power systems in the next month according to the construction model.

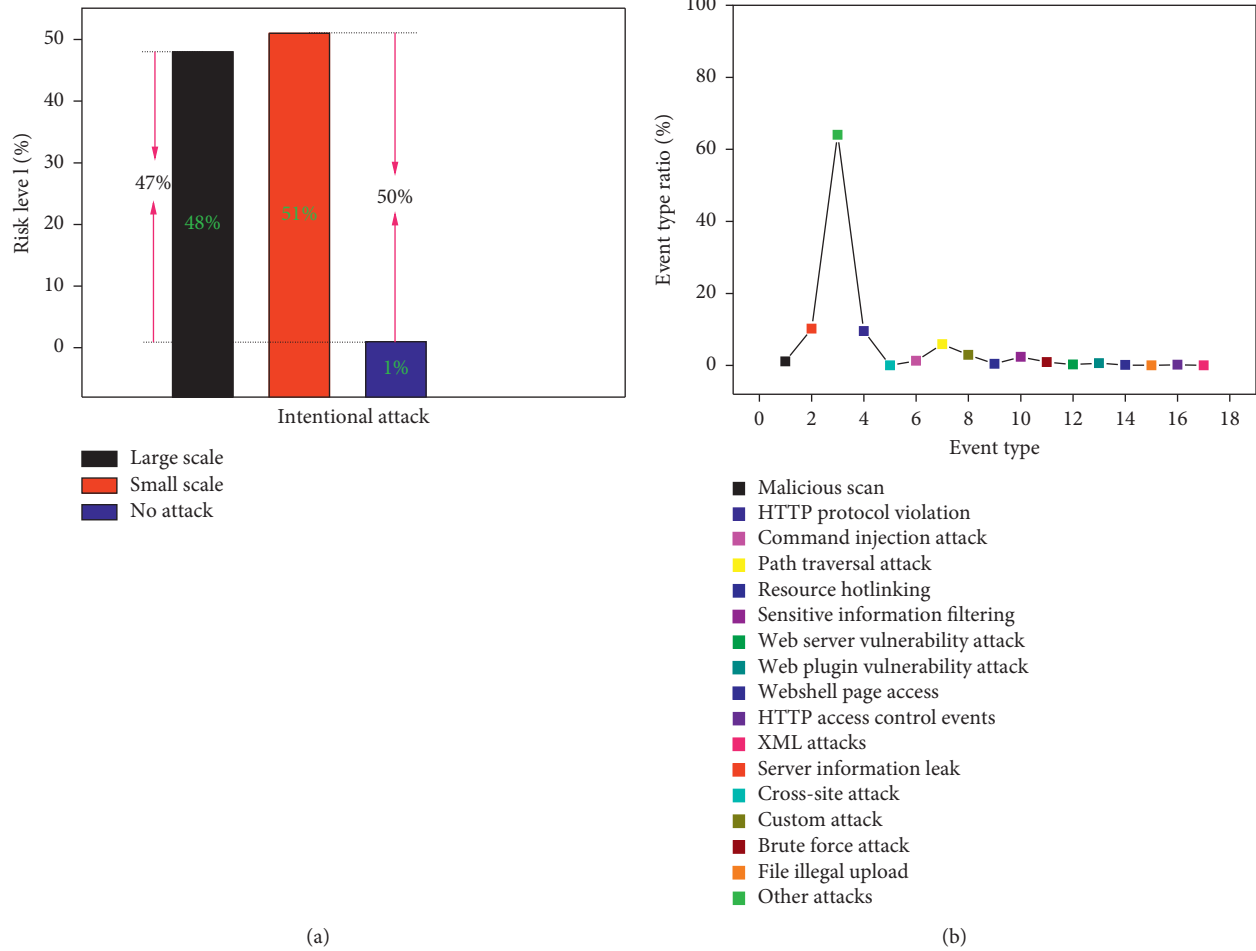


FIGURE 7: (a) The probability of high-, medium-, and low-risk events in the power systems; (b) the type of attack events that occurred.

A production subnet in the power systems is taken as an example, as shown in Figure 6. The entire production systems network is an internal private network, divided into multiple subareas, and its terminals of each subarea include personal PCs, cameras, electric vehicles, and temperature detectors. As can be seen from Figure 6, Web Application Firewall (WAF), the admission device, and other security devices are respectively connected at the core switching layer with the purpose to monitor abnormal traffics and prevent malicious attacks on the power systems.

4.1. Model Validation. WAF is one of the most commonly used network security protection devices in the power systems and is well known for its ability to detect malicious attacks in accordance with rules in a timely manner. To accurately predict the probability of attacks of different scales every day, a total of 454 sets of real-time data are extracted with a time step of 10 minutes.

The average value of the high-, medium-, and low-risk events in the WAF is used as the basis for judging, so as to

infer the frequency of various attacks inside the power systems every day. Here, the high-, medium-, and low-risk attacks correspond to the large-, small-, and no-attack strategies of the hackers in turn. The calculated daily probability of high-, medium-, and low-risk attack events is 48%, 51%, and 1%, as shown in Figure 7(a). A total of 27 possible combined attacks are discussed in this article among which the odds of launching large-scale, small-scale, and no-attack by attackers are $9/30$, $18/30$, and $3/30$, with the conclusion that the basic attack strategy of the hackers is mainly small-scale strategy combining with large-scale and no-attack strategy followed occasionally. The probability of high- and medium-risk events in Figure 7(a) is 48% and 51% higher than the probability of no security event, respectively. The data in Figure 7(a) show that the probability of no-attack on the power systems is extremely small, which is very close to the change trend inferred from the theoretical model. Obviously, the actual attack data in the power systems also confirm the results of the attack and defense model, which indicates that the models constructed in the article are consistent with the actual production environment.

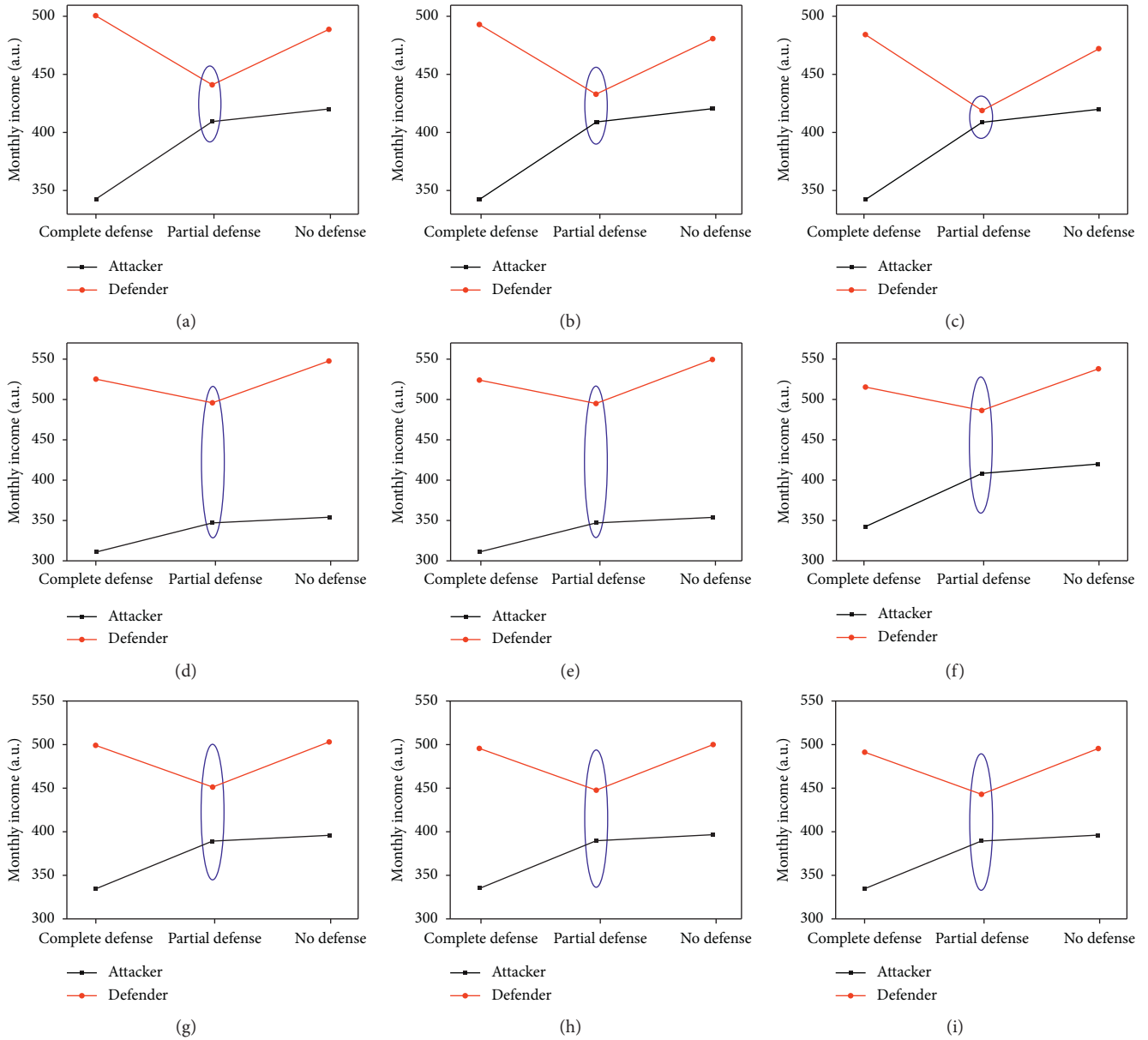


FIGURE 8: Under (a) complete-defense of host, (b) partial-defense of host, and (c) no-defense of host, both sides benefit when the hackers launch different attack strategies; under (d) complete-defense of host, (e) partial-defense of host, and (f) no-defense of host, both sides benefit when the hackers launch different attack strategies; under (g) complete-defense of network, (h) partial-defense of network, and (i) no-defense of network, both sides benefit when the hackers launch different attack strategies.

Figure 7(b) reveals 17 types of attack events that occurred in 454 sets of data, which are recorded by waf in the experimental stage.

4.2. Building Revenue Function. Taking the value of the above risk types as empirical values, the monthly revenue function formula (13) of both the sides is given, and then we discuss how the power systems should take precautions to maximize its revenue:

$$f(I) = 30 * (0.48x + 0.51y + 0.1z). \quad (13)$$

Among which, x , y , and z correspond to the final revenues obtained by the attack side using large-scale, small-scale, and no-attack. Assuming that the attack plan chosen by the hackers is the same within one month, thus the total monthly revenue of the power systems can be calculated. Considering that in actual production, the hackers are less likely to launch an attack on a single target and most of them would take the form of a combined attack. Therefore, this article only discusses the revenues of the both parties in the dual model.

In the first place, the revenues of both parties in the host and network attack defense model are discussed. The three

models corresponding to complete-, partial-, and no-defense of the host are described in Figures 8(a) and 8(c), respectively. The changes in the three figures are exactly the same, both of them reach the Nash equilibrium during partial-defense. The probability of this situation in Figure 8(c) in the actual production environment is almost zero. Thus, only when the host is in complete-defense and the network is in partial-defense can the power systems gain abundant revenues. Therefore, this defense mode should be the main one in the next month.

Then, the revenues of both parties in the host and management attack and defense model are discussed. The three models corresponding to complete-, partial-, and no-defense of the host are shown in Figures 8(d) and 8(f), respectively. The data trends in the three figures are the same as those in Figure 8, which implies that the power systems have the largest gain when the host side is in complete-defense and management side is in partial-defense. It is clear that this defense mode should be dominated in the next month.

Finally, the revenues of both parties in network and management attack and defense model are discussed. The three models corresponding to complete-, partial-, and no-defense of the network are shown in Figures 8(g) and 8(i), respectively. The data trends in the three figures are the same as in Figures 8(d) and 8(f), which also means that the power systems have the largest gain when the network side is in complete-defense and the management side is in partial-defense. In the next month, this defense mode should be the optimal choice in the power systems.

5. Conclusions and Future Work

In summary, the idea of modeling cyber-attack and defense as contending for power system benefit is proposed for the first time. Then, we have taken a subnet of the power systems as a case and employed the attack data in the actual production environment to verify the 27 dual attack-defense models constructed in this paper. By this approach, we derive the monthly benefit function applicable to this environment and calculated the benefits of both sides and the best defense mode on the power systems in the next month. Additionally, this paper analyzes in detail the possible attacks from the perspective of the attackers and evaluates the impact on the power systems, thereby changing its defense strategy from passive to proactive; we explored the optimum proactive defense strategy for the power systems from the angle of the game between the attacker and defender. In the next step, we will combine with actual production business processes to further study more specific proactive strategies to achieve the transition from qualitative defense to quantitative defense strategies.

Data Availability

No dataset was used in this experiment. The supporting data for the experimental results are all from WAF equipment, and the data template on the official website cannot be seen (the link may have expired). See the hyperlink (<https://pan.baidu.com/s/1cxUE51JPnwS3KG5i0VE-Ew>; Extraction code: Jw25) for the specific data the authors extracted.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] C. Kang, J. Huang, Z. Zhang et al., "An automatic algorithm of identifying vulnerable spots of internet data center power systems based on reinforcement learning," *International Journal of Electrical Power & Energy Systems*, vol. 121, Article ID 106145, 2020.
- [2] N. Moreira, E. Molina, J. Lázaro, E. Jacob, and A. Astarloa, "Cyber-security in substation automation systems," *Renewable and Sustainable Energy Reviews*, vol. 54, pp. 1552–1562, 2016.
- [3] J. X. Jacob, X. Zhang, and X. Q. Zhu, "The exploration of mobile target defense method in a diversified environment," *Electric Power Information and Communication Technology*, vol. 16, no. 6, pp. 1–5, 2018.
- [4] M. Hu, H. K. Boddapati, and S. Prakriya, "Performance off cluster-based multi-hop underlay networks with energy harvesting nodes," *IET Communications*, vol. 14, no. 9, pp. 1476–1484, 2020.
- [5] S. Vuppala, A. E.-D. Mady, and A. Kuenzi, "Moving target defense mechanism for side-channel attacks," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1810–1819, 2020.
- [6] C. X. Liu, X. S. Ji, and J. X. Wu, "A mimic defense mechanism for mobile communication user data based on MSISDN virtualization," *Chinese Journal of Computers*, vol. 41, pp. 275–287, 2018.
- [7] S. Radha, G. J. Bala, and P. Nagabushanam, "FRAME routing with game theory optimization for wireless networks," *International Journal of Communication Systems*, 2019.
- [8] Q. Y. Zhu and T. Basar, "Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: games-in-games principle for optimal cross-layer resilient control systems," *IEEE Control Syst Mag*, vol. 35, no. 1, pp. 46–65, 2015.
- [9] Y. Zhao, J. L. Yu, and M. F. Ban, "A distribution-market based game-theoretical model for the coordinated operation of multiple microgrids in active distribution networks," *International Transactions on Electrical Energy Systems*, vol. 30, pp. 1–11, 2020.
- [10] S. Rass and Q. Zhu, "GADAPT: a sequential game-theoretic framework for designing defense-in-depth strategies against advanced persistent threats," *Lecture Notes in Computer Science*, vol. 9996, pp. 314–326, 2016.
- [11] J. T. Chen, C. Touati, and Q. Y. Zhu, "A dynamic game analysis and design of infrastructure network protection and recovery," *Electrical Engineering and Systems Science*, vol. 45, pp. 125–128, 2017.
- [12] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas, "A hybrid stochastic game for secure control of cyber-physical systems," *Automatica*, vol. 93, pp. 55–63, 2018.
- [13] R. Pappas, Q. Chen, L. Chen, J. Xiong, and D. Wu, "A privacy-preserving personalized service framework through Bayesian game in social IoT," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8891889, 13 pages, 2020.
- [14] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2019.