IJNS
International Journal of Network Security

ISSN 1816-353X (Print)
ISSN 1816-3548 (Online)

Vol. 24, No. 2 (March 2022)

INTERNATIONAL JOURNAL OF NETWORK SECURITY

PUBLISHING OFFICE

Min-Shiang Hwang
Department of Computer Science & Information Engineering, Asia University, Taichung 41354, Taiwan, R.O.C.
Email: mshwang@asia.edu.tw

International Journal of Network Security is published both in traditional paper form (ISSN 1816-353X) and in Internet (ISSN 1816-3548) at http://ijns.jalaxy.com.tw

PUBLISHER: Candy C. H. Lin

Volume: 24, No: 2 (March 1, 2022)

International Journal of Network Security

An Efficient Differential Privacy Method with Wavelet Transform for Edge Weights of Social Networks

Jun Yan[1,2], Hai Liu[3], and Zhenqiang Wu[1]

(Corresponding author: Zhenqiang Wu)

School of Computer Science, Shaanxi Normal University, Xi'an 710119, China[1]
School of Mathematics and Computer Applications, Shangluo College, Shangluo 72600, China[2]
Guizhou Provincial Key laboratory of Public Big Data, Guizhou University, Guiyang 550025, China[3]
Email: zqiangwu@snnu.edu.cn

(Received Mar. 12, 2021; Revised and Accepted Aug. 14, 2021; First Online Nov. 13, 2021)

Abstract

With more and more attention paid to privacy preservation in social networks, many effective methods based on differential privacy have been presented. To preserve the sensitive information of edge weights in a weighted social network, we combine the differential privacy with wavelet transform and devise a DPEW (Differential Privacy based Edge weight with Wavelet Transform) method. The proposed method satisfies differential privacy and provides better data utility. In this method, WTDP (Wavelet Transform based on Differential Privacy) algorithm can achieve differential privacy preserving while less noise is added on the edge weights. In addition, the properties of the original graph are maintained by EDU (Enhancing Data Utility) algorithm. The experimental results show that the DPEW method achieves ϵ-differential privacy and reduces the information loss of the edge weight than other methods.

Keywords: Differential Privacy; Perturbation Ratio; Wavelet Transform; Weighted Social Network

1 Introduction

Nowadays, with the widespread popularity of mobile Internet, the Internet has become more and more close to our life. For example, about ten years ago, we usually went out to shopping in the supermarket, but now we can buy almost anything online at home. With the help of Internet, we can use a mobile phone to record and analyze information about running and walking, which can promote us to keep training. Especially, the online social network, which merges our online and offline lives, has played a more and more important role in our daily lives. Through the Facebook platform, the largest social network in the world, which has 2 billion monthly active users, we can make friends and share information anytime and anywhere. In addition, we can also do many things on social networks, such as shopping, advertising, Video Live Broadcasting and so on, which bring great convenience to our lives. More importantly, recently, with the development of VSNs (Vehicular Social Networks), the numerous applications for VSNs will occupy our daily lives. Hence, we can say, social networks online have greatly changed our lifestyle.

However, social networks online appear to be a double-edged sword:although they bring us a lot of conveniences, they also present a great challenge to us. For example, social networks online contain a great quantity relationships between every individual, such as schoolmate relation, colleague relation and so on. Moreover, these relationships may be relevant to all kinds of attributes (weight values, directions), which are personal sensitive information. As a result, when social networks are published without privacy preserving, it is a great possibility to infer the hidden and secret information with high accuracy, which results in many privacy leakage problems. Therefore, in order to preserve privacy of social networks, it is critical for us to present effective privacy preserving methods.

For preserving privacy of social networks, we can abstract a social network as a graph where the vertices represent the individuals and the edges represent relationships among individuals. Therefore, the graph modified methods are widely used in this area. A simple method is naiVe anonymization method which is presented by Hay [6]. To resist connection-based attacks, edge and vertex modification methods which randomly perturbed the original graph are proposed. For example, Hay [6] also proposed the random perturbation algorithm. In addition, a random perturbation method called Blockwise random Add/Delete was developed by Ying [19]. In order to improve the data utility, many constrained perturbation methods to satisfy some desired constraints were

developed, such as a spectrum preserving approach [20] and the k-anonymity model [12, 14].

In addition to the methods mentioned above, a well-known privacy-aware computation method called differential privacy [3], which can defend against any attacks based on background knowledge, has been widely applied for privacy preserving in many areas, such as the smart grid information system [4]. Due to being able to provide rigorous privacy guarantee, the popular differential privacy mechanism has been used to publish sensitive graph data, such as the number of triangles and k-stars [16]. Different from adding noise to the graph data, the differential privacy technology based on the Stochastic Kronecker Graph Model [8] was introduced to provide privacy preserving on a graph, which can improve data utility significantly [13]. Thus, as a useful privacy preserving technology, differential privacy technology can also be widely applied in weighted social networks.

In weighted social networks, for preserving the weight value of edges which indicate the degree of intimacy between individuals, the researchers have proposed many methods which can be divide into two classes. One class is based on the K-anonymous technology [22], and the other is based on the differential privacy. Compared with the K-anonymous technology, the differential privacy will result in insufficient data utility because of a lot of Laplacian noise when preserving social networks. In order to improve the data utility, all kinds of transformation methods are used in differential privacy. As a special transformation method, the wavelet transform can not only provide rigorous privacy guarantee but also can keep a certain degree data utility, which is presented by Xiao [18].In this paper, a well known Haar wavelets, which has the simplest orthogonal basis among all discrete wavelet transforms, is used to achieve differential privacy while reducing the perturbation of noise. Further more, the shortest distances between some important nodes in the weighted social network are kept unchanged in our method, which make our method to have a better data utility than other methods.

In summary, our contributions are described as follows:

1) We devise a DPEW (Differential Privacy based Edge weight with Wavelet Transform) method, which satisfies the differential privacy with better data utility.

2) We propose two algorithms. The first algorithm is WTDP (Wavelet Transform based on Differential Privacy), which can achieve differential privacy preserving while adding less noise on the edge weights. The second is EDU (Enhancing Data Utility), which is an algorithm that can maintain the properties of original weighted social network to enhance the data utility.

3) We present the PR (perturbation ratio) to evaluate the different methods in privacy preserving, which is more intuitively than parameter ϵ. and we compare our method with other different methods in the synthesis and real data sets.

In the following sections, the organization of this paper is outlined as follows. In Section 2, we introduce many kinds of privacy preserving methods which are applied in social networks. we give some preliminaries, including the differential privacy, the wavelet transform and the properties of graph in Section 3. Section 4 describes our privacy preserving method and algorithms. The experimental results and comparison are illustrated in Section 5. Finally, Section 6 concludes this paper.

2 Related Work

Since differential privacy was put forward by C. Dwork, a lot methods based on differential privacy have been proposed, which can be classified into edge differential privacy and node differential privacy [7]. As one of the most important properties of a graph, the degree distribution was protected by an efficient algorithm based on K-edges differential privacy, which was provided by Hay [1]. In order to protect another important statistics, such as subgraph counts, Zhang [10] introduced a new method which guarantees differential privacy by using ladder framework. Comparing with edge differential privacy, node differential privacy could satisfy stronger privacy guarantees, but preform lower data utility. In order not to change original data significantly, Kasiviswanathan [5] use several techniques to develop node differential privacy algorithms, which improve the data utility. In the method based on node differential privacy [21], the aggregation technique and the cumulative histogram technique were used to obtain better data utility in publishing the degree distribution.

In weighted social networks, being a significant property, edge weighs can be protected by many techniques, such as K-anonymous technique and differential privacy technique. To prevent attacks based on background, k-anonymity of nodes method [15] and [k1, k2]-shortest path privacy method [17] have been presented. Based on differential privacy, a method with the MB-CI strategy is proposed to protect edge weight, which enhanced the accuracy and utility of the published data [9].

Due to having a better property on privacy preserving and data utility, the wavelet transform as a signal transformation method can be used for data perturbation. To prevent the privacy in certain data from being revealed in data mining, Liu [11] presented a method based on wavelet transform which maximized data utility. For better privacy, Xiao [18] achieved differential privacy by combining wavelet transform. In privacy preserving clustering, Dishabi [2] proposed a different privacy based method with daubechies-2 wavelet transform.

3 Preliminaries Knowledge

In this paper, a weighted social network is regarded as a simple, undirected, weighted graph $G=(V, E, W)$, where

V=(v_1, v_2, $\cdots$, v_n) with each v_i representing an individual in social network, E=(e_1, e_2, $\cdots$, e_n) with each e_i describing a relationship between two v_i, W=(w_1, w_2, $\cdots$, w_n), each w_i describes a kind of attribute of e_i.

Definition 1. *(Neighboring graph). For two weighted graphs G_1 =(V_1,E_1,W_1), G_2=(V_2,E_2,W_2), if $|V_1 \bigoplus V_2|+|E_1 \bigoplus E_2|=2$, where $\bigoplus$ is Exclusive - OR operation, we can say that G_1 and G_2 are neighbors. Assuming V_1=V_2, if $|E_1 \bigoplus E_2|=2$, G_1 and G_2 are neighbors. In this paper, we assume that there are two different edges between two graphs G_1 and G_2. In general, because the difference of two graphs is two edges, edge differential privacy is used to achieve differential privacy.*

Definition 2. *(Differential Privacy). If a randomized algorithm* R *satisfies ϵ-differential privacy, there is a conclusion as following:*

$$P_r[\mathrm{R}(G_1) \in \mathrm{T}] \leq exp(\epsilon) P_r[\mathrm{R}(G_2) \in \mathrm{T}].$$

where T $\subseteq$ *Range(*R*), G_1,G_2 are neighbors and ϵ is a privacy budget. In order to achieve differential privacy, we comply with Laplace Mechanism to add the Laplace noise on the result of queries.*

Definition 3. *(Laplace Mechanism). In a weighted graph, assuming a query function is* Q*, where* G *is a weighted graph,* w *is a weight sequence of* G*. Given two G_1 and G_2, which are neighbors, according to the definition 1, the sensitive of* Q *is as following:*

$$\Delta Q = max_{G_1,G_2} \|Q(G_1) - Q(G_2)\|_1.$$

The Laplace mechanism is a special technique, which adds Laplace noise to the output of a query function to satisfy differential privacy.

$$R(G) = Q(G) + Lap(\frac{\Delta Q}{\epsilon}).$$

where the Laplace noise satisfies Laplace distribution, which is described as follows.

$$d(x) = \frac{1}{2b} exp(-\frac{|x-\mu|}{b})$$

where μ=0 b=$\frac{\Delta Q}{\epsilon}$, μ is a horizontal deviation, b is a scale variable and x is a variable.

Definition 4. *(Post-Processing). Given a randomized algorithm* A *that satisfies ϵ-differential privacy,* F *is an arbitrary randomized function. Then a randomized algorithm* F$\cdot$ A *satisfies ϵ-differential privacy.*

Definition 5. *(Wavelet transformationation). As a special technique in mathematics,* DWT*(discrete wavelet transformation) can divide an input discrete sample into* AC*(approximation coefficients) and* DC*(detail coefficients),which respectively correspond to the low frequency and high frequency parts of the original sample. Such a wavelet decomposition process can be carried out recursively up to the expected decomposition. On the contrary,* IDWT *(inverse discrete wavelet transformation) can recombine* AC *and* DC *into the original sample. The* AC *and* DC *are respectively defined as follows:*

$$AC = \sum_{k=-\infty}^{\infty} x(\mathrm{k}) g(2l-k)$$

$$DC = \sum_{k=-\infty}^{\infty} x(\mathrm{k}) h(2l-k).$$

where g *is a low frequency filter and* h *is a high frequency filter. In AC, ac_{jl} denotes the* j-th *approximation coefficients in the* l-th *level of decomposition.*

In this paper, we choose a well known Haar wavelets, which has the simplest orthogonal basis among all discrete wavelet transforms. The scaling function of Haar wavelet transform is represented by S, which is indicated as follows:

$$s = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & otherwise \end{cases}$$

The mother wavelet of Haar wavelet transform is denoted by M, which is described as follows:

$$M(x) = \begin{cases} 1 & 0 \leq x < 0.5 \\ -1 & 0.5 \leq x < 1 \\ 0 & otherwise \end{cases}$$

Definition 6. *(degree centrality). The degree centrality of node v_i is the sum of the number of adjacent nodes, which is denoted by* Cd(v_i)*. Formally, the degree centrality is given by:*

$$Cd(v_i) = \sum_{j=1}^{n} a(v_i, v_j)$$

where a(v_i,v_j) denotes the edge between node v_i and node v_j. In general, a(v_i,v_j) equals 1.

In a weighted graph, the weight degree centrality of node v_i is the sum of the weights of edges which connect node v_i. We can define the weight degree centrality as:

$$Cd_w(v_i) = \sum_{j=1}^{n} w(v_i, v_j)$$

where w(v_i,v_j) represents the weight of edge between node v_i and node v_j.

Definition 7. *(The between centrality). The between centrality of node v_i is given by:*

$$\mathrm{Cb}(v_i) = \sum_{j,k} \frac{p(v_j, v_i, v_k)}{p(v_j, v_k)}$$

where $p(v_j, v_k)$ denotes the number of shortest paths between node v_j and node v_k, $p(v_j, v_i, v_k)$ is the number of shortest paths between node v_j and node v_k which go

through v_i. In a weighted graph, the betweenness centrality is defined by:

$$Cb_w(v_i) = \sum_{j,k} \frac{p_w(v_j, v_i, v_k)}{p_w(v_j, v_k)}$$

where $p_w(v_j, v_k)$ is the sum of edge weight in shortest paths between node v_j and node v_k, $p_w(v_j, v_i, v_k)$ is the sum of edge weight in shortest paths between node v_j and node v_k which go through v_i.

4 DPEW Method

In this section, we propose a DPEW method to preserve edge weight privacy in a social weighted network when it is published. In this method, we devise two algorithms: WTDP algorithm and EDU algorithm. WTDP algorithm can achieve differential privacy preserving while adding less noise on the edge weights, and EDU algorithm can maintain the properties of original weighted social network. In addition, we prove that DPEW method satisfies differential privacy while obtaining better data utility.

4.1 The Model of DPEW Method

For preserving edge weight privacy in the weight social networks, we introduce a practical method that combines wavelet transform with differential privacy, which also maintains the shortest path length between some important nodes in original weight social network unchanged. The frame structure of proposed method is illustrated in Figure 1.

In this model, the input is an original weighted social network, which has sensitive information: the edge weights. The output is a published weighted social network which is preserved by differential privacy. In order to provide rigorous privacy guarantee, wavelet transformation and differential privacy are combined in WTDP algorithm, which satisfies ϵ-differential privacy. Owing to the deficiency of data utility caused by the Laplace noise, this model present EDU algorithm which aims to preserve the character of original weighted social network for enhancing data utility. Therefore, our method can not only preserve the privacy of the original weighted social network but can also keep the data utility of the published weighted social network.

4.2 WTDP and EDU Algorithm

4.2.1 WTDP Algorithm

With the application of wavelet transformation, we propose a new algorithm which adds less noise to achieve differential privacy for weights of edges. In this algorithm, we first get a weight sequence of edge weights W and apply wavelet transformation on it, After that, we gain the approximation coefficients and the detail coefficients of sequences wavelet transformation. According to Laplace mechanism, we add Laplace noise to the approximation coefficients to achieve differential privacy. Thus, we can generate a preserved weighted graph by using inverse wavelet transformation. The frame structure of WTDP algorithm is illustrated in Figure 2 and Algorithm 1.

Algorithm 1 The WTDP algorithm

Input: The original weighted social network: G=(V, E, W);the best decomposition level: C ; privacy budget: ϵ;
Output: The noised weighted social network: G'=(V, E, W')
1: $Wm \leftarrow Max(W)$
2: Sensitivity: Δ f = $\frac{Wm}{C}$
3: $b \leftarrow \frac{\Delta f}{\epsilon}$
4: $Wa \leftarrow$ wavelet transform in W
5: for wa_i in Wa:
6: A Laplace noise $n_i \leftarrow$ Laplace(b)
7: Adding n_i on wa_i
8: $Wa' \leftarrow Wa$
9: $W' \leftarrow$ Inverse Wavelet transformation Wa' and Wd
10: Return noised weighted social network: G'=(V, E, W')

In WTDP Algorithm, we input a social weighted network G, the best decomposition level C and privacy budget ϵ. For preserving edge weights in this social weighted network G, we first get the max edge weight in line 1. Then, from line 2 to line 3 the scale parameter b in Laplace distribution is obtained. Line 4 describes the wavelet transformation of W and gains approximation coefficients Wa. In line 5-8, for every wa_i in Wa, a Laplace noise n_i is added and we get the perturbed approximation coefficients Wa'. By using the perturbed approximation coefficients Wa' and the original detail coefficients, line 9 describes the inverse wavelet transformation and obtains a noised edge weight sequence W'. Finally, we get an edge weight sequence W' which is preserved by WTDP algorithm.

4.2.2 EDU Algorithm

For minimizing the changes of edge weight and achieving better data utility, we propose an algorithm to keep some characters of original social weighted network unchanged in the noised social weighted network. In order to achieve this purpose, we select some important nodes which possess large degree centrality and betweenness centrality in original social weighted network and make the shortest distance between these selected important nodes in the noised social weighted network equal to that in the social weighted network graph. Thus, the description of EDU algorithm is given in Figure 3 and Algorithm 2.

In this algorithm, we use the composite parameters Nc to evaluate the importance of node, which is shown as follows.

$$Nc = \sqrt{Cd_w(v_i)^2 + Cb_w(v_i)^2}$$

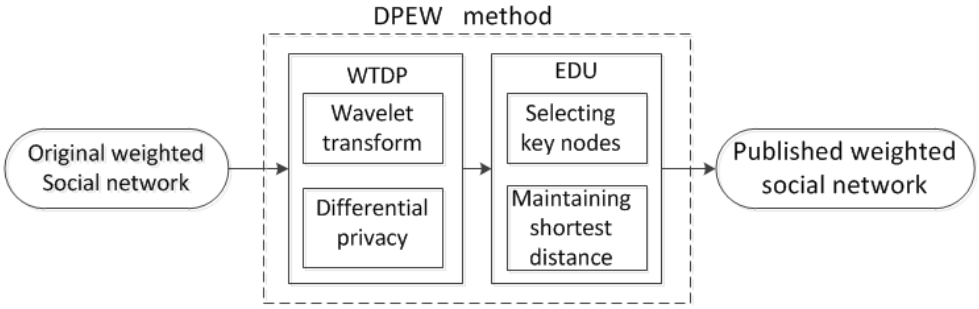


Figure 1: The model of DPEW method

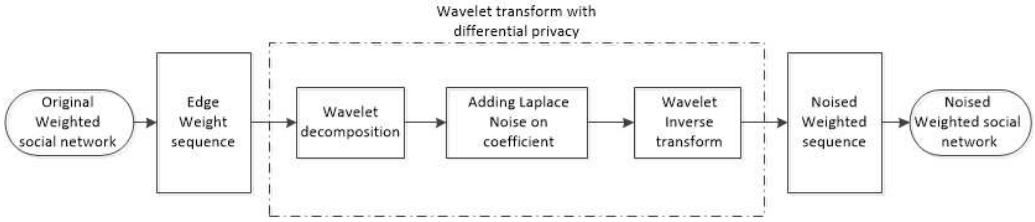


Figure 2: The WTDP algorithm

where $Cd_w(v_i)$ denotes the weight degree centrality of v_i, $Cb_w(v_i)$ represents the betweenness centrality of v_i. The larger Nc, the more important node v_i is. When the shortest distance between two important nodes is maintained unchanged in a noised social weighted network, the shortest distance between any two nodes in a noised social weighted network will be closed to that in the original social weighted network. If more shortest distance are kept unchanged in noised weight social network, there may be less perturbation on the published weight social network.

For the EDU algorithm, the detail is described as follows. Firstly, we select k important nodes according to the value of Nc, then we work out the shortest distance among those nodes in the original social weighted network and in the noised social weighted network. Secondly, we make the edge weights in the noised social weighted network to be equal to these in the original social weighted network. At last, we get a perturbed social weighted network which not only preserves the original social weighted network but also gains better data utility.

In EDU algorithm, we calculate Nc of nodes in an original weight social network in line 1. Line 3 generates the important node sequence D' after selecting k nodes according to value of Nc. Line 4 to line 9 outline how to keep the shortest distance of nodes in D' unchanged in $G"$. Line 4 and line 7 calculate the shortest path of nodes in the original weight social network and get the edges list L_e and edge weights list W_e in shortest path, The modification of the edge weight in G^* is described in line 8 and line 9, which keep the length of the shortest path in G unchanged. Finally, this algorithm returns a perturbed weight social network $G"$ which preserves the shortest distance length in the original weight social network.

Algorithm 2 The EDU algorithm

Input: weighted social network: G=(V, E, W); noised weight social network: G'=(V, E, W')

Output: perturbed weight social network: $G"$=(V, E,$W"$)

1: Calculating Nc of nodes in G
2: Selecting k nodes from V according to value of Nc
3: Generating a sequence D' containing k important nodes
4: for i in D':
5: for j in D':
6: Calculate shortest path from node i to node j in G
7: Get edges list L_e and edge weights list W_e in shortest path
8: Keep the length of shortest path unchanged in G'
9: Modifying the edge weight in G'
10:Return perturbed weighted social network: $G"$=(V, E,$W"$)

4.3 Theoretical Analysis

Given a weighted social network G=(V, E, W), W is the edge weight sequence. After transforming the W into wavelet domain, we get Wa, which denotes the approximation coefficients, and Wd, which represents the detail coefficients.

Assume that two weighted social networks, G_1 and G_2 are neighbors, and the difference between G_1 and G_2 is two edges. Let $Q(\bullet)$ be a query function $Q: G \rightarrow Wa$, so $Q(G_1)=Wa_1$, $Q(G_2)=Wa_2$. According to the definition

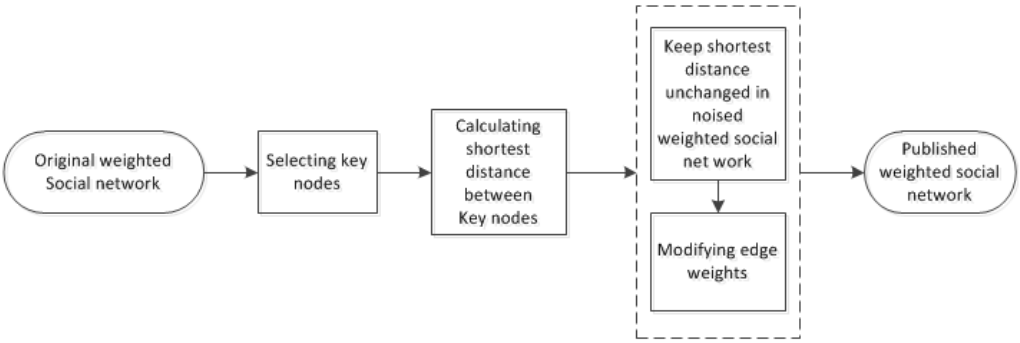


Figure 3: The EDU algorithm

of sensitive, we get the sensitivity of Q:

$$\begin{aligned}\Delta Q &= max_{G_1,G_2}|Q(G_1) - Q(G_2)|_1 \\ \Delta Q &= max|Wa_1 - Wa_2|_1 \\ &= max|(wa_{11}, wa_{12}, ..., wa_{1m}) - (wa_{21}, wa_{22}, ..., wa_{2m})| \\ &= \frac{\Delta(w_{max} - w_{min})}{2^{ND}}\end{aligned}$$

where w_{max} and w_{min} are the maximum and minimum values in the W, ND is the level of decomposition. Then, we add the Laplace noise to the output of Q in accordance with the Laplace Mechanism, where LA is the Laplace Mechanism.

Let Pr $[G_1]$ denotes the probability density function of LA (G_1, Q, ε), and Pr $[G_2]$ indicates the probability density function of $LA(G_2$, Q, $\varepsilon)$. Then, the proof is described as follows.

$$\begin{aligned}&\frac{Pr[LA(G_1)]}{Pr[LA(G_2)]} = \frac{Pr[\eta(G_1)]}{Pr[\eta(G_2)]} \\ &= \frac{Pr[R - Q(G_1)]}{Pr[R - Q(G_2)]} \\ &= \frac{\frac{1}{2\frac{\Delta Q}{\epsilon}} exp(-\frac{|R-Q(G_1)|}{\frac{\Delta Q}{\varepsilon}})}{\frac{1}{2\frac{\Delta Q}{\epsilon}} exp(-\frac{|R-Q(G_2)|}{\frac{\Delta Q}{\varepsilon}})} \\ &= \frac{exp(-\frac{|R-Q(G_1)|}{\frac{\Delta Q}{\varepsilon}})}{exp(-\frac{|R-Q(G_2)|}{\frac{\Delta Q}{\varepsilon}})} \\ &= exp(\frac{\varepsilon|R - Q(G_1)|}{\Delta Q} - \frac{\varepsilon|R - Q(G_2)|}{\Delta Q}) \\ &= exp(\frac{\epsilon(|R - Q(G_1)| - |R - Q(G_2)|)}{\Delta Q}) \\ &\leq exp(\frac{\epsilon(|Q(G_1) - Q(G_2)|)}{\Delta Q}) \\ &\leq exp(\frac{\epsilon\Delta Q}{\Delta Q}) = e^{\epsilon}\end{aligned}$$

Therefore, we can achieve differential privacy preserving for Wa. After conducting IDWT on the noised approximation coefficients Wa and detail coefficients, we generate a noised edge weight sequence $W^{'}$, which can be used to construct a noise weight social network. For better data utility, we carry out modifying the noised $W^{'}$ while maintaining the properties of original weighted social network. At last, we construct a published weighted social network based on the $W^{''}$ to preserve the original weighted social network. According to the requirement of post-processing, we achieve differential privacy preserving for original weighted social network with better data utility.

5 Experiments and Results

5.1 Datasets

In experiments, there are two kinds of data sets: the synthetic weighted network data and the real weighted network data. All the data sets used are shown below.

1) Synthetic weighted network. In the experiment with synthetic data, we generate two random graphs with 500 nodes and 1,000 nodes, which are randomly connected to each other with probability p=0.3. For each edge, an integer weight is assigned randomly in the range [1,200]. We call this synthetic graph as Random Graph.

2) Real weighted network. In the Windsurfers network, there are 43 nodes and 336 edges, which contains interpersonal contacts between windsurfers in southern California during the fall of 1986. The Infectious SocioPatterns dataset contains the daily cumulated networks represented in the Infectious SocioPatterns visualization, which includes 307 nodes and 1924 edges. The weights associated with the edges are the number of 20 seconds intervals during which close-range face-to-face proximity has been detected. The high-energy theory collaborations (Hetc) data set is a weighted network of coauthorships between scientists posting preprints on the High-Energy Theory E-Print Archive between Jan 1, 1995 and December 31, 1999. It has 5835 nodes and 13815 edges.

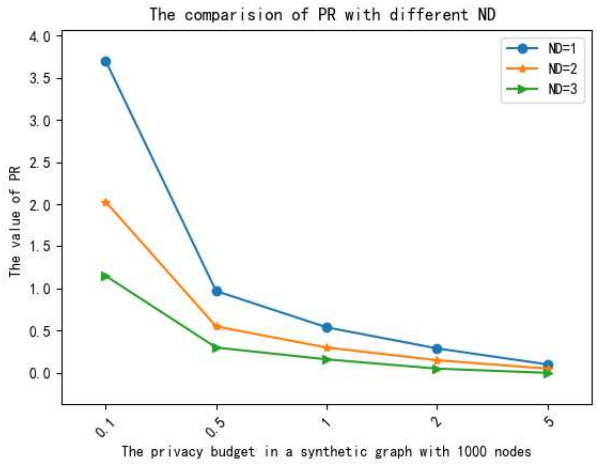


Figure 4: Comparison of different *ND* in a synthetic graph

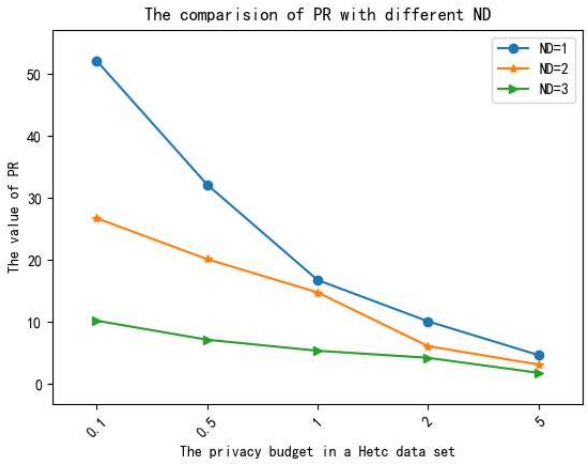


Figure 5: Comparison of different *ND* in a Hetc data set

5.2 Privacy Evaluation

For evaluating our method, we take advantage of the PR (perturbation ratio) to measure the performance in privacy preserving. Moreover, we compare our method with other methods in privacy preserving.

5.2.1 Privacy Measurement

In this section, the perturbation ratio (*PR*) is used to measure the performance of preserving privacy, which is the ratio of *(Wp -W)* to *W*, where the perturbed edge weight is *Wp* and the original edge weight is *W*. The larger *PR*, the better privacy preserving.

$$PR = \frac{W_p - W}{W} = \frac{\sum_{i=1}^{n} |wp_i - w_i|}{\sum_{i=1}^{n} w_i}$$

Meanwhile, *ND*, the number of wavelet decomposition, which is equal to the level of decomposition *l*, can determine the sensitivity in our method. If we want better privacy preserving, we can decrease *ND*, otherwise, we can increase *ND* for better data utility.

In order to compare with our method, we select four methods: GR method(Gaussian randomization method) [10], k-anonymization mothod [15], Edge-DP method(edge-differential privacy based method), DP-MB method(differential privacy based on merger of barrels method) [9]. In the experiments, we set ϵ in [0.1, 0.5, 1, 2, 5], the *ND* is set in [1,2, 3]. Due to the uncertainty of the noise, we execute all data sets 10 times by using our approach and other approaches to average out the results.

5.2.2 Privacy Analysis

In privacy analysis, we first conduct the experiment on the synthetic data sets and real data sets by using our method and keep the experiment results in Table 1. As shown in Table 1, when *ND* is 1, ϵ is 5, the *PR* in synthetic graphs with 500 nodes and 1,000 nodes is respectively 0.11 and 0.10, while the *PR* in three real data sets is 0.63, 0.10, 0.10 respectively. If we decrease ϵ from 5 to 0.1, the *PR* in synthetic graphs with 500 nodes increases from 0.11 to 3.72, as does the PR in other data sets. This result indicates that the smaller ϵ, the better privacy preserving. When ϵ is 2, if we increase *ND* from 1 to 3, the *PR* in synthetic graphs with 500 will decrease from 0.30 to 0.05, as will the PR in other data sets, which shows that *ND* can affect the privacy preserving.

Next, we describe the changing tendency of *PR* in our method with ϵ varying in Figure 4 and Figure 5, where *ND* is from 1 to 3, respectively. As shown in Figure 4, when ϵ increases from 0.1 to 5, *PR* in synthetic graphs with 1000 nodes decreases simultaneously no matter how much *ND* is. When ϵ is a fixed value, the value of *PR* declines as the value of *ND* increases, which means the wavelet transformation can control the privacy preserving of the method. In Figure 5, the *PR* in a Hetc data set is same as that in Figure 4. By using *PR*, it is clear that our method can achieve privacy preserving for the edges. In addition, for better understanding the comparison among different methods in a synthetic graph with 1000 nodes and a Hetc data set, the details are demonstrated in Figure 6 and Figure 7 respectively. When ϵ is from 0.1 to 5, *ND* is 2, the *PR* obtained by these methods in a synthetic graph with 1000 nodes is illustrated in Figure 6, where the *PR* in our method is larger than DP-MB method and smaller than that in the other three methods. Specially, the change of *PR* in GR method is small when ϵ increases from 0.5 to 5. In a Hetc data set, *PR* in our method, which is described in Figure 7, is smallest in those five methods no matter what ϵ is. All the results show that our method can improve data utility owing to adding less noise to edge weights.

To sum up, the experimental results show that our method can achieve differential privacy preserving for weighted graphs. In addition, by using the wavelet transform in our method, we can control the Laplace noise

Table 1: The value of PR in our method

ND	ϵ	synthetic data1	synthetic data2	Windsurfers network	SocioPatterns	Hetc
1	0.1	3.72	3.70	21.66	40.15	52.11
1	0.5	0.97	0.97	5.82	11.74	32.16
1	1	0.55	0.54	3.06	8.15	16.78
1	2	0.30	0.29	1.06	6.26	10.12
1	5	0.11	0.10	0.63	5.05	4.69
2	0.1	2.05	2.03	9.14	16.72	26.73
2	0.5	0.56	0.55	2.79	6.95	20.14
2	1	0.29	0.30	1.30	5.69	14.78
2	2	0.16	0.15	0.78	4.99	6.12
2	5	0.05	0.05	0.37	4.63	3.16
3	0.1	1.18	1.15	4.86	9.11	10.22
3	0.5	0.30	0.30	1.24	5.17	7.16
3	1	0.16	0.16	0.74	4.72	5.38
3	2	0.05	0.05	0.34	4.48	4.46
3	5	1.5e-16	1.5e-16	0.18	5.24	2.18

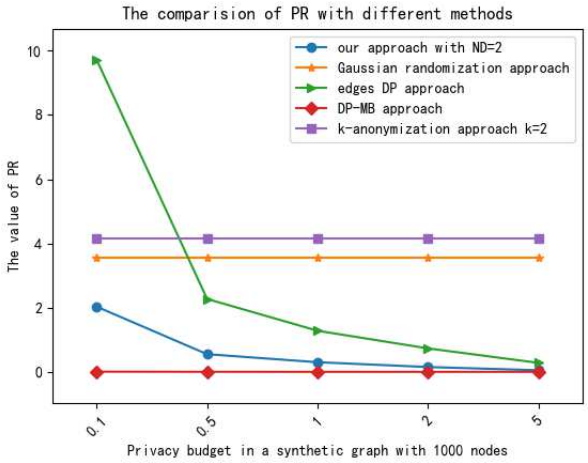


Figure 6: Comparison of different method in a synthetic graph(ND=2)

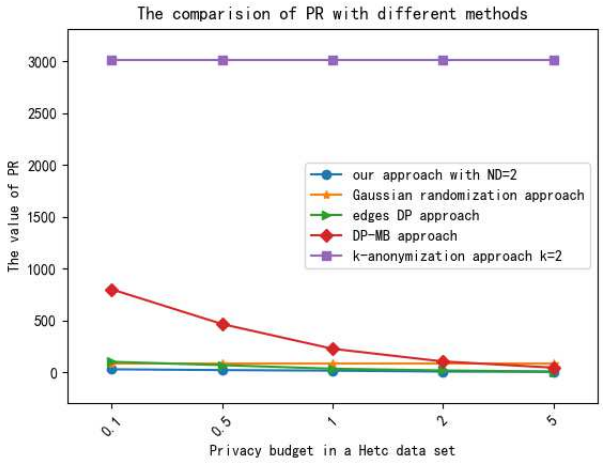


Figure 7: Comparison of different method in a Hetc data set(ND=2)

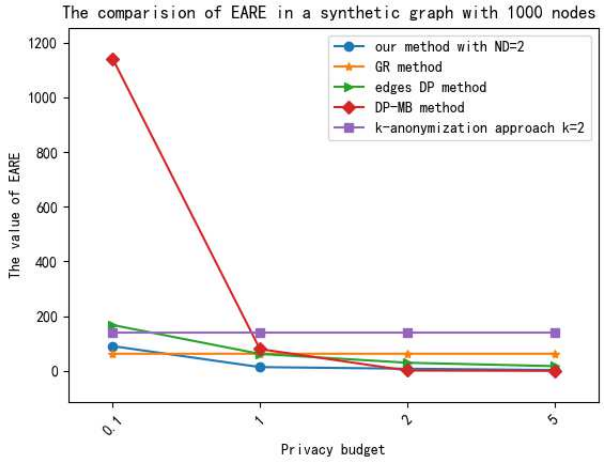


Figure 8: Comparison of *EARE* in a synthetic graph

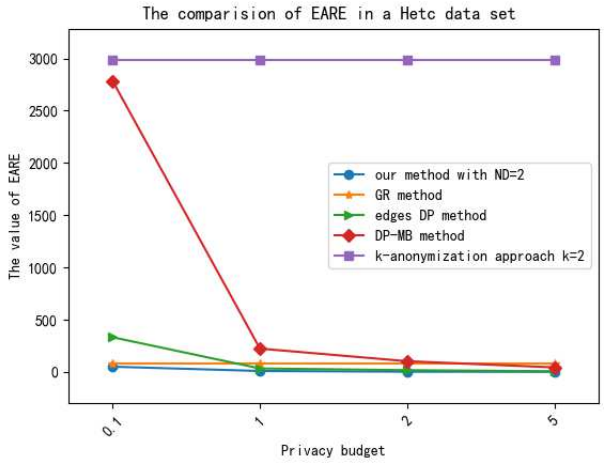


Figure 9: Comparison of *EARE* in a Hetc data set

which is added on the edge weights. Therefore, our method gains better privacy preserving than DP-MB method and has better data utility than GR method, k-anonymization mothod, and edge-DP method.

5.3 Utility Evaluation

In this section, we define some metrics of the graph to evaluate the data utility. Then, after analyzing and discussing our method in data utility, we compare our method with other methods.

5.3.1 Utility Metrics

To evaluate the data utility, we use four metrics: *EARE* (edge average relative error), *NARE* (node average relative error), *ASD* (average shortest distance) and *KSPL* (Keeping Shortest Path length).

1) *EARE*. *EARE* is the average relative error of edge weight, which indicates the edge change caused by privacy preserving. The smaller the value, the higher data utility.

$$EARE = \frac{\sum_{i=1}^{n} |Wp_i - W_i|}{n}$$

where Wp_i denotes the edge weight in published weighted social network, W_i represents the edge weight in original weighted social network.

2) *NARE*. *NARE* is the average relative error of node weight, which describes the node change caused by perturbation. The smaller *NARE*, the better data utility.

$$NARE = \frac{\sum_{i=1}^{n} |VWp_i - VW_i|}{n}$$

where VWp_i denotes the node weight in published weighted social network, VW_i represents the node weight in original weighted social network.

3) *ASD*. *ASD* is an important property of the weighted graph, which is the average shortest distance among all pairs of nodes.

$$ASD = \sum_{s,t \in V} \frac{d(s,t)}{n(n-1)}$$

where V is the set of nodes in G, $d(s,t)$ is the shortest path from s to t, and n is the number of nodes in G.

4) *KSPL*. *KSPL* is the proportion of unchanged shortest path length.

$$KSPL = \frac{Np^{'}}{Np}$$

where Np^{*} is the number of unchanged shortest path lengthen in in published weighted social network, while *Np* denotes the total number of shortest path length in original weighted social network. The larger *KSPL*, the more the shortest path lengths are unchanged.

5.3.2 Utility Analysis

In this experiment, we set ϵ in [0.1,1,2, 5] and *ND* in 2. In addition, four methods, such as GR method (Gaussian randomization method) [10], k-anonymization mothod [15], Edge-DP method(edge-differential privacy based method), DP-MB method(differential privacy based on merger of barrels method), are used for comparison. Due to the uncertainty of the noise, we conduct our method and other methods 10 times to average out the results.

In the utility analysis, first of all, we discuss the experimental results gained by our method. As shown in Table 2, when ϵ is 0.1, *ND* is 2, the results of *EARE*, *NARE*, *ASD*, *KSPL* in a synthetic data set with 500 nodes

Table 2: Utility metrics in our method ND=2

data sets	***metrics***	**original network**	**ϵ=0.1**	**ϵ=1**	**ϵ=2**	**ϵ=5**
synthetic data1	*EARE*	0	90.30	13.73	7.79	2.90
synthetic data1	*NARE*	0	8072.21	553.83	202.96	53.32
synthetic data1	*ASD*	6.12	4.66	4.20	4.56	5.09
synthetic data1	*KSPL*	1	0.08	0.12	0.16	0.19
synthetic data2	*EARE*	0	88.92	14.13	8.23	3.03
synthetic data2	*NARE*	0	15990.36	1197.27	460.88	147.30
synthetic data2	*ASD*	4.57	4.13	3.12	3.23	3.59
synthetic data2	*KSPL*	1	0.06	0.12	0.14	0.16
Windsurfers network	*EARE*	0	76.88	15.58	10.92	5.37
Windsurfers network	*NARE*	0	1195.11	226.58	158.77	75.79
Windsurfers network	*ASD*	2.19	12.25	4.95	3.77	2.53
Windsurfers network	*KSPL*	1	0.065	0.066	0.065	0.047
SocioPatterns	*EARE*	0	124.05	41.60	36.24	32.40
SocioPatterns	*NARE*	0	1526.51	493.70	429.52	384.53
SocioPatterns	*ASD*	4.66	71.17	23.33	16.26	11.26
SocioPatterns	*KSPL*	1	0.05	0.08	0.10	0.11
Hetc	*EARE*	0	221.12	21.16	10.12	4.32
Hetc	*NARE*	0	1081.24	112.99	58.94	18.17
Hetc	*ASD*	4.57	253.78	72.45	28.74	13.22
Hetc	*KSPL*	1	0.04	0.05	0.06	0.08

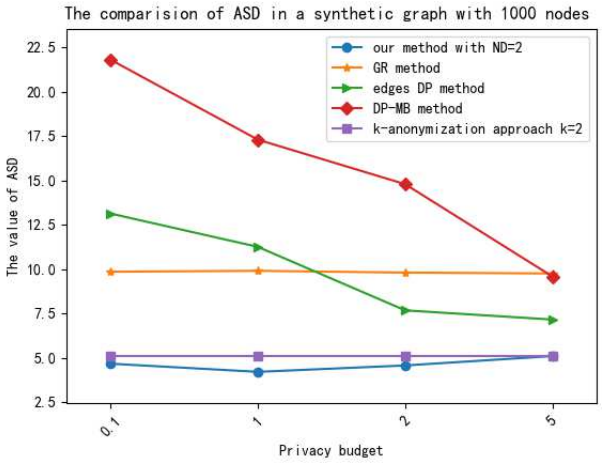


Figure 10: Comparison of *ASD* in a synthetic graph

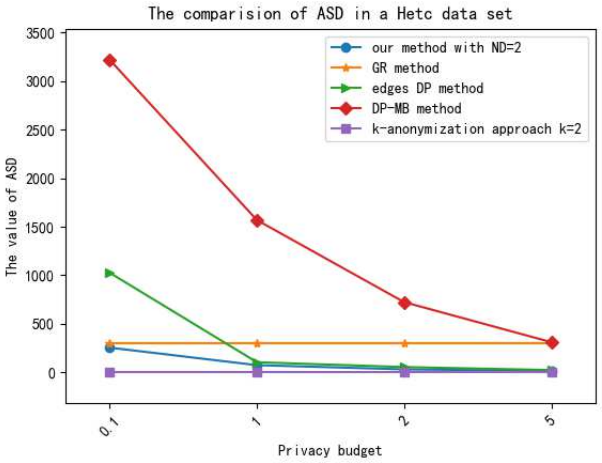


Figure 11: Comparison of *ASD* in a Hetc data set

are 90.30, 8072.21, 4.66, 0.08, respectively. When ϵ is increased to 5, the values of *EARE* and *NARE* will decrease simultaneously together. In addition, the value of *ASD* is close to that in the original graph because *ASD* is mostly associated with the number of the selected important nodes. In particular, the value of *KSPL* changes slightly. Furthermore, it is worth noting that the results in other data sets are equivalent to those in the synthetic data set with 500 nodes. All the results state clearly that data utility will be improved with the increase of ϵ.

Next, particularly when ϵ is changed from 0.1 to 5 in a synthetic data set with 1000 nodes and a Hetc data set, the comparison of these methods is illustrated by these figures as follows. As shown in Figure 8 and Figure 9, the values of *EARE* in different methods decline with ϵ increasing. Specially, in Figure 8, the value of *EARE* in our method is smaller than that in the k-anonymization mothod, the edge-DP method, and the Edge-DP method when ϵ increases from 0.1 to 5, while it is larger than that in the GR method as ϵ is less than about 0.5. In Figure 9, we can see that the value of *EARE* in our method is smallest in these mothods. As illustrated in the Figure 10, the change of the *ASD* in different methods and the values of *ASD* in other four methods are larger than that in our method. For example, when ϵ equals to 1, the values in other four methods are 5.21,10.01, 11.24,16.45, respectively, while the value in our method is 4.2. In addition, when ϵ is smaller than 1, the value of *ASD* obtained by our method is the larger than that in k-anonymization mothod, which is shown in the Figure 11. Therefore, the result shows that our method can obtain a better data utility compared with other methods.

Finally, owing to the wavelet transform and post-processing, the results indicate that our method can achieve better performance in data utility than GR method, k-anonymization mothod, DP-MB method and edge-DP method. Therefore, we can see that our method can improve the data utility while satisfying the differential privacy.

6 Conclusions

For preserving the privacy data of social networks, the differential privacy which is able to provide strict privacy guarantee has been extensively applied. Compared with other differential privacy based methods, in this work, we focus on achieving differential privacy for edge weights while keeping the data utility as much as possible and publishing a preserved weighted social network. Therefore, we propose a method which combines wavelet transform with differential privacy. In this method, we first apply the wavelet transform on the edge-weight sequence and add the Laplace noise to the wavelet coefficients, then we take advantage of inverse wavelet transform to realize differential privacy. At last, for modifying the error of shortest distance of noised graph, a special algorithm is used to improve the data utility. In addition, we present two algorithms: WTDP algorithm and EDU algorithm. To evaluate the performance of our method, the *PR* is used to evaluate the privacy preserving of different methods when ϵ is fixed. Moreover, the theory analysis and experimental results show that our method not only satisfies ϵ-differential privacy but also improves data utility. In the future, due to the perturbation caused by the stochastic noise in ϵ-differential privacy, we must work hard to maintain the property of graph while satisfying ϵ-differential privacy.

Acknowledgments

This study was supported by the National Natural Science Foundation of China under grant No. 61962033, the Natural Science Basic Research of Shaanxi province under grant No. 2020JM-288, the Fundamental Research Funds for the Central Universities under grant No. GK201704017 and No. GK201903011.We are gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] W. Y. Day, N. Li and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *Proceedings of 10th Theory of Cryptography Conference*, pp. 133-138, 2016.

[2] M. R. E. Dishabi, M. A. Azgomi, "Differential privacy preserving clustering using Daubechies-2 wavelet transformation," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 13, no. 14, pp. 1550028, 2015.

[3] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquiium on Automata, Languages and Programming*, pp. 1-12, 2006.

[4] S. Guo, M. Wen, X. H. Liang, "A differentially private k-means clustering scheme for smart grid," *International Journal of Network Security*, vol. 23, no. 1, pp. 126-134, 2021.

[5] M. Hay, C. Li, G. Miklau, *et al.*, "Accurate estimation of the degree distribution of private networks," in *Proceedings of 9th IEEE International Conference on Data Mining*, pp. 169-178, 2009.

[6] M. Hay, G. Miklau, D. Jensen, *et al.*, "Anonymizing social networks," in *Computer Science*, pp. 07-19, 2007. (`https://scholarworks.umass.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1175;context=cs_faculty_pubs`)

[7] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, A. Smith, "Analyzing graphs with node-differential privacy," in *Proceedings of 10th Theory of Cryptography Conference*, pp. 457-476, 2013.

[8] J. Leskovec, C. Faloutsos, "Scalable modeling of real graphs using kronecker multiplication," in *Proceedings of the 24th International Conference on Machine Learning*, pp. 497-504, June 2007.

[9] X. Y. Li, J. Yang, Z. L. Sun, "Differential privacy for edge weights in social networks," *Security and Communication Networks*, vol. 2017, pp. 1-10, 2017.

[10] L. Liu, J. Wang, J. Liu, *et al.*, "Privacy preserving in social networks against sensitive edge disclosure," *Preserving Data Privacy in Knowledge Discovery*, 2008. (`https://www.researchgate.net/publication/228972647_Privacy_preserving_in_social_networks_against_sensitive_edge_disclosure`)

[11] L. Liu, J. Wang, J. Zhang, "Wavelet-based data perturbation for simultaneous privacy-preserving and statistics-preserving," in *Proceedings of International Conference on Data Mining Workshops*, pp. 27-35, 2008.

[12] T. Ma, Y. Zhang, J. Cao, J. Shen, M. Tang, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, "KDVEM: A k-degree anonymity with vertex and edge modification algorithm," *Computing*, vol. 97, no. 12, pp. 1165-1184, 2015.

[13] D. Mir, R. N. Wright, "A differentially private estimator for the stochastic kronecker graph model," in *Proceedings of Joint EDBT/ICDT Workshops*, pp. 167-176, 2012.

[14] F. Nagle, L. Singh, A. Gkoulalas-Divanis, "EWNI: Efficient anonymization of vulnerable individuals in social networks," in *Proceedings of the 16th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pp.359-370, 2012.

[15] M. E. Skarkala, M. Maragoudakis, S. Gritzalis, "Privacy preservation by k-anonymization of weighted social networks," in *Proceedings of International Conference on Advances in Social Networks Analysis and Mining*, pp. 423-428, 2012.

[16] C. Task, C. Clifton, "What should we protect? Defining differential privacy for social network analysis," *State of the Art Applications of Social Network Analysis*, pp. 139-161, 2014.

[17] Y. C. Tsai, S. L. Wang, T. P. Hong, "Extending [K1, K2] anonymization of shortest paths for social networks," in *Proceedings of International Conference on Multidisciplinary Social Networks Research*, pp. 187-199, 2015.

[18] X. Xiao, G. Wang, J. Gehrke, "Differential privacy via wavelet transformations," *IEEE Transactions on Knowledge & Data Engineering*, vol. 23, no. 8, pp. 1200-1214, 2011.

[19] X. Ying, X. Wu, "On link privacy in randomizing social networks," in *Proceedings of Pacic-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pp. 28-39, 2009.

[20] X. Ying, X. Wu, "Randomizing social networks: A spectrum preserving approach," in *Proceedings of the SIAM International Conference on Data Mining*, pp.739-750,2008.

[21] J. Zhang, G. Cormode, C. M. Procopiuc, *et al.*, "Private release of graph statistics using ladder functions," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 731-745, 2015.

[22] Y. B. Zhang, Q. Y. Zhang, Y. Yan, *et al.*, "A k-Anonymous location privacy protection method of polygon based on density distribution," *International Journal of Network Security*, vol. 23, no. 1, pp. 57-66, 2021.

Biography

Jun Yan received the M.S. degree in College of Earth Exploration Science and Technology, Jilin University(2007). He is currently pursuing the Ph.D. degree in College of Computer Sciensce, Shaanxi Normal University. His research interests include network security and privacy preserving.

Hai Liu received his B. S. degree (2012) and M.S. degree (2015) from Guizhou University,and obtained Ph.D. degree (2019) from School of Computer Science, Shaanxi Normal University. His main research interest includes privacy protection.

Zhenqiang Wu received his B.S. degree in 1991 from Shaanxi Normal University, China, and received his M.S. and Ph.D degrees in 2002, and 2007 respectively, all from Xidian University, China. He is currently a full professor of Shaanxi Normal University, China. Dr. Wu's research interests include computer communications networks, mainly wireless networks, network security, anonymous communication, and privacy protection etc. He is a member of ACM and senior of CCF.

Formal Security Evaluation and Improvement of BACnet/IP Protocol Based on HCPN Model

Tao Feng, Si-Meng Zhao, and Xiang Gong
(Corresponding author: Si-Meng Zhao)
School of Computer and Communication & Lanzhou University of Technology
Lanzhou, Gansu 730000,China
Email: 1078293826@qq.com
(Received Feb. 2, 2021; Revised and Accepted Dec. 8, 2021; First Online Feb. 19, 2022)

Abstract

The BACnet/IP protocol is widely used in building automation systems. However, while realizing remote monitoring of building equipment, it faces a more significant threat of cyber attacks. To solve the current situation of the building automation system being attacked, this paper takes the building automation protocol "BACnet/IP equipment authentication" as the research object and proposes a model detection method based on the colored Petri net theory and the Dolev-Yao attack method, and evaluates and improves the security of the protocol. First, the protocol's device authentication mechanism was verified for consistency based on the Petri net theory and CPN Tools model tools. Then the Dolev-Yao attack model was introduced to evaluate the security of the original model of the protocol. It was found that the protocol had replay, deception, and tampering 3 Man-in-the-middle attack vulnerability. Finally, a new solution is proposed for the loopholes in the protocol, which uses timestamps to enhance the security of the conversation between devices. It again uses the CPN model detection tool to verify the security of the new solution. Through verification, it can be found that the new solution improves the difficulty of the attack, thereby ensuring the ability of the BACnet/IP protocol to resist replay, deception, and tampering attacks.

KeyWords: BACnet/IP Protocol; Colored Petri Nets; Formal Analysis; Safety Assessment; Time Stamp

1 Introduction

Intelligent buildings provide people with a safe, efficient, comfortable and convenient building environment [26]. The development of building intelligence has always been closely related to the Internet. With the rapid development of Internet technology, Internet-based intelligent building application technologies and products have emerged in large numbers. In order to manage building equipment more efficiently and conveniently, the interconnection between the BACnet network [20] and the Internet has become an inevitable trend. However, because the BACnet protocol itself did not consider security issues at the beginning of its design, many common network attack methods can also threaten the BAS (Building Automation System) [15] connected to the Internet, directly attacking the BACnet server or building automation equipment, causing network paralysis.

In the field of smart buildings, the BACnet network has been proven to be insecure [12, 23]. Literature [18] designed a context-aware intrusion detection framework based on abnormal behavior analysis widely deployed in BACnet networks to accurately detect abnormal events triggered by network attacks or any functional failures. Literature [9] elaborated on the vulnerable types of BACnet and Internet connection and gave corresponding countermeasures under different attack methods. The literature [11, 16] conducted a detailed study on the identification and authentication, denial of service, eavesdropping, and buffer overflow in the core functions of the BACnet network, and added remote management technologies for corporate intranet and Internet connections. However, the above-mentioned documents have not done formal modeling analysis on the internal data transmission security of the protocol and proposed effective security assessment methods and improvement schemes.

Therefore, this article uses formal methods to conduct security modeling research on the BACnet/IP protocol, which is widely used in the field of building automation systems, but lacks effective security research. The main work includes the following three aspects:

1) A model detection method based on colored Petri net theory and Dolev-Yao attack method was proposed;

2) The BACnet/IP protocol equipment authentication mechanism was analyzed in detail and formalized description. The protocol is modeled by modeling tools, and the consistency of the model was verified. The Dolev-Yao attacker model was introduced to evalu-

ate the security of the protocol, and the loopholes in the protocol were found;

3) A new scheme for introducing a time stamp mechanism to enhance the strength of device authentication was proposed for the security vulnerabilities in the protocol, and the new scheme was also re-verified.

2 Related Theories and Concepts

2.1 Overview of BACnet/IP Protocol

At present, BACnet network has been widely used in intelligent buildings, however, in order to better realize the management and control of building equipment, it is extremely necessary to interconnect multiple different BACnet network equipment through the Internet to realize the interconnection and interoperability of equipment in different regions. The BACnet standard currently uses two technologies to realize the connection between the IP network and the BACnet network: One is the PAD technology, and the other is the BACnet/IP technology [21].

PAD technology is a relatively mature and developed BACnet network and Internet interconnection technology. However, this technology lacks flexibility in its use. First, when the configuration data of the network device is changed, all data and information of the PAD device must also be completely modified to maintain the correctness of the routing information. Second, when new devices are dynamically added to the BACnet network via the Internet, it is difficult and costly. In view of the shortcomings of PAD, the IP working group of the BACnet Standards Committee (SSPC135) developed a more scalable and flexible BACnet interconnection protocol [17]. The interconnection protocol is the BACnet protocol based on IP, referred to as the "BACnet/IP" standard for short. The main function of BACnet/IP is to directly encapsulate BACnet data packets into IP frames for data transmission. The BACnet/IP protocol mainly includes the following 7 parts:

1) Proposed and described in detail the concept of a BACnet network composed of one or more IP subnets.

2) Describes the use of BACnet non-confirmed services for the management of local, remote and global broadcasts between BACnet/IP networks and non-BACnet/IP networks.

3) A new device is defined, called BACnet Broadcast Management Device (BBMD), for broadcast management.

4) By defining a new protocol layer called BACnet Virtual Link Layer (BVLL), BACnet/IP communication is realized.

5) Provides a method for external devices to access the BACnet/IP network.

6) Provides routing between BACnet/IP networks and non-BACnet/IP networks.

7) Specifies the routing between multiple BACnet/IP networks. The BACnet architecture after joining the BACnet/IP protocol is shown in Figure 1.

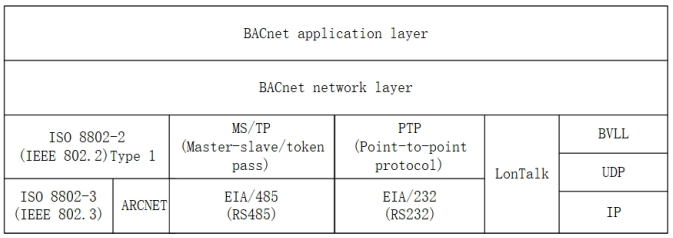


Figure 1: BACnet/IP architecture diagram

2.2 Colored Petri Nets and Modeling Tools

The concept of Petri net was first proposed by German scientist Carl Adam Petri, and then many expanded concepts appeared, such as time Petri net, stochastic Petri net, CPN, *etc.* [1, 22]. Colored Petri Net (Colored Petri Net, CPN) is an advanced form of basic Petri Net. It increases the ability of P/T net to simulate and describe the model. It also has strict formal definitions. In terms of formal analysis of security protocols, CPN, as a high-level network system, integrates the advantages of Petri nets and high-level programming language abstract mechanisms, has the ability to describe types and hierarchical structures, and can describe a complex system compactly and simplified [25]. It has a rich and flexible color set and function definition, suitable for the standardized definition of the message in the security protocol. CPN modeling can also perform incremental syntax checking and code generation, which to a certain extent also ensures the correctness of the model. The formal modeling tool CPN Tools has features such as editing, simulation, state space analysis and performance analysis, and can accurately locate errors generated through a feedback mechanism. This tool is developed by Aarhus University in Denmark based on Design/CPN. The user graphical interface (GUI) is designed using good man-machine interface technology. It can not only edit, simulate and analyze colored Petri nets, but also support time CPN and hierarchical CPN, with the help of CPN tools, users can not only model easily, but also simulate and analyze parallel systems [3, 4].

3 BACnet/IP Protocol Equipment Certification HCPN Modeling

3.1 BACnet/IP Protocol Equipment Authentication Message Flow Model

The BACnet/IP protocol device authentication message flow (MSC) model is shown in Figure 2. RequestKey means requesting the session key from the key server, Ks means the session key distributed by the key server to devices A and B, IDA means the device number of A, IDB means the device number of B (the device number is unique), and Ka means The master key of device A, Kb represents the master key of device B (only shared with the key server). Authenticate represents the authentication request service between peer entities, Pseudo Random Number represents the pseudo-random number in the message, ComplexACK represents the complex response message, and Modified Random Number represents the modified random number in the response message.

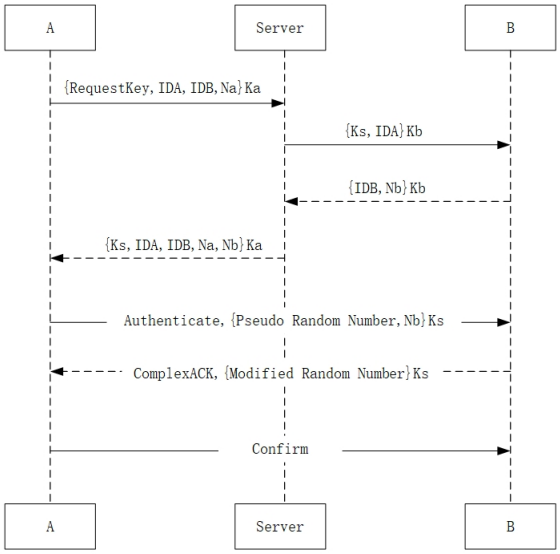


Figure 2: Authentication mode message flow model (MSC)

The equipment certification process is as follows:

1) Devices A and B run the DES algorithm to generate their own private keys Ka and Kb (only shared with the key server).

2) Device A initiates a "RequestKey" request to the key server to request the session key Ks for communication between devices A and B.

3) After receiving the request message from device A, the key server generates the session key Ks, uses the Kb of device B to encrypt Ks and IDA, and sends it to device B.

4) Device B uses Kb to encrypt and send its own device number IDB to the key distributor, and the key distributor verifies the device number of device B.

5) After decrypting using Kb, the key server first compares the obtained IDB with the one sent by device A. If they are consistent, use the Ka of device A to encrypt Ks and IDB and send them to device A.

6) Device A receives Ks and starts to authenticate device B. Device A initiates an Authenticate request. The protocol data part and Pseudo Random Number of this request are encrypted with Ks and sent to B.

7) Device B decodes the received authentication request of device A, changes the Pseudo Random Number to Modified Random Number, uses Ks to encrypt, and returns a ComplexACK message to device A.

8) Device A decrypts the received response message. If the ComplexACK packet contains the correct Modified Random Number, the device authentication is successful.

3.2 Message Analysis and Color Set Definition

The modeling process introduces multiple types of color set definitions. The color set ID defines the unique device numbers of devices A and B. The color set KEY defines the master keys (Ka and Kb) of the two devices and the session distributed by the key server. The key Ks, the color set NONCE defines the pseudo-random number Na and the modified random number Nb used in the equipment authentication phase. There are two encryption and decryption formats used in the communication process. One is that the key server uses the pre-shared master keys Ka and Kb of device A and device B to decrypt the data information obtained, and the other is device A and device B. B uses the master key Ka, Kb to encrypt its own device number and request information. According to the CPN ML language, use the product type color set definition to integrate the main information and the key according to a specific method to represent the encryption and decryption operations, thereby obtaining four message formats. On this basis, use the record (record) type color set definition for subsequent encrypted messages.Due to the complexity of the BACnet/IP protocol, there are many color sets defined. Here we only list some important related color set definitions, as shown in the Table 1.

3.3 HCPN Model Establishment

3.3.1 Top-Level Model

Based on the above analysis, the corresponding model diagram can be drawn, as shown in Figure 3. The double-

Table 1: Definition of color set for BACnet/IP protocol equipment certification

Category	Color set name	Color set definition
Preliminary preparation	ID	colset ID=with A— B;
	KEY	colset KEY=with Ka—Kb—Ks;
	NONCE	colset NONCE=with Na—Nb;
	CONFIG	colset CONFIG=product ID*ID;
	CRY1	colset CRY1=product KEY*KEY;
	CRY2	colset CRY2=product ID*KEY;
Key distribution	MSG1	colset MSG1=product CONFIG*KEY;
	MSG2	colset MSG2=product CRY2*KEY;
	MSG3	colset MSG3=product ID*KEY;
	MSG4	colset MSG4=product CRY2*KEY;
Equipment certification	ACK	colset ASK=record m:MSG*k:KEY;
	RSP	colset RSP=product NONCE*NONCE;
	RPL	colset RPL=record r:RSP*k:KEY;
	CFM	colset CFM=record n:NONCE*k:KEY;
Data	PACKET	colset PACKET=union MSG1+MSG2+MSG3+MSG4;

line rectangle in the figure is the alternative transition, and each substitution transition corresponds to a subpage of the physical layer. The three alternative transitions represent three different entities, from left to right: Device A, Key Server, and Device B. The places P1 to P7 represent the communication network. The top-level model completely simulates the BACnet/IP protocol device authentication session process, including the key distribution process and the device authentication process, as well as the communication data processing process.

3.3.2 Physical Layer Model

As shown in Figure 4, the model of entity A in BACnet/IP protocol equipment authentication includes 16 message places and 8 transitions. This model describes the sending and receiving process of the message initiator A requesting the key server Server to obtain the session key and initiating identity authentication to the message responder B. In the model, the indexAB and KEYA1 of the fusion place are used to configure the identity information and key generation parameters of the session initiator before the session occurs. Transition T1 integrates the message into MSG1 type and sends it to the port place p1;When the entity A initiates a session, the model is in the session participation state, and then the data will be sent to the responder, and the responder is entity B in the model;When the corresponding key server receives the message of entity B, it sends a message of type MSG4 to the port place p4. The initiator uses the shared key Ka with the key server to decrypt and obtain the session key Ks. Transition encA obtains the pseudo-random number Na from the place NumA and adds it to the data packet and integrates it into ACK format information and sends it to device B for identity authentication.

As shown in Figure 5, the model describes the receiving and processing process of entity B receiving the session key distributed by the key server and the data message that initiates identity authentication on entity A.When configuring the session participation mode, you need to introduce the index place, the specific function of the place is to set each entity and its identity information;After the data is sent to the message responder, that is, entity B participates in the session to perform Step 2, Step 3, and Step 6. Entity B in MSG2 decrypts the received data with the shared key Kb between the key server and obtains the session key Ks and the identity of the initiator entity A, and saves the obtained session key in the places KEYS2 and KEYS4. Then send an MSG3 type message to authenticate with the key server.

As shown in Figure 6, the model describes the process in which the key server Server distributes the session keys for identity authentication to entities A and B. The key server receives the message MSG1, judges the encryption key, if it is Ka, then performs a decryption operation to obtain the identity of the session initiator and the responder, and save it to the corresponding place indexB1. Then according to the identity of the responder in the message MSG1, synthesize and send the message MSG2, and send the session key Ks and the identity of the session initiator to the responder. The key server receives and uses the shared key Kb with the key server to decrypt the message MSG3 sent by the responder, and determine whether the responder's identity is correct.Finally, the server uses Ka to encrypt the session key Ks and responder identity according to the initiator identity in the message MSG1, synthesizes and sends the message MSG4.

3.3.3 Functional Consistency Verification of the Original Model

The State Space tool component in the CPN Tools tool can calculate the original model state space and generate a state space report. The details are shown in the

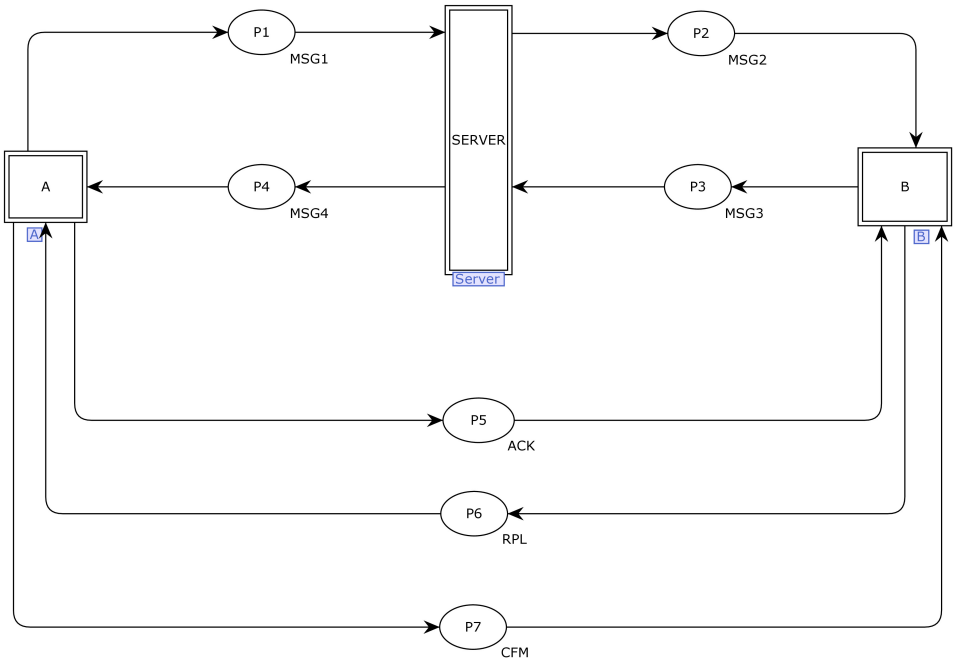


Figure 3: CPN top-level model of BACnet/IP protocol equipment certification

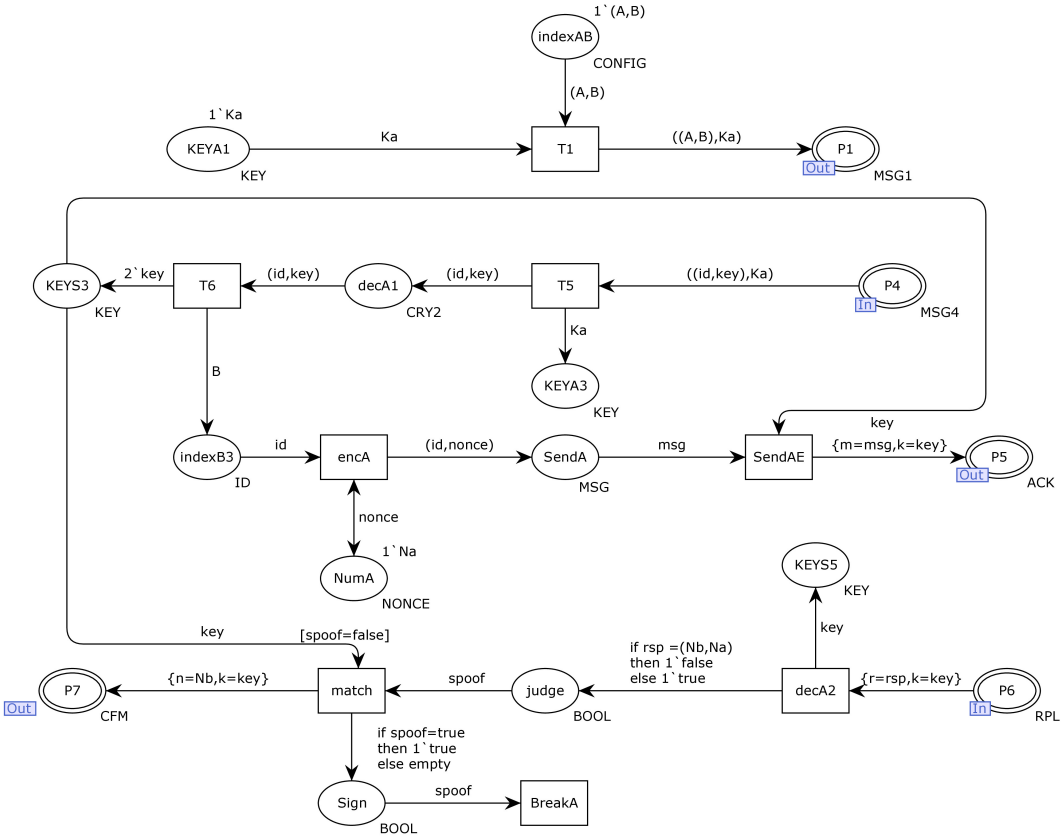


Figure 4: CPN model of entity A certified by BACnet/IP protocol equipment

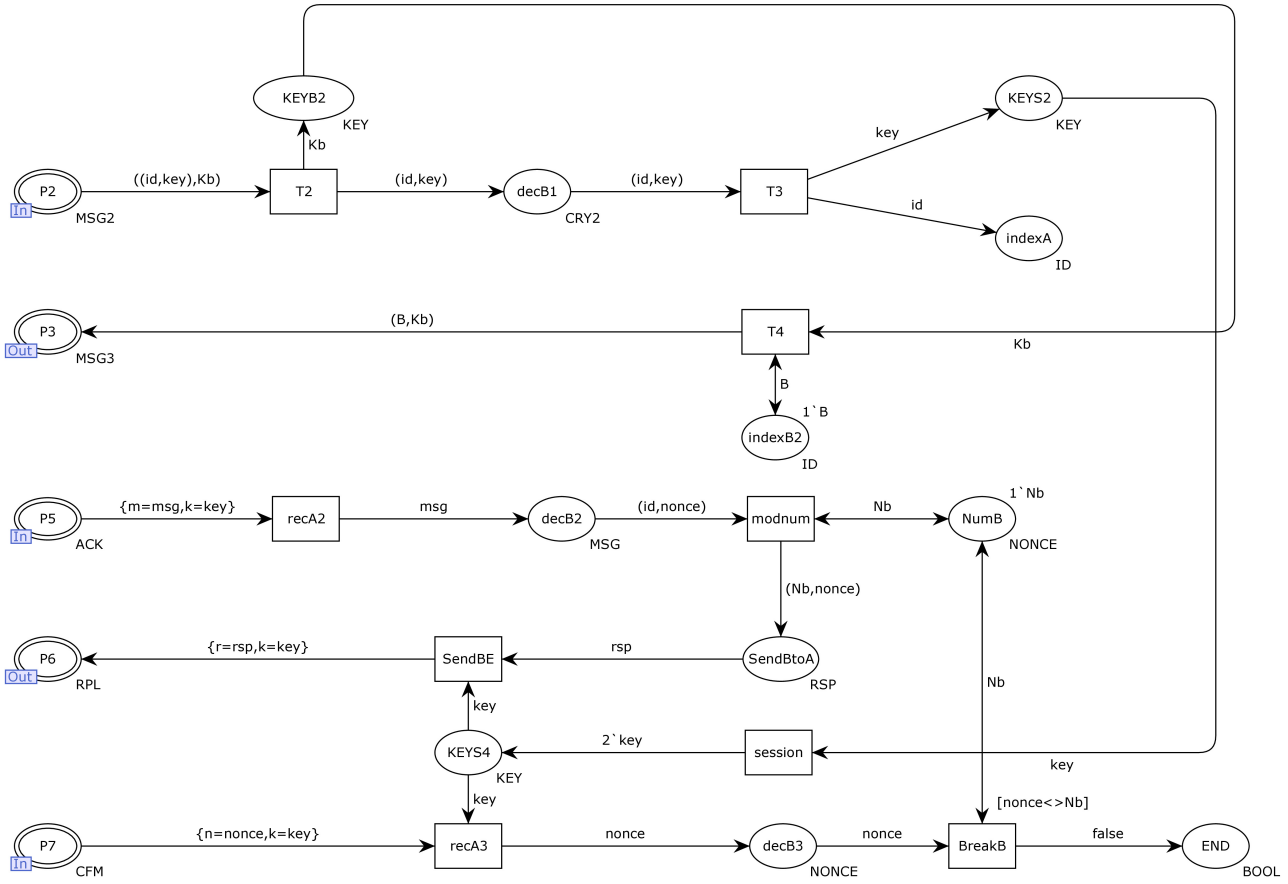


Figure 5: CPN model of entity B certified by BACnet/IP protocol equipment

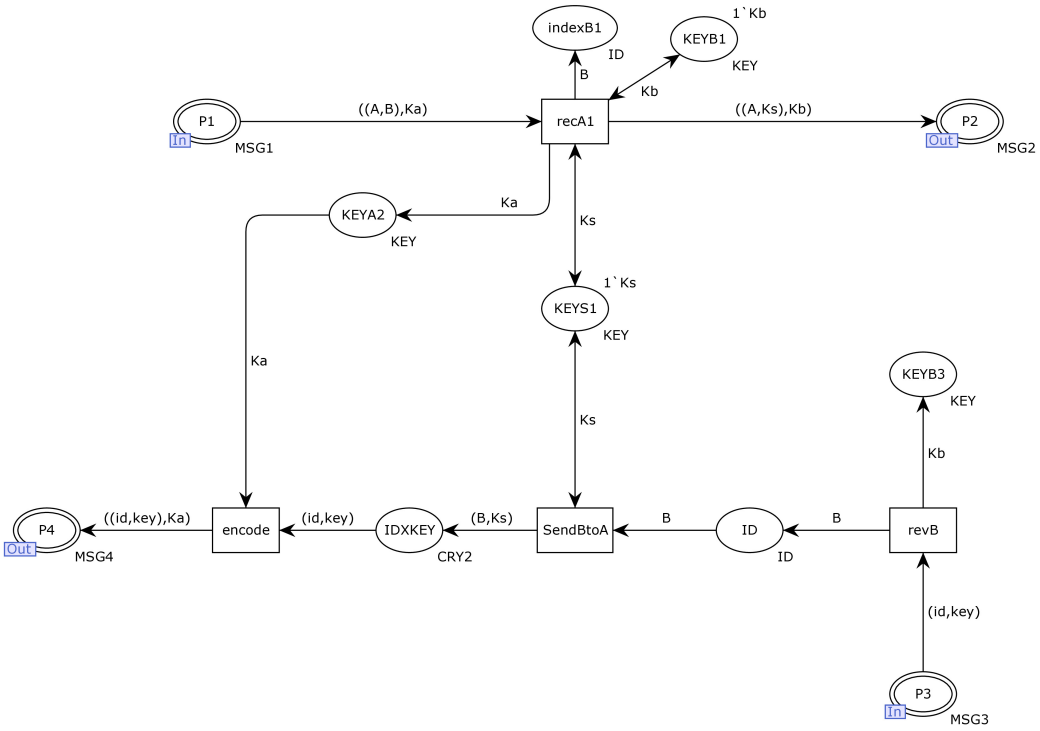


Figure 6: The CPN model of the key server for BACnet/IP protocol device authentication

Table 2: State space query results of HCPN model certified by BACnet/IP protocol equipment

Categorys	No.	Name
State space node	40	/
Directed arc	59	/
Strongly connected node	40	/
Strongly connected arc	59	/
Master state node	0	/
Dead node	1	[40]
Death transition	2	BreakA/BreakB
Live transition	0	/

following Table 2.

Perform state analysis on the CPN model, focusing on investigating the node statistics, bounded data, and liveness data, *etc.*, and then compare the survey results with the expected state to determine whether the model is consistent with expectations and whether it meets the protocol specifications. According to the BACnet/IP protocol device certification specification, when a device initiates an authentication request, the data packet will carry the generated pseudo-random number. Whether the device can be successfully authenticated depends on whether the modified random number in the response data packet meets expectations.When the device is successfully authenticated, device A will trigger the transitions decA2 and match instead of BreakA, and device B will trigger the transitions recA and decB3, but not the transition BreakB, so it can be predicted that the transitions BreakA and BreakB are two death transitions of the model. The model automatically ends after the certification request is completed. According to this feature, it can also be judged that there should be no live transitions when the model is in the terminated state.

4 Protocol Security Evaluation Based on Attacker Model

4.1 Dolev-Yao Attack Model

Research shows that attackers'attack methods include attacks on cryptographic algorithms, attacks on the protocol itself, and attacks on both cryptographic algorithms and the protocol itself.Under the premise of the perfect password assumption, the attacker can also launch passive or active attacks on the protocol itself. Passive attacks include eavesdropping and traffic analysis, but the attacker only passively detects the data stream transmitted in the network, and it is difficult to detect passive attacks on the protocol. Therefore, passive attacks can only collect effective information, prepare for active attacks, and will not conduct malicious attacks against the protocol. Active attacks on the protocol include interception attacks [27], replay attacks [13], integrity violation attacks [10], type error attacks, concurrent session attacks, and denial of service attacks [2], *etc.*

In 1983, Dolev and Yao published an important document in the history of security protocol development [5]. In this paper, Dolev and Yao proposed to distinguish the security protocol itself from the specific cryptographic algorithm used by the security protocol, and analyze the correctness, security and redundancy of the security protocol itself based on the assumption of a complete cryptographic system [8]. As a result, the analysis of security protocols is clearly divided into two different levels:First study the security nature of the security protocol itself, and then discuss the specific details of the implementation level and the specific cryptographic algorithm used. Dolev and Yao also established the corresponding attacker model to accurately describe the attacker's behavior:

1) Attackers can eavesdrop and intercept all messages passing through the network;

2) Attackers can store intercepted or self-constructed messages [6];

3) Attackers can send intercepted or self-constructed messages;

4) The attacker can participate in the operation of the protocol as a legal subject [24].

4.2 Protocol Security Evaluation Model Establishment

Since the Delov-Yao attacker model will generate a large number of repeated messages, it is easy to cause the state space to explode, which limits the use of the model to a certain extent, so this article adopts an improved Delov-Yao attacker model. On the one hand, the attack is applied to the arc expression in a parameterized form to reduce the state space;On the other hand, it restricts the messages that the attacker can split and combine, and only split and combine key messages that are effective to organize the attack into a disordered state to prevent the explosion of state space [19].

The security assessment model of the protocol security assessment model constructed based on the improved Delov-Yao attack model, as shown in Figure 7. According to Delov-Yao's attack hypothesis, the attacker has the ability to eavesdrop, tamper, and replay. It can pretend to be the initiator and responder of the session, but not a trusted third-party server. As shown in the figure, the blue-labeled arc expression of the transition AT simulates tampering attacks and replay attacks, and the transitions in the purple-labeled part and the library simulate spoofing attacks.Different definition places correspond to different color sets, and their functions are also different.For example, the color set of resovle is defined as DB, whose function is to store the intercepted information;The main function of the definition place CB1 is to store and split the original message.Different types of transitions play

Table 3: State space query results of BACnet/IP protocol equipment certification security evaluation model

Categorys	Numbers
State space node	248
Directed arc	495
Strongly connected node	248
Strongly connected arc	495
Dead node	3
Death transition	2

different roles. For example, the transition CB saves the decrypted message and key to the places CB1 and CB2 respectively. In the process of changing AB to synthesize messages, it is also necessary to introduce concurrency control place SP to restrict and regulate.

4.3 Analysis of State Space Table of Safety Assessment Model

When the three attacks are launched at the same time, the corresponding state space report of the security assessment model is shown in Table 3. The static characteristics of the system are represented by statistical data, which have two attributes, namely: state space attributes and strongly connected graph attributes. The model includes a variety of elements, such as node identifiers, arc connecting nodes, the former is represented by nodes. The result of the state is full, which means that the corresponding appearance graph is complete. The space state table provides a variety of operating tools. By performing Home Properties and Live Properties operations, you can see the model's identification name, live transition name, *etc.*Through the above method, the state space query result can be obtained.

4.4 Protocol Security Analysis

The number of arcs of the security evaluation model has increased significantly compared with the original model, which meets the research needs. In addition, because the number of arcs and the number of nodes are the same, it can be seen that there is no iterative behavior in the security evaluation model, and the added Delov-Yao attack is also verified. The author model is effective. It can be seen from the report that the model generates a total of 248 state space nodes, including 3 dead nodes, which shows that unexpected behaviors have occurred in the security assessment.Use ListDeadMarking() to determine the serial numbers of 3 dead nodes. Through the NodeDescriptor() function to check the status of all dead nodes, it is found that the attacker at node 236 has successfully forged a legitimate participant to tamper with the session key between devices A and B to Kt and initiate a replay attack;Node 244 is caused by an unexpected final state of the protocol due to a spoofing attack.

5 Protocol Improvement and Analysis

5.1 A New BACnet/IP Protocol Equipment Authentication Security Scheme Based on HCPN Modeling

Through the evaluation test of the security evaluation model, it can be seen that the BACnet/IP protocol device authentication mechanism does not meet the design requirements of device authentication in the protocol specification, and does not have the ability to resist tampering, replay and other attacks. In response to the above-mentioned security threats, this article proposes a protocol improvement scheme that introduces time stamps to strengthen the device authentication strength, which specifically includes security improvements in the key distribution phase and device authentication phase, and uses the security evaluation model again to improve the security of the protocol. Verified. The improved authentication message flow (MSC) model is shown in Figure 8. Timestamp adds time stamp information for the improved protocol.

The improved equipment certification process is as follows:

1) Devices A and B run the DES algorithm to generate their own private keys. Device A has a key Ka, and device B has a key Kb (only shared with the key server).

2) Device A sends a "RequestKey" request to the key server, requesting a session key to ensure the security of the logical connection to device B. The message contains the transmitted time stamp information and the identity of device A and device B.

3) After receiving the message from device A, the key server Server uses Ka to perform the data source identification process, determines whether the request is issued by device A, and then determines whether the timestamp meets the security requirements. If not, discard the data packet to stop the authentication process, and continue to use the DES algorithm to generate the session key Ks and use the master key Kb of device B to encrypt the session key Ks, IDA and time stamp information, and send it to device B.

4) After device B receives the message sent by the key server Server, it uses Kb to perform the data source identification process to determine whether it is sent by the key server Server and whether the timestamp meets security standards.Device B decrypts and stores the session key Ks to be used later, and uses Kb to encrypt the device number IDB and time information and send it to the key server. The key server verifies the device number of device B.

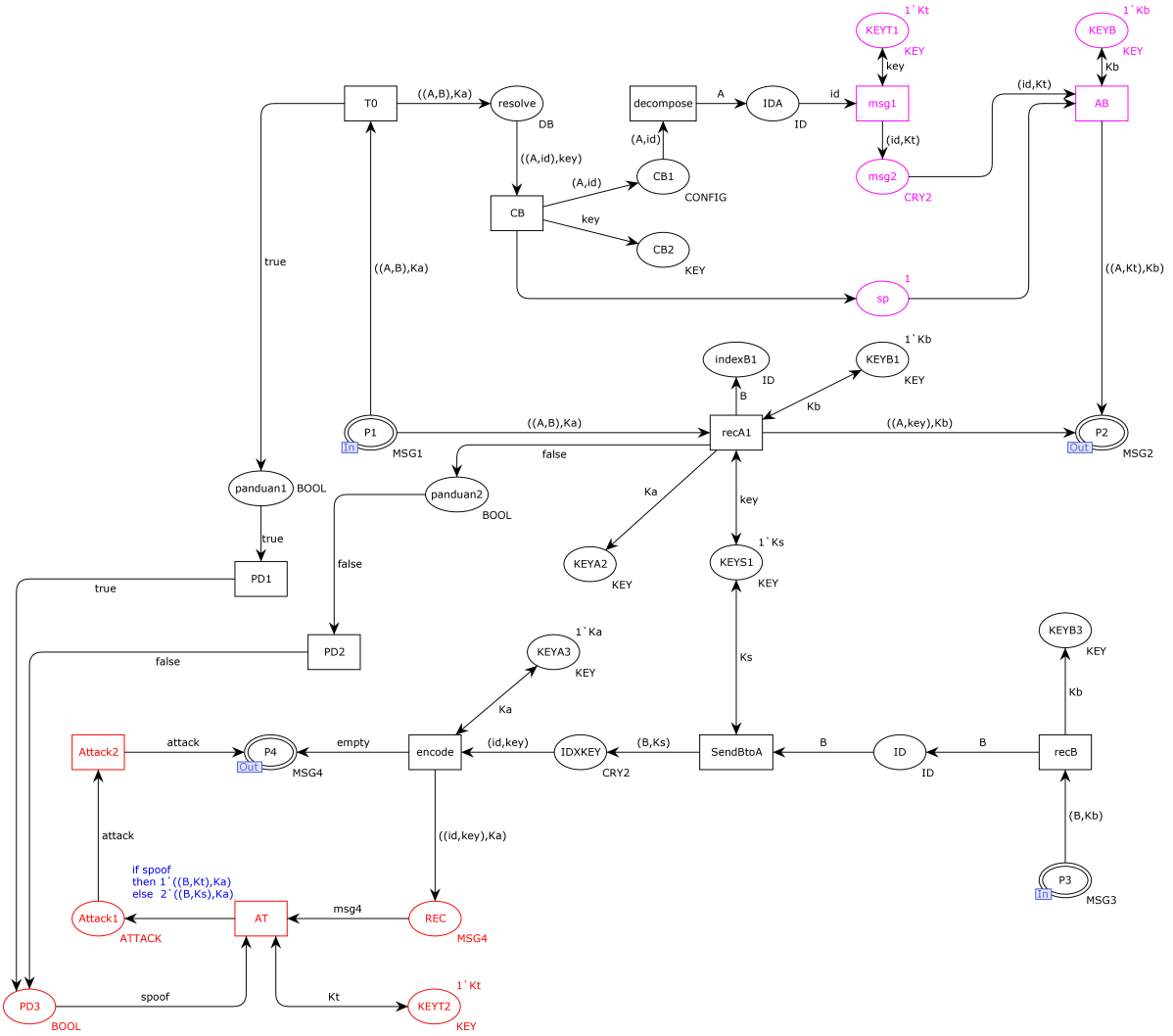


Figure 7: The security assessment model of BACnet/IP protocol equipment certification

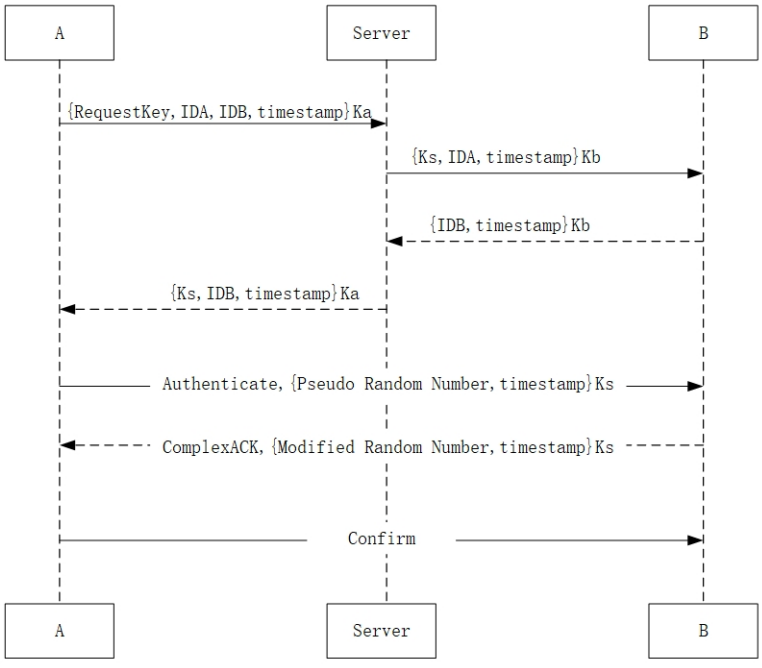


Figure 8: Improved authentication mode message flow model (MSC)

5) After the key server uses Kb for identification and decryption again, it first compares the obtained IDB with the one sent by device A. If they are the same, it uses the Ka of device A to encrypt Ks, IDB and the time stamp information, and sends it to device A.

6) Device A receives Ks and starts to identify device B. Device A generates an Authenticate service request. The protocol data part, Pseudo Random Number and time information of this request are encrypted with the session key Ks and sent to device B.

7) Device B decodes the received authentication request of device A, changes the Pseudo Random Number to Modified Random Number, uses Ks to encrypt, and returns a ComplexACK message to device A.

8) Device A decodes the received response message and checks whether the ComplexACK packet contains the correct "Modified Random Number". If it is correct, the device is successfully authenticated and can start communication.

5.2 Improved CPN Physical Layer Model for Equipment Certification

5.2.1 Improved CPN Model of Equipment Certification Entity A

Based on the modeling of the original entity A, a new color set T representing time stamp information and two constant definitions representing time delay are added. The constant Prdelay represents the operation delay of the places in the model to the data, and the constant Trdelay represents the maximum possible time for normal message transmission in the BACnet network. If this limit is exceeded, the message is lost or the system has been invaded. In order to fully verify the relevant security properties, the message recipient should check the time value before decrypting the message for processing. Figure 9 shows the improved CPN model of device authentication entity A.

5.2.2 Improved CPN Model of Device Authentication Entity B

Figure 10 shows the improved CPN model of the device authentication entity B. Its behavior includes receiving the session key Ks sent by the key server Server, responding to the device A with a modified random number for identity verification, and verifying the time stamp information and calculations. The new place Timestamp2 is used to store and calculate the current timestamp information of the device.

Table 4: Comparison of state space of security evaluation model before and after BACnet/IP protocol equipment certification improvement

Categorys	Before	After
State space node	248	1475
Directed arc	495	4193
Strongly connected node	248	1475
Strongly connected arc	495	4193
Dead node	2	5
Death transition	2	2

5.2.3 Improved CPN Model of Device Authentication Key Server Server

The behavior of the key server Server mainly includes identifying and decrypting messages from device A, and obtaining the device numbers of devices A and B, and then encrypting the session key Ks in two messages and distributing them to devices A and B. In the CPN model of the key server with the attacker, two Timestamp places are also added to store and calculate timestamp information, which enriches the attacker's ability to split combined messages. Figure 11 shows the improved BACnet/IP protocol device authentication key server server CPN model.

5.3 State Space Analysis of Improved CPN Model

Table 4 shows the comparison of the state space results of the improved BACnet/IP protocol equipment certification security evaluation model with that before the improvement. Due to the introduction of timestamps and the addition of related color definitions, the model correspondingly increases the number of transitions and places, and the number of states and directed arcs after the improvement are significantly increased compared to before the improvement.

In the security evaluation phase, add attack parameters to the arc expression to verify whether the improved BACnet/IP protocol device authentication can resist both tampering and replay attacks. The SML statement investigation of the dead state in the above state space report shows that at node 35, because the attacker's decomposition and synthesis message time exceeds the threshold, device B discards data packets, making subsequent attacks invalid, and nodes 1441, 1452, and 1474 are sent to ports. The message MSG4 of the library P4 was intercepted and replayed by the attacker, which caused the system token to be exhausted and the final authentication could not be completed. The improved BACnet/IP protocol equipment authentication can resist information tampering and message replay attacks, and meet the authentication attribute requirements defined by the BACnet/IP protocol equipment authentication specification.

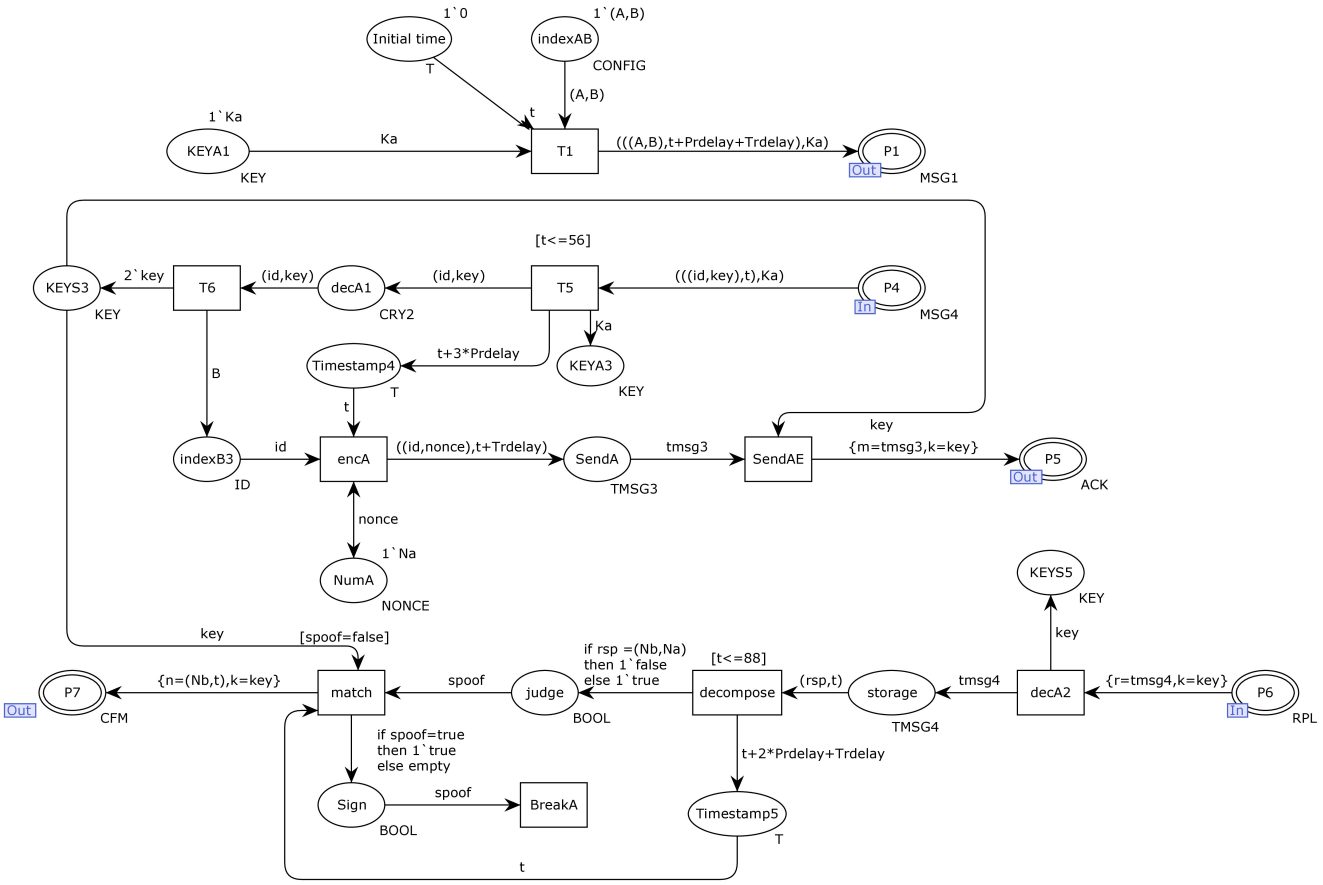


Figure 9: The improved BACnet/IP protocol device authentication entity A's CPN model

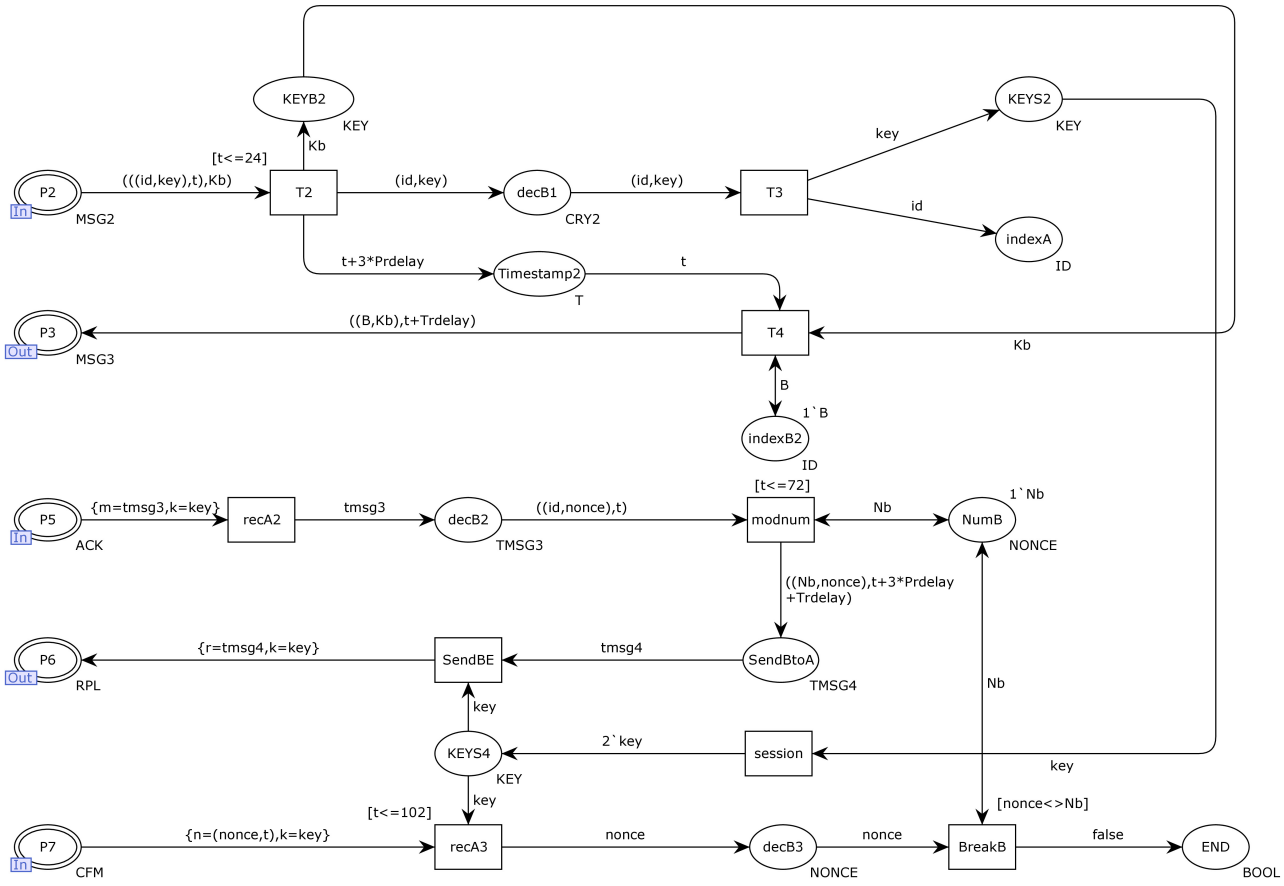


Figure 10: CPN model of entity B for improved BACnet/IP protocol equipment authentication

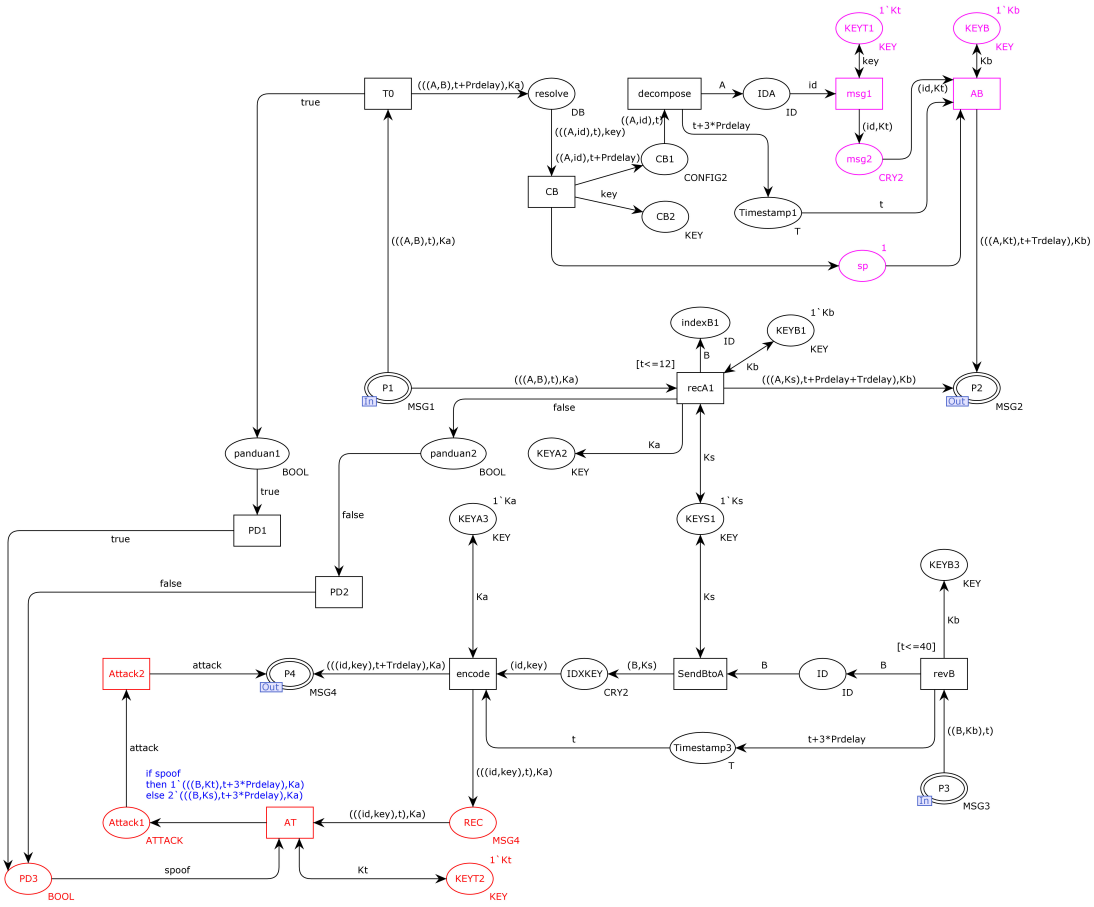


Figure 11: The improved BACnet/IP protocol device authentication key distributor Server CPN model

6 Conclusion

This paper, guided by colored Petri net theory and the Delov-Yao attack method, takes the building automation network communication protocol, BACnet/IP equipment certification, as the research object and uses the CPN Tools model detection tool to formalize the BACnet/IP protocol modeling and safety assessment. By modeling and analyzing the equipment authentication mechanism of the protocol, mining and verifying security vulnerabilities, a security improvement plan that uses timestamps to enhance the authentication strength between devices is proposed. The security of the proposed scheme is verified using CPN model detection tools. The next step is to improve the fine-grained protocol modeling based on the current research and verify the security of other protocol services and other forms of attacks.

References

[1] M. Abbaszadeh, S. Saeedvand, "Weak consistency model in distributed systems using hierarchical colored petri net," *Journal of Computers*, vol. 13, no. 2, pp. 236-243, 2018.

[2] L. An, G. H. Yang, "Decentralized adaptive fuzzy secure control for nonlinear uncertain interconnected systems against intermittent DoS attacks," *IEEE Transactions on Cybernetics*, no. 99, pp. 1-12, 2018.

[3] D. Arena, F. Criscione, N. Trapani, "Risk assessment in a chemical plant with a CPN-HAZOP tool," *IFAC-Papers OnLine*, vol. 51, no. 11, pp. 939-944, 2018.

[4] I. V. Artamonov, A. P. Sukhodolov, "CPN tools-based software solution for reliability an alysis of processes in microservice environments," *International Journal of Simulation: Systems,Scienceand Technology*, vol. 19, no. 6, 2018.

[5] D. Dolev, A. Yao, "On the security of public key protocols," *IEEE Trans on Information Theory*, vol. 29, no. 2, pp. 198-208, 1983.

[6] Y. Fang, "Research on automatic verification method of security protocol based on model checking,"Hunan University,2015.

[7] O. Gasser, Q. Scheitle, C. Denis, *et al.*, "Security implications of publicly reachable building automation systems," *IEEE Security and Privacy Workshops (SPW'17)*, 2017. DOI: 10.1109/SPW.2017.13.

[8] J. A. Herzog, "Computational interpretation of Dolev–Yao adversaries," *Theoretical Computer Science*, vol. 340, no. 1, pp. 57-81, 2005.

[9] T. Hong, "Research on the security problems and countermeasures of BACnet network," Chongqing University,2006.

[10] Information Technology - Data Integrity; Reports on Data Integrity from North China Electric Power University Provide New Insights (Data Integrity

Attack Detection for Node Voltage In Cyber-physical Power System)[J]. Information Technology Newsweekly,2020.

[11] W. Kastner, G. Neugschwandtner, S. Soucek, "Communication systems for building automation and control," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1178-1203, 2005.

[12] J. Kaur, J. Tonejc, S. Wendzel, "Securing BACnet's pitfalls," in *The 30th International Information Security and Privacy Conference*, pp. 616-629, 2016.

[13] J. Kim, J. S. Song, "A simple and efficient replay attack prevention scheme for LoRaWAN," in *The 7th International Conference*, pp. 32-36, 2017.

[14] L. Li, F. Basile, Z. Li, "An approach to improve permissiveness of supervisors for GMECs in time petri net systems," *IEEE Transactions on Automatic Control*, no. 99, pp. 1-1, 2019.

[15] K. Lohia, Y. Jain, C. Patel, "Open communication protocols for building automation systems," *Procedia Computer Science*, vol. 160, pp. 723-727, 2019.

[16] M. Mylrea, S. N. G. Gourisetti, "Cybersecurity and optimization in smart "Autonomous" buildings," in *Autonomy and Artificial Intelligence: A Threat or Savior?*, pp. 263-294, 2017.

[17] M. Nast, B. Butzin, F. Golatowski, *et al.*, "Performance analysis of a secured BACnet/IP network," *The 15th IEEE International Workshop on Factory Communication Systems (WFCS'19)*, 2019. DOI: 10.1109/WFCS.2019.8758009.

[18] Z. Pan, S. Hariri, J. Pacheco, "Context aware intrusion detection for building automation systems," *Computers & Security*, vol. 85, pp. 181-201, 2019.

[19] M. Peacock, M. N. Johnstone, C. Valli, "An exploration of some security issues within the BACnet protocol," in *Information Systems Security and Privacy*, pp. 252-272, 2018.

[20] Z. Qing, D. Shiyun, "Building equipment management system based on BACnet protocol," *Intelligent Building*, no. 06, pp. 58-60, 2020.

[21] A. G. Siemens Schweiz, Patent Issued for System And Method For Isolating Device Communications In A BACnet/IP Building Automation Network (USPTO 10,812,287)[J]. Internet Weekly News,2020.

[22] M. Simon, D. Moldt, D. Schmitz, *et al.*, "Tools for curry-coloured petri nets," in *Application and Theory of Petri Nets and Concurrency*, pp. 101-110, 2019.

[23] J. Xiaoyan, "Formal security assessment and improvement of BACnet protocol based on HCPN model checking method," Lanzhou University of Technology,2020.

[24] W. Xiong, L. Robert, "Threat modeling–A systematic literature review," *Computers & Security*, vol. 84, pp. 53-69, 2019.

[25] T. Xuecheng, "Security analysis of EtherNet/IP protocol in industrial control system," Lanzhou University of Technology,2020.

[26] J. Yuzheng, "Application of PKI technology in information security protection of military intelligent buildings," Shan Dong University,2006.

[27] H. Zhou, W. Yang, C. Yang, "Privacy preserving consensus under interception attacks," *The 36th Chinese Control Conference (CCCC'17)*, 2017. DOI: 10.23919/ChiCC.2017.8028702.

Acknowledgments

This research is supported by The National Natural Science Foundation of China (Grant No. 61762060), Educational Commission of Gansu Province, China (Grant No.2017C-05), Foundation for the Key Research and Development Program of Gansu Province, China (Grant No.20YF3GA016). Tao Feng is the corresponding author.

Biography

Feng Tao, was born in 1970, researcher/PhD supervisor, CCF senior member, IEEE and ACM member.He graduated from Xidian University,and then worked at school of computer and communication in Lanzhou University of Technology. His research interests include Cryptography and information security.

Zhao Si-Meng, was born in 1996, CCF member.He is a master's student at lanzhou university of technology.His research interests include technical information security and industrial control systems.

Gong Xiang, was born in 1986, CCF member. He is a doctor's student at lanzhou university of technology. His research interests include technical information security and industrial control systems.

General Certificate-based Key-Exposure Resilient Provable Data Possession without Certifier

Feng Wang[1,2], Li Xu[2], Zhide Chen[2], and Qikui Xu[3]
(Corresponding author: Feng Wang)

School of Computer Science and Mathematics, Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology[1]
Fuzhou, Fujian 350118, China
Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University[2]
Fuzhou, Fujian 350007, China
Fudan University[3]
Shanghai 200433, China
(Email: w.h.feng@163.com)
(Received Feb. 7, 2021; Revised and Accepted Nov. 17, 2021; First Online Feb. 26, 2022)

Abstract

With the rapid development of the internet of things and cloud storage, more and more industrial big data have been outsourced to cloud servers. Therefore, protecting the integrity of data outsourced in the cloud becomes a hot issue. Provable data possession (PDP) is a popular method for protecting the integrity of cloud-stored data. However, all the existing public-key PDPs need a trusted third party, and there is little attention paid to certificate-based PDPs. Therefore, we proposed a basic certificate-based PDP system and security model. Furthermore, we removed the certifier by combining the features of certificate-based cryptology with provable data possession. We made the cloud server do the duties of the certifier to simplify key management. The general construction of this improved version of certificate-based PDP, together with the secure proof, was also presented in this paper. Our proposed scheme not only simplifies the management of users but also possesses some other properties, such as non-reputability, key-exposure resistance, *etc.*

Keywords: Certificate-based; Cloud Storage; Key-Exposure Resilience; Provable Data Possession; Without Certifier

1 Introduction

With the rapid development of the internet of things, especially for the era of Industrial 4.0, more and more industrial data need to be stored. Since cloud storage is avail and reliable at a relatively low cost [9], more and more individuals and organizations intend to outsource their data into cloud server. However, losing the control of data in the cloud storage can induce some security and privacy risks. For example, the cloud service provider may hide the event of data loss for the sake of reputation. Therefore, the integrity of the outsourced data is one of the users' main concerns.

In order to protect the integrity of the outsourcing data, Ateniese *et al.* [3] proposed the provable data possession (PDP) model in 2007. In their model, the auditor can audit the integrity of outsourced data with a high probability without fetching it back. Proof of retrievability (POR) model [4, 16] is a variation of PDP model. In POR model, if the outsourced data passed the auditor's integrity audit, the user can extract the data.

Following Ateniese *et al.*'s work, many PDP schemes have been proposed for different application scenarios. Dynamic PDP schemes [5, 23, 26, 30] intend to audit the integrity of dynamically refreshed outsourced data; non-repudiable PDP schemes [23,24] intend to protect the benefits of cloud service provider from the users' misbehavior; PDP with key-exposure resistance schemes [27,28] intend to solve the key-exposure problem of PDP schemes; identity-based PDP schemes [14, 17, 20] intend to eliminate the complicated certificate management of the traditional public key PDP schemes, and Li *et al.* [13] extended identity-based PDP scheme to fuzzy identity-based PDP scheme; certificateless PDP schemes [7,8,11,19,29,30] and certificate-based schemes [18, 21] intend to solve the key escrow problem of identity-based PDP schemes; *etc.*

In identity-based PDP scheme [20], the authors considered company-oriented cloud storage. When a company purchases the cloud storage service, only the staff members of company are allowed to upload data to the

cloud server. Therefore, in order to convince that the data are coming from the staff members of the company, the cloud server must check the users' certificate of public key, which will cost a large amount of computational power. In order to eliminate the complicated certificate management, the identity-based PDP schemes were proposed.

Although identity-based cryptology simplifies the certificate management of traditional public key cryptography, it brings the key escrow problem [1]. In order to solve this problem, Al-Riyami and Paterson [2] proposed the notion of certificateless cryptography, and Gentry [6] proposed the notion of certificate-based cryptography. In certificateless cryptography, the user generates the secret value behind closed doors, and the key generation center (KGC) generates the partial secret key from the user's identity. The user combines the secret value and partial secret key as his secret key, and generates and publishes his public key without certificate. Following Al-Riyami and Paterson's work, many certificateless cryptography schemes [10, 15] were proposed.

Combining certificateless cryptography with PDP scheme, Wang *et al.* [19] proposed the certificateless PDP scheme in 2013, He *et al.* [8] pointed out that Wang *et al.*'s scheme was vulnerable to public key replacement adversary attack, and proposed an improved scheme. Kim and Jeong [11] proposed another certificateless PDP scheme. Unfortunately, we find that both Wang *et al.*'s scheme [19] and Kim and Jeong's scheme [11] are vulnerable to malicious KGC attack, which is similar to the malicious KGC attack described in [22]. Zhang *et al.* [29] proposed a certificateless PDP scheme against malicious auditors in 2016. He *et al.* [7] proposed a certificateless PDP scheme for smart grid in 2018. Zhou *et al.* [30] proposed a certificateless integrity auditing scheme in 2020, which supports multi-copy storage and dynamic data updates.

In certificate-based cryptography [6, 12, 25], each user generates his partial secret key and the corresponding public key, then requests a certificate from the certifier by using his identity and public key, and forms his secret key from the partial secret key and the certificate.

However, there is little attention paid to certificate-based PDP schemes. To date, only two proposed schemes [18, 21] fall into this category, which focus on private auditing and lightweight auditing respectively. Therefore, we proposed a basic certificate-based PDP. Furthermore, all the existing public-key PDPs need a trusted third party. In order to further simplify the key management of PDP, we combined the features of certificate-based cryptography and PDP scheme, and proposed an improved certificate-based PDP by using the cloud server undertaking the duty of certifier. The main contributions are described below.

First, the basic certificate-based PDP is introduced, which is rarely paid attention to. Then, the security model of it is given.

Second, in order to further simplify the key management of PDP, the certifier is moved, and the cloud server is made to do the duties of certifier. This results in an improved certificate-based PDP with the security model of it given.

Third, based on the hypothesis that the secure basic certificate-based PDP scheme has existed, a general construction of the improved certificate-based PDP is proposed, which is proved to be secure.

Finally, our improved certificate-based PDP not only simplifies the management of users, but also harbors some other properties, such as non-repudiability, key-exposure resistance, *etc.*

The rest of the paper is organized as follows. Section 2 introduces the preliminaries, including public key cryptography, certificate-based signature, provable data possession (PDP) model. The basic certificate-based PDP model is proposed in Section 3 and the improved certificate-based PDP model in Section 4. The secure analysis of the improved certificate-based PDP scheme is given in Section 5. Our proposed schemes were compared with others in Section 6, with the conclusion of our manuscript outlined in Section 7.

2 Preliminaries

In this section, a review of certificate-based signature [6, 12], and provable data possession (PDP) model [3, 5, 13, 19, 20] will by given.

2.1 Certificate-based Signature

In certificate-based cryptography [6, 12], each user generates his partial secret key and the corresponding public key, then requests a certificate from the certifier. Unlike traditional public key cryptography, the user forms his secret key from the partial secret key and the certificate. There are three entities involved in a certificate-based signature, *i.e.*, the signer, the verifier, and the certifier. The certifier generates its system public parameters and master secret key. The signer generates his public key pk and partial secret key. Then, given the signer's identity ID and pk, the certifier generates the signer's certificate and sends it to the signer via a secret channel. The signer generates his secret key sk by using his partial secret key and certificate. Given a message m, the signer uses a signature generation algorithm $CBSign(\cdot,\cdot)$ to generate signature, *i.e.*, the signature $\sigma = CBSign(sk, m)$. Given the pk, m and σ, the verifier checks the validity of σ by using the signature verification algorithm $CBVeri(\cdot,\cdot,\cdot,\cdot)$, *i.e.*, checks whether or not the equation $CBVeri(ID, pk, m, \sigma) = 0/1$ holds. If it holds, the signature is valid and outputs 1; otherwise, it is invalid and outputs 0.

Generally, there are two kinds of adversaries in certificate-based signature. One is the public key replacement adversary, who can select the user's identities and replace the public keys at his will, but cannot have access to the master secret key. The other is malicious certifier

adversary, who can have access to the master secret key and try to impersonate the user, but cannot replace the public key.

2.2 Provable Data Possession Model

Provable data possession (PDP) [3,5,13,19,20] provides a method by which the user can verify the integrity of the outsourced data without retrieving it. Generally, there are three entities and five algorithms involved in it. The three entities are the cloud server, user, and auditor. The cloud server is an entity managed by the cloud server provider. It has generous storage space and computational resources to maintain the users' data. The user is a resource-constrained consumer who has a lot of data files and wants to store his data into cloud server for data maintenance and computation. The auditor is a trusted third-party provides data auditing service for users with cloud servers. Sometimes, it also can be user himself.

The five algorithms are KeyGen, TagGen, Chall, ProofGen, ProofVerify. The KeyGen algorithm is performed by the user, and generates the user's secret/public key pair. The TagGen algorithm is performed by the user. Given the secret key and the file, it generates the homomorphic verifiable tags. The Chall algorithm is performed by the auditor (or user). Given the abstract information of a file, it generates the challenge of it. The ProofGen algorithm is performed by the cloud server. Given the challenge, it generates the response of it. The ProofVerify algorithm is performed by the auditor (or user). Given the response, it verifies the validity of the response.

A PDP scheme is secure means that if a cheating prover produces the response for a challenge, and passed the ProofVerify algorithm, he must have stored the challenged file. That is to say, there is no adversary who can generate a valid response without storing the file.

3 The Basic Certificate-based PDP

In this section, following the ideas of (fuzzy) ID-based PDP model and certificateless PDP model, the system model and the security model of the basic certificate-based PDP are introduced.

3.1 The General Construction of Basic Certificate-based PDP

There are four entities in the basic certificate-based PDP model, *i.e.*, the certifier, cloud server, user and auditor. The certifier is an independent trusted third party, and has the tasks of generating system parameters and certificate of user from user's identity and public key. The other three entities are the same as those described in Subsection 2.2.

There are seven algorithms, *i.e.*, BSetup, BUserKeyGen, BCertify, BTagGen, BChall, BProofGen, BProofVerify in the basic certificate-based PDP model. The relationship of the entities and algorithms is described as bellow, and it can also be seen from Figure 1.

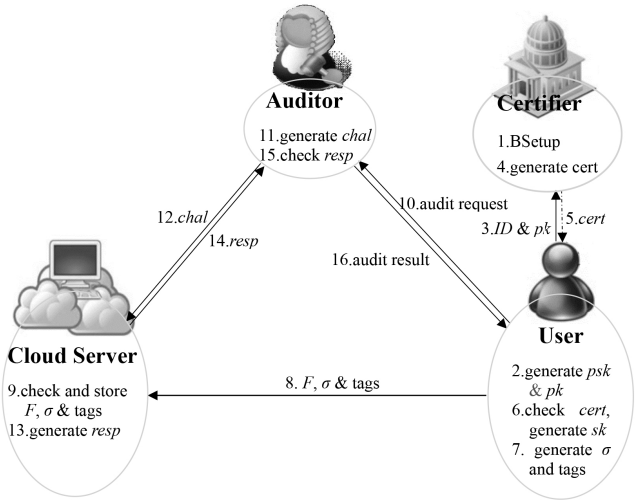


Figure 1: The structure of the basic certificate-based PDP

The **BSetup** algorithm (see Step 1 of Figure 1). With the security parameter λ, the certifier generates the system public parameters spp and the system master secret key $smsk$. Let $CBSign(\cdot,\cdot)$ and $CBVeri(\cdot,\cdot,\cdot,\cdot)$ be the certificate-based sign and verify algorithms pair described in Subsection 2.1. The certifier makes spp, $CBSign(\cdot,\cdot)$ and $CBVeri(\cdot,\cdot,\cdot,\cdot)$ public, and keeps the system master secret key $smsk$ secret. This algorithm is denoted by $(spp, smsk, CBSign, CBVeri) = BSetup(\lambda)$.

The **BUserKeyGen** algorithm (see Steps 2-3 of Figure 1). With the system public parameters spp, the user generates his partial secret key psk and public key pk, sends his public key pk, and identity ID to certifier and keeps his partial secret key psk secret. This algorithm is denoted by $(psk, pk) = BUserKeyGen(spp)$.

The **BCertify** algorithm (see Steps 4-6 of Figure 1). There are two steps in this algorithm, the certifier performs Step 1, and the user performs Step 2.

Step 1. With the user's ID and public key pk, the certifier generates the certificate $cert$, and sends it to the user via a private channel. This algorithm is denoted by $cert = BCertify(spp, smsk, ID||pk)$.

Step 2. Upon receiving $cert$, the user checks the validity of $cert$, and generates his secret key sk by using his partial secret key psk and certificate $cert$.

The **BTagGen** algorithm (see Steps 7-9 of Figure 1). There are three steps in this algorithm, the user performs Steps 1-2, and the cloud server performs Step 3.

Step 1. With a file F, the user picks a random file name $filename$, splits the file F into n blocks, *i.e.*, $F = (F_1, F_2, \ldots, F_n)$, computes signature $\sigma = CLSign(sk, filename||n)$, the homomorphic verifiable tags $\tau_i (i = 1, 2, \ldots, n)$ by using his secret key

sk. Then, the user uploads the file F, the signature σ and the tags $\tau_i(i = 1, 2, \ldots, n)$ to cloud server. This algorithm is denoted by $(\tau_i(i = 1, 2, \ldots, n), \sigma) = BTagGen(F, spp, ID, pk, sk)$.

Step 2. Upon receiving F, σ and $\tau_i(i = 1, 2, \ldots, n)$, the cloud server checks the validity of the signature σ and $\tau_i(i = 1, 2, \ldots, n)$. If one of them is invalid, stops, otherwise, stores F, σ and $\tau_i(i = 1, 2, \ldots, n)$.

Step 3. If the upload is successful, the user deletes the file F, the signature σ and the tags $\tau_i(i = 1, 2, \ldots, n)$ on local storage.

The **BChall** algorithm (see Steps 10-12 of Figure 1). Upon receiving the block number b_n from the user, the auditor authorized by the user randomly selects a subset L in set $\{1, 2, \cdots, n\}$ with b_n elements, and corresponding random numbers $\{v_l\}_{l \in L}$, generates the challenge $chal = (L, \{v_l\}_{l \in L})$, and sends it to the cloud server. This algorithm is denoted by $chal = BChall(b_n)$.

The **BProofGen** algorithm (see Steps 13-14 of Figure 1). Upon receiving the challenge $chal$, the cloud server generates the corresponding response $resp$, and returns it to the auditor. This algorithm is denoted by $resp = BProofGen(chal)$.

The **BProofVerify** algorithm (see Steps 15-16 of Figure 1). Upon receiving the response $resp$ form the cloud sever, the auditor checks the validity of the response $resp$. If it is valid, the file F isn't changed; otherwise, the file F is changed. Then the auditor sends the audit result to the user. This algorithm is denoted by $0/1 = BProofVerify(resp)$, where $0/1$ denotes the file is changed/unchanged.

3.2 The Security Model of Basic Certificate-based PDP

Inspired by Subsections 2.1, 2.2, there are two adversaries of polynomial time in the model of basic certificate-based PDP, *i.e.*, public key replacement adversary and malicious certifier adversary.

The public key replacement adversary can select user's identities and replace the public keys at his will, but cannot have access to master secret key. Therefore, the certifier cannot play the part of public key replacement adversary because he knows the master secret key.

The malicious certifier adversary can have access to the master secret key and try to impersonate the user, but cannot replace the public key.

If a basic certificate-based PDP scheme can resist the attacks of both public key replacement adversary and malicious certifier adversary, the scheme is secure.

4 The Improved Certificate-based PDP

Note that a trusted third part certifier can complicate the scheme. For example, different users may want to use different certifiers, the cloud server will check the validity all the certifiers, and store the public parameters of them. Furthermore, in certificate-based signature, everyone can verify the signature. However, from Section 3 we can see, every signature and tag must be verified by the cloud server in the certificate-based PDP. Therefore, certificate-based PDP is much different from certificate-based signature.

It is worth notifying that the user can select his own public key at his will in certificateless signature. However, it is much different in certificate-based signature [25]. If one can replace someone's public key pk with $pk^{\#}$, he must have the certificate generated from ID and $pk^{\#}$. This can prove that the trusted third-party certifier is dishonest.

Combining this feature of certificate-based signature with basic certificate-based PDP model, an improved certificate-based PDP model was proposed. In the improved certificate-based PDP model, the certifier is removed and the cloud server is made to do the duties of the certifier.

Based on the basic certificate-based PDP scheme described in Section 3, a general construction of improved certificate-based PDP is given in Subsection 4.1, and the security model of the improved certificate-based PDP is given in Subsection 4.2.

4.1 The General Construction of the Improved Certificate-based PDP

In the improved certificate-based PDP, there is no certifier, and the cloud server performs the duties of the certifier. Therefore, only three entities, the cloud server, user, and auditor are involved in the system model of improved certificate-based PDP. The user and auditor have the same task as they have in the basic certificate-based PDP. However, the cloud server not only undertakes the duties of itself but also those of the certifier in the basic certificate-based PDP model.

Therefore, the cloud server has more power than it has in the basic certificate-based PDP model. Thus, the cloud server's behavior must be limitted against its superpower. A signature of the user's public key pk and identity ID signed by the cloud server in the ICertify (BCertify in Subsection 3.1) phase can afford this responsibility. It is noticeable that this signature does not need to be verified except in ICertify phase. The reasons will be described in Theorem 3 in Section 5.

Furthermore, in order to simplify the key management, a time period, tp, is introduced as the active time period of the user's key. For example, tp may be the service time period that the user has purchased from the cloud server.

Similarly, there are eight algorithms, *i.e.*, ISetup, IUserKeyGen, ICertify, ITagGen, IChall, IProofGen, IProofVerify, and ICertifyRenew in the system model of the improved certificate-based PDP, which can be seen from Figure 2. The IUserKeyGen (see Step 1 of Figure 2), IChall (see Steps 12-14 of Figure 2), IProofGen (see Steps 15-16 of Figure 2), and IProofVerify (see Steps 17-18 of Figure 2) algorithms are the same as the corresponding algorithms of the system model of the basic certificate-based PDP model in Subsection 3.1. We describe the ISetup, ICertify, ITagGen, and ICertifyRenew algorithm as following.

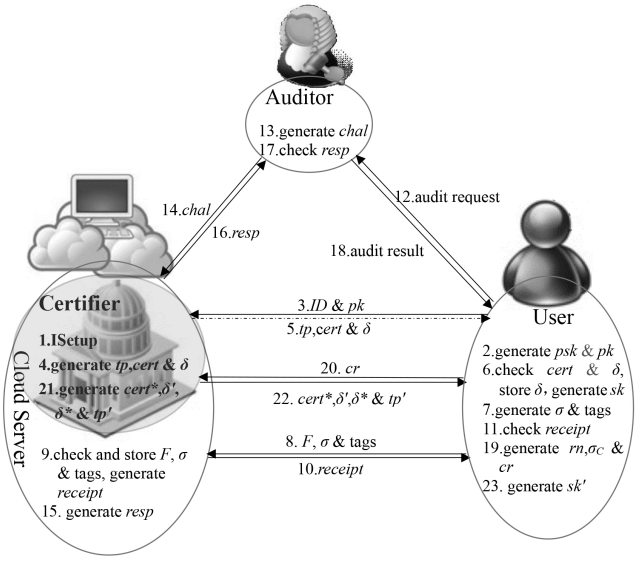


Figure 2: The structure of the improved certificate-based PDP

The **ISetup** algorithm (see Step 1 of Figure 2). With the security parameter λ, the cloud server (or certifier which is included in the cloud server) computes $(spp, smsk, CBSign, CBVeri) = BSetup(\lambda)$, picks a public and secret key pair pk_S and sk_S of itself, a traditional public key sign and verify algorithm pair $Sign(\cdot,\cdot)$ and $Veri(\cdot,\cdot,\cdot)$, and a encrypt and decrypt algorithm pair $Encrypt(\cdot,\cdot)$ and $Decrypt(\cdot,\cdot)$. Then, the cloud server makes spp, $CBSign$, $CBVeri$, $Sign$, $Veri$, $Encrypt$, $Decrypt$ and pk_S public, and keeps $smsk$ and sk_S secret.

The **ICertify** algorithm (see Steps 4-6 of Figure 2). There are two steps in this algorithm. The cloud server (or certifier which included in cloud server) performs Step 1, and the user performs Step 2.

Step 1. With the user's ID and public key pk, the cloud server selects a time period tp as active time period for the certificate (for example, tp is the time period that the user has bought), computes $cert = BCertify(spp, smsk, ID||pk||tp)$, $\delta = Sign(sk_S, ID||pk||tp)$. Then the cloud server sends the tp, cert and δ via a private channel (tp and δ also can be sent via a public channel).

Step 2. Receiving tp, $cert$ and δ, the user checks the validity of $cert$, and the validity of δ by using the equation $Veri(pk_S, ID||pk||tp, \delta) = 1$. If both of them are valid, the user stores δ, and generates his secret key sk by using his partial secret key psk and certificate $cert$.

The **ITagGen** algorithm (see Steps 7-11 of Figure 2). It is as same as BTagGen algorithm in Subsection 3.2 except that Step 3 are replaced with the following two steps.

Step 3. The cloud server computes receipt $receipt = Sign(sk_S, ID||PK||tp||filename||n||\sigma)$, and sends receipt to the user.

Step 4. Upon receiving the receipt, the user checks the validity of the receipt $receipt$ by using the equation $Veri(pk_S, ID||PK||tp||filename||n||\sigma, receipt) = 1$. If it is invalid, stops, otherwise, stores $receipt$, and deletes the file F, the signature σ and the tags $\tau_i (i = 1, 2, \ldots, n)$ on local storage.

The **ICertifyRenew** algorithm (see Steps 19-23 of Figure 2). If the user's time period tp of active time for the certificate expires, he wants to renew the time period, for example, he buys a new time period service of the cloud server. There are three steps in this algorithm, the user performs Steps 1 and 3, and the cloud server (or certifier included in cloud server) performs Step 2.

Step 1. The user uses his existed public key pk, selects a random number rn, computes $\sigma_C = CLSign(sk, ID||pk||rn||tp)$, $cr = Encrypt(pk_S, ID||pk||rn||tp||\sigma_C)$, and sends cr to the cloud server.

Step 2. Upon receiving cr, the cloud server computes $ID||pk||rn||tp||\sigma_C = Decrypt(sk_S, cr)$, and checks the validity of the signature σ_C by $Veri(ID, pk, ID||pk||rn||tp)$. If the signature σ_C is valid, the cloud server selects a new tp', computes the new certificate $cert' = BCertify(spp, smsk, ID||pk||tp')$, $cert^* = cert' \oplus rn$, $\delta' = Sign(sk_S, ID||pk||tp')$, $\delta^* = Sign(sk_S, cert^*||\delta')$, then sends the $cert^*, \delta', \delta^*, tp'$ to the user via a public channel.

Step 3. Upon receiving $cert*, \delta', \delta*$ and tp', the user checks the validity of $\delta*$ and δ' by using $Veri(pk_S, cert^*||\delta', \delta^*)$ and $Veri(pk_S, ID||pk||tp', \delta')$. If both of them are valid, the user computes $cert' = cert^* \oplus rn$, checks the validity of $cert'$. If it is valid, stores δ', and generates his secret key sk' by using his partial secret key psk and certificate $cert'$.

4.2 The Security Model of the Improved Certificate-based PDP

In this section, we would like to show that the corresponding improved certificate-based PDP is secure.

Note that the improved certificate-based PDP is a variation of the basic certificate-based PDP, therefore, the

security model of the improved version based on the hypothesis that the corresponding basic version is secure. That is to say, the corresponding basic certificate-based PDP can resist the attacks of both public key replacement adversary and malicious certifier adversary.

Compared to the basic certificate-based PDP, the certifier is not an independent trusted third party, but included in the cloud server in the improved version. (Figures 1, 2). Such changes cannot give any advantages for external adversaries but for the cloud server. That is to say, the cloud server has more power, which can not only act as malicious certifier adversary, but also act as public key replacement adversary. Therefore, it is necessary to keep cloud server from replacing the public keys attack. We denote this adversary by cloud server's public key replacement adversary and the public key replacement adversary by external public key replacement adversary.

We use a signature σ of ID and pk signed by the cloud server in Steps 4-6 in Figure 2 to resist cloud server's public key adversary. However, the signature also restricts the ability of cloud server (*i. e.*, certifier). This restriction is not advantageous to the external adversaries, but the user. So, the user probably replaces his own public key for benefits. Therefore, we must keep user from doing replace his own public keys replacement attack. We denote this adversary by user's own public key replacement adversary.

Therefore, if a basic certificate-based PDP scheme is secure, and the corresponding improved certificate-based PDP scheme can resist the attacks of both the cloud server's and the user's public key replacement adversary, then the improved certificate-based PDP scheme is secure.

5 The Security Analysis of Our Proposed Improved General Certificate-based PDP Scheme

In this section, the security of our proposed improved certificate-based PDP scheme will be discussed based on the hypothesis that the corresponding basic certificate-based PDP scheme is secure. According to Section 3, the secure basic certificate-based PDP scheme can resist the attacks of external public key replacement adversary and malicious certifier adversary. Therefore, we use Theorems 1-2 to explain why our improved scheme can resist the attacks of both the cloud server's and the user's public key replacement adversary. Furthermore, we use Theorem 3 to explain why the signature σ described in the ICertify phase in Subsection 4.1 does not need to be verified except in the ICertify phase, Theorem 4 to explain why our improved scheme is non-repudiable, and Theorem 5 to explain why the ICertifyRenew phase of our improved scheme is secure.

Theorem 1. *The proposed improved certificate-based PDP scheme can resist the attack of cloud server's public key replacement adversary, if the corresponding basic certificate-based PDP scheme is secure.*

Proof. As for the cloud server, if the cloud server does not replace the public key pk, he cannot generate the tags because the basic certificate-based PDP scheme is secure. If the cloud server replaces the public key pk with pk^*, he must ensure that $Sign(sk_S, ID||pk^*||tp) = Sign(sk_S, ID||pk||tp)$, otherwise, the user will detect the public key is changed, and show the signature $\delta = Sign(sk_S, ID||pk||tp)$ to indicate the attack from the cloud server. Note that a secure signature algorithm must ensure that different message has different signature, otherwise, the adversary can forge a new message's signature by using an old signature. Therefore, the improved certificate-based PDP scheme can resist the attack of cloud server's public key replacement adversary. □

Theorem 2. *The proposed improved certificate-based PDP scheme can resist the attack of the user's own public key replacement adversary, if the corresponding basic certificate-based PDP scheme is secure.*

Proof. If the user wants to replace his own public key pk with $pk^{\#}$, he must generate the certificate $cert^{\#}$ for secret key. However, the certifier is the only party with the ability to produce certificate $cert^{\#}$ in certificate-based signature [25], therefore in the secure basic certificate-based PDP scheme. So, both the basic certificate-based PDP scheme and the improved certificate-based PDP scheme can resist the attack of user's own public key replacement adversary. □

Theorem 3. *The signature δ described in the ICertify phase in Subsection 4.1 does not need to be verified except in the ICertify phase*

Proof. The cloud server computes $\delta = Sign(sk_S, ID||pk||tp)$ in ICertify phase in Subsection 4.1. The cloud server need not verify the δ because that the user cannot replace his public key according to Theorem 2. The user does not need to verify the δ because he has his own public key pk, and/so he can do everything by using the genuine public key pk. □

Theorem 4. *The proposed improved certificate-based PDP scheme is non-repudiable.*

Proof. If a dishonest user insisted that the cloud server lost his data but he did not update his data at all, the cloud server can require the user to show the receipt $receipt$. Note that if the user did not update his data, he does not have the receipt $receipt$. So, the proposed improved certificate-based PDP scheme is non-repudiable. □

Theorem 5. *The ICertifyRenew phase of our improved certificate-based PDP scheme is secure.*

Proof. In the ICertifyRenew phase of our improved certificate-based PDP scheme, the $cert^* = cert' \oplus rn$. Note that rn is the session key selected by the user and encrypted by the cloud server's public key pk_S, so the adversary cannot know rn, and therefore cannot know the new

Table 1: Compared with other schemes in key management side

Schemes	Trusted third party	Check users' legality	Store the list of legal users	Can TTP be removed
Traditional public key PDP	PKI	identity, date, public key	Yes	No
Identity-based PDP	PKG	identity, date	Yes	No
Certificateless PDP	KGC	identity, date	Yes	No
Our basic certificate-based PDP	certifier	identity, date	Yes	No
Our improved certificate-based PDP	-	-	No	Yes

certificate $cert'$. Furthermore, the user and cloud server have completed authentication by signature in the ICertifyRenew phase. Therefore, the ICertifyRenew phase of our proposed improved certificate-based PDP scheme is secure. □

6 Comparison with Others

All the existed public key PDP schemes have a trusted third party (TTP) to help the cloud server manage the public key. In order to simplify the key management, we removed the TTP and let the cloud server perform the TTP's duty in our improved certificate-based PDP scheme, and gave a detailed analysis in Subsection 6.1 and Table 1. Furthermore, our improved certificate-based PDP scheme can resist key exposure problem, which is described in Subsection 6.2 and Table 2.

6.1 The Key Management Simplification

For the cloud server, a user who has purchased the cloud storage service is a legal user. Usually, the cloud server controls a list of legal users in cloud storage. The list includes the users' identities, time period of purchasing the cloud storage service, *etc.* When a user wants to upload data to the cloud server, the cloud server must check the his legality.

In traditional public key PDP, a public key infrastructure (PKI) is responsible for binding public keys with respective identities of entities by using public key certificate. In order to check a user's legality, the cloud server checks if the identity of the user is in the list of legal users, and if the date is in the time period of purchasing the cloud storage service. If one of them does not hold, stops, otherwise, the cloud server checks the public key certificate of the user which was downloaded from the PKI. If it is invalid, stops, otherwise, the user is legal.

In identity-based PDP scheme, the user's public key is his identity, and the his secret key is generated by private key generation (PKG). In order to check a user's legality, the cloud server only checks if the identity of the user is in the list of legal users, and the date is in the time period of purchasing the cloud storage service. So, the identity-based PDP scheme is simplified in terms of key management, compared with the traditional public key PDP. However, identity-based PDP scheme brings the key escrow problem, *i.e.*, the PKG knows the users' secret key, and can generate the valid data tags.

In order to solve the key escrow problem, certificateless PDP scheme was proposed. In certificateless PDP scheme, the user generates the secret value behind closed doors, and the key generation center (KGC) generates the partial secret key from the user's identity. The user combines the secret value and partial secret key as his secret key, and generates and publishes his public key without certificate. In order to check a user's legality, the cloud server obtains the user's public key from KGC and only checks if the identity of the user is in the list of legal users, and the date is in the time period of purchased the cloud storage service. So, the certificateless PDP scheme is not only simplified in terms of key management compared with the traditional public key PDP, but also solved the key escrow problem of identity-based PDP scheme.

We proposed another two methods to solve the key escrow problem, *i.e.*, basic certificate-based PDP scheme and improved certificate-based PDP scheme. In the basic certificate-based PDP scheme, the user generates the partial secret key and the corresponding public key behind closed doors, and the certifier generates the certificate from the user's identity and public key. The user combines the partial secret key and the certificate as his secret key. In order to check a user's legality, the cloud server obtains the user's public key from the certificate and only checks if the identity of the user is in the list of legal users, and the date is in the time period of purchasing the cloud storage service. So, the certificate-based PDP scheme is not only simplified in terms of key management compared with the traditional public key PDP, but also solved the key escrow problem of identity-based PDP scheme, too. The specific basic certificate-based PDP schemes can refer schemes [18, 21].

In the improved certificate-based PDP scheme, there is no certifier in it, and the cloud server performs the duties of the certifier. All the certificates of users are issued from the cloud server according to the time period in which the user purchased the cloud storage service, *i.e.*, only legal users can have the certificate, and generate his secret key. So, if there is any illegal user, he cannot obtain the certificate, and therefore cannot generate his secret key. Therefore, the cloud server can check the legality of the user by checking the data tags upload by the user. So,

Table 2: Compared with some key exposure resistant schemes

Schemes	[27]	[28]	[21]	Our improved certificate-based PDP
The type of PDP	Traditional public key PDP	Traditional public key PDP	Certificate -based PDP	Certificate -based PDP
Preset the number of key updates	Yes	No	-	No
The level type of key exposure resilience	Weak	Strong	-	Weak
Secret key size	$O((\log\gamma)\lambda)$	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$
Public key size	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$
Challenge overhead	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$
Response overhead	$O((\log\gamma)\lambda)$	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$

the cloud does not need to store the list of legal users.

Note that we cannot remove the trusted third party form the traditional public key PDP, identity-based PDP, or certificateless PDP. In the traditional public key PDP scheme, PKI is responsible for binding public keys with respective identities of entities by using public key certificate. Even if we remove the PKI, and let the cloud server undertake the duty of PKI, the cloud server must check the public key certificate of the users to ensure that the public key and the identity have been bound. In identity-based PDP scheme, PKG knows all the users' secret key. If we remove the PKG, and let the cloud server undertake the duty of PKG, the cloud server can generate the tags. In the certificateless PDP scheme, the user can replace his public key at his will without the help of KGC. If we remove the KGC, and let the cloud server undertake the duty of KGC, the scheme cannot resist the attack of user's own public key replacement adversary. Although each user has a signature δ in ICertify phase in Subsection 4.2, the cloud server does not need to check the validity of the δ when the user updates the data according to Theorem 3 in Section 5. We also described those analysis in Table 1.

6.2 The Key-exposure Resistance

In view of the fact that the certifier is generated by the cloud server, the management of users becomes easier. The cloud server can generate certifier according to the users' purchased service time. Furthermore, it also can split the users' purchased service time to prevent the secret key exposure easily.

In the ICertify phase of subsection 4.2, the cloud server splits the time period tp into γ segments, *i.e.*, $tp_1, tp_2, \ldots, tp_\gamma$. For each tp_i, $i = 1, 2, \ldots, \gamma$. the cloud server replaces tp with tp_i in Subsection 4.2, the scheme can resist key-exposure problem.

From Table 2, we can see that Yu *et al.*'s scheme [27] and Yu and Wang's scheme [28] are Traditional public key PDP schemes, and Wang *et al.*'s scheme [21] and our improved version are certificate-based PDP schemes. However, Wang *et al.*'s scheme [21] cann't resist key-exposure problem. Furthermore, [27] needs presetting the number of key updates in setup phase, and the times of key updated cannot exceed this value. However, the user can update his key as many times as he wants in our improved version and [28]. The secret key size and response overhead are logarithmic in time period segments γ in [27] and those of our improved version and [28] are independent of γ. The [28] pointed that the level type of key exposure resilience is weak in [27], *i.e.*, if the key exposure happens and was not found, the adversary can update the exposed key as same as the user does, and propose a strong key exposure resilient auditing scheme. Our proposed scheme is a weak one either, and deeper investigation will be carried out into the key exposure resilient auditing scheme in the future.

7 Conclusion

Inspired by identity-based PDP and certificateless PDP scheme, we combined certificate-based cryptography with PDP scheme to propose two system models of certificate-based PDP: The basic certificate-based PDP and the improved certificate-based PDP, and gave the security models of them. Then we gave a general construction of improved certificate-based PDP based on the hypothesis that there is a secure basic certificate-based PDP scheme.

Our proposed improved certificate-based PDP removed the certifier, and the cloud server does the duties of certifier. Therefore, our improved certificate-based PDP need not any trusted third party. We gave the secure proof of our improved certificate-based PDP. Furthermore, our improved certificate-based PDP not only simplifies the management of users, but also has some other properties, such as non-repudiability, key-exposure resistance, *etc.*

Acknowledgments

The authors would like to thank the National Natural Science Foundation of China [NOs. U1905211, 61771140, U1405255, 61702100], Natural Science Foundation of Fujian Province of China [No. 2021J011066], Project of Industry-Academic Cooperation of Fujian Provincial Department of Science Technology [NO. 2017H6005], Open-

ing Foundation of Fujian Provincial Key Laboratory of Network Security and Cryptology Research Fund, Fujian Normal University (NO. NSCL-KF2021-01), Scientific Research Staring Foundation of Fujian University of Technology [No. GYZ20171], University-Industry Collaborative Education Program of Ministry of Education of China [No. 202102089006], and Undergraduate Teaching Reform Research Project of Fujian University of Technology [No. jg2021060].

References

[1] H. Abelson, R. Anderson, S. M. Bellovin, J. Benaloh, M. Blazes, W. Diffie, J. Gilmore, P. G. Neumann, R. L. Rivest, J. I. Schilier, B. Schneier, "The risks of key recovery, key escrow, and trusted third-party encryption," *World Wide Web Journal*, vol. 2, pp. 241–257, 1997.

[2] S. S. Al-Riyami, K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology*, pp. 452–473, 2003.

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song, "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07)*, pp. 598–609, 2007.

[4] D. Cash, A. Küpçü, D. Wichs, "Dynamic proofs of retrievability via oblivious RAM," *Proceedings of Advances in Cryptology*, pp. 279–295, 2013.

[5] C. C. Erway, A. Küpçü, C. Papamanthou, R. Tamassia, "Dynamic provable data possession," *ACM Transactions on Information and System Security (TISSEC'15)*, vol. 17, no. 4, pp. 1–29, 2015.

[6] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Advances in Cryptology*, pp. 272–293, 2003.

[7] D. He, N. Kumar, S. Zeadally, H. Wang, "Certificateless provable data possession scheme for cloud-based smart grid data management systems," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1232–1241, 2018.

[8] D. He, S. Zeadally, L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, vol. 12, no. 1, pp. 64–73, 2018.

[9] S. Kamara, K. Lauter, "Cryptographic cloud storage," in *Proceedings of International Conference on Financial Cryptograpy and Data Security*, pp. 136–149, 2010.

[10] K. S. Kim, I. R. Jeong, "A new certificateless signature scheme under enhanced security models," *Security and Communication Networks*, vol. 8, pp. 801–810, 2015.

[11] D. Kim, I. R. Jeong, "Certificateless public auditing protocol with constant verification time," *Security and Communication Networks*, vol. 2017, pp. 1–14, 2017. (`https://doi.org/10.1155/2017/6758618`)

[12] J. G. Li, Z. W. Wang, Y. C. Zhang, "Provably secure certificate-based signature scheme without pairings,' *Information Sciences*, vol. 233, pp. 313–320, 2013.

[13] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, 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, 2019.

[14] J. Ni, K. Zhang, Y. Yu, T. Yang, "Identity-based provable data possession from RSA assumption for secure cloud storage," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2020. DOI: 10.1109/TDSC.2020.3036641.

[15] L. Pang, Y. Hu, Y. Liu, K. Xu, H. Li, "Efficient and secure certificateless signature scheme in the standard model," *International Journal of Communication Systems*, vol. 30, no. 5, 2017.

[16] H. Shacham, B. Waters, "Compact proofs of retrievability," *Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'08)*, pp. 90–107, 2008.

[17] H. Wang, D. He, S. Tang, "Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1165–1176, 2016.

[18] H. Wang, J. Li, "Private certificate-based remote data integrity checking in public clouds," in *Computing and Combinatorics*, pp. 575–586, 2015.

[19] B. Wang, B. Li, H. Li, F. Li, "Certificateless public auditing for data integrity in the cloud," in *Proceedings of IEEE Conference on Communications and Network Security (CNS'13)*, pp. 136–141, 2013.

[20] H. Wang, Q. Wu, B. Qin, J. Domingo-Ferrer, "Identity-based remote data possession checking in public clouds," *IET Information Security*, vol. 8, no. 2, pp. 114–121, 2014.

[21] F. Wang, L. Xu, K. K. R. Choo, y. Zhang, H. Wang, J. Li, "Lightweight certificate-based public/private auditing scheme based on bilinear pairing for cloud storage," *IEEE Access*, vol. 8, pp. 2258-2271, 2020.

[22] F. Wang, L. Xu, J. S. Pan, "Security analysis on "Strongly secure certificateless key-insulated signature secure in the standard model"," in *Proceedings of the Eleventh International Conference on Intel-ligent Information Hiding and Multimedia Signal Processing*, pp. 195–198, 2015.

[23] F. Wang, L. Xu, H. Wang, Z. Chen, "Identity-based non-repudiable dynamic provable data possession in cloud storage," *Computers & Electrical Engineering*, vol. 69, pp. 521–533, 2018.

[24] H. Wang, L. Zhu, C. Xu, Y. Lilong, "A universal method for realizing non-repudiable provable data possession in cloud storage," *Security and Communication Networks*, vol. 9, no. 14, pp. 2291–2301, 2016.

[25] W. Wu, Y. Mu, W. Susilo, X. Huang, "Certificate-based signatures: New definitions and a generic construction from certificateless signatures," in *Proceedings of International Workshop on Information Security Applications*, pp. 99–114, 2008.

[26] H. Yan, J. Li, J. Han, Y. Zhang, "A novel efficient remote data possession checking protocol in cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 78–88, 2017.

[27] J. Yu, K. Ren, C. Wang, V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1167–1179, 2015.

[28] J. Yu, 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.

[29] Y. Zhang, C. Xu, S. Yu, H. Li, X. Zhang, "SCLPV: Secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 4, pp. 159–170, 2016.

[30] L. Zhou, A. Fu, G. Yang, H. Wang, Y. Zhang, "Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2020. DOI: 10.1109/TDSC.2020.3013927.

Biography

Feng Wang received his M.S. degree in Applied Mathematics from the Guangzhou University, in 2006, Ph.D. degree in Applied Mathematics from Fujian Normal University, in 2020. Currently, he is an Associate Professor in the School of Computer Science and Mathematics at Fujian University of Technology. His research interests include computer cryptography, network and information security, *etc.*

Li Xu received the B.S and M.S. degrees from Fujian Normal University, Fuzhou, China, in 1992 and 2001, respectively, and the Ph.D. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004. He is currently a Professor and Doctoral Supervisor with the College of Mathematics and Informatics, Fujian Normal University. He is currently the Director Central of Network and Data, Fujian Normal University, and also the Director of Key Lab of Network Security and Cryptography, Fujian Normal University. He has authored or co-authored over 150 papers in international journals and conferences, including the IEEE Transactions on Computer, ACM Transactions on Sensor Network, the IEEE Transactions on Reliability, the IEEE Transactions on Parallel and Distributed Systems, Information Science, and Computer Network. His current research interests include network and information security, wireless network and communication, complex network and system, and intelligent information in communication network. Dr. Xu has been invited to act as a PC Chair or member at more than 30 international conferences. He is a member of ACM, IEEE, and a Senior Member of CCF and CIE in China.

Zhide Chen received his Ph.D. degree from the School of Computer Science and Technology, Fudan University, China, 2005. He is currently a Professor at the Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, China. His research interests include Network and Information Security, The Internet of Things and Mobile Computing.

Qikui (Henry) Xu is a doctoral student at Westlake University. His research interest lies in the intersection between computational science and biology.

JPEG Image Encryption Algorithm Based on Hyperchaotic, Mixed Hash and Dynamic DNA

Qiu-Yu Zhang and Yu-Tong Ye
(*Corresponding author: Qiu-Yu Zhang*)

School of Computer and communication, Lanzhou University of Technology
No. 287, Lan-Gong-Ping Road, Lanzhou 730050, China
Email: zhangqylz@163.com

(Received July 28, 2020; Revised and Accepted May 24, 2021; First Online Feb. 26, 2022)

Abstract

This paper proposes a new JPEG image encryption algorithm based on hyperchaotic, mixed hash, and dynamic DNA to solve the problem of poor robustness and low efficiency in existing JPEG image encryption schemes. First, the plaintext image is converted into a DNA matrix through DNA coding rules and permutated by the encryption key, generated by the chaotic system and a mixed hash function (MD5 and SHA-256). Secondly, the encryption key is used to dynamically control the DNA operation for diffusion. Finally, the ciphertext image is obtained by decoding the DNA codec rules. Experiments show that the proposed Algorithm is efficiently resistant to statistical attacks and has efficient encryption. Furthermore, the critical space can reach 2^{507}, the critical sensitivity is high, the pixel correlation coefficient is close to 0, the information entropy is close to 8, the UACI and NPCR values are close to the ideal value, and it has good robustness against noise, cropping and JPEG compression attacks.

Keywords: Dynamic DNA Coding; Image Encryption; JPEG Image; Mixed Hash; 5D Lorenz Hyperchaotic System

1 Introduction

With the rapid development of Internet technology, the speed and scale of data dissemination have reached unprecedented levels. The insecure factors in transmission and the image itself are characterized by large data volume and high redundancy makes the security and efficiency of digital image transmission extremely important [2, 23].JPEG images are widely used in daily life. With the widespread popularity of the network, the number of malicious programs such as viruses and Trojan horses is also increasing. Therefore, information security technologies such as JPEG image encryption and data hiding have received extensive attention [3, 6, 7, 9–11, 13, 14, 17–19].

At present, the number of encryption algorithms for specific image formats is still relatively small. Existing JPEG image encryption algorithms are mainly aimed at scrambling DC coefficients. For example, Siricottedumrong *et al.* [18] reduced the color information by dividing the image into smaller blocks to improve encryption performance of the scheme. To increase the algorithm resistance to statistical attacks. Chuman *et al.* [3] based on the puzzle algorithm proposed a new scheme. However, the scheme in [3, 18] inability to resist multiple types of attacks. Li *et al.* [10] combined compression and encryption algorithms to give two encryption schemes to improve compression performance and algorithm security. On the premise of ensuring compression efficiency, Li *et al.* [9] devised a scheme that effectively improved the encryption performance. He *et al.* [7] based on bitstream proposed an encryption scheme which is improved in the aspect of format compatibility and can effectively resist many types of attacks.

Chaos encryption technology is widely used, because of its pseudo-random characteristics and high sensitivity [6, 13]. Man *et al.* [13] devised an image segmentation encryption scheme. The efficiency of the key generation can be improved by using a set of chaotic sequences as the key tool to obtain the key. But the quantized AC coefficient value is significantly smaller than the quantized DC coefficient and the DC coefficient is easy to locate, so this scheme cannot effectively resist contour attacks. Ghazvini *et al.* [6] proposed a scheme by combining the advantages of genetic algorithms and chaotic systems. Use the scramble-diffusion framework to obtain the ciphertext image. The ciphertext image is obtained using a genetic algorithm to optimize the ciphertext image. However, chaotic systems used in these schemes have limitations. The practical application range of the encryption scheme will be limited. Mondal *et al.* [14] devised a secure image encryption scheme based on cellular automata and oblique tent mapping. This scheme is ideal in terms of robustness but poor in encryption efficiency. Combine with the chaotic system and permutation, Li *et*

al. [11] devised an encryption scheme. By summarizing the above encryption schemes, the existing methods have their advantages and disadvantages.

In recent years, hyperchaotic encryption systems have the advantage of high speed but have poor in key sensitivity. To increase the security of multimedia data and combine the advantages of various encryption technologies, more and more hybrid encryption schemes are proposed [17, 19]. Due to the high parallelism and high storage density of DNA molecules, DNA technology is widely used in the field of cryptography to improve the efficiency and security of encryption schemes. Salman *et al.* [17] proposed an encryption scheme based on DNA coding and chaotic systems. This scheme has better encryption performance and shorter encryption time for JPEG images. Thanikaiselvan *et al.* [19] proposed a two-stage reversible data hiding scheme based on DNA coding and chaotic systems. Steganographic images are encrypted by combining hash function, DNA coding and chaotic system.

Therefore, to overcome the deficiency of existing methods, this paper takes the JPEG image as the research carrier and presents a JPEG image encryption algorithm based on hyperchaotic, mixed hash and dynamic DNA. The main contributions of this paper can be summarized as follows:

1) SHA-256 is combined with MD5 hash function, and plaintext hash sequence is obtained by calculating the plaintext image, to increase the encryption systems for the chosen-plaintext attacks and known-plaintext attacks resistance.

2) DNA coding and operation rules are dynamically controlled by an encryption key, and the operation results cannot be accurately predicted, which improves the security of the key.

3) 5D Lorenz hyperchaotic system is used to expand the key space of the encryption algorithm, making the encryption algorithm more secure.

The remaining part of this paper is organized as follows. Section 2 describes the relevant theories in detail. Section 3 gives the JPEG image encryption scheme and its processing. Section 4 gives the experimental results and the performance analysis compared with other related methods. Finally, we conclude our work in Section 5.

2 Related Theories Analysis

2.1 5D Lorenz Hyperchaotic System

The classical Lorenz system dynamic equation [12] is shown in Equation (1):

$$\begin{cases} \frac{dx}{dt} = a(y-x) \\ \frac{dy}{dt} = bx - xz - y \\ \frac{dz}{dt} = xy - cz \end{cases} \tag{1}$$

where $a = 10$, $c = 8/3$, when $b > 24.74$,the Lorenz system enters into chaotic state.

The low-dimensional chaotic system has a simple structure and low algorithm complexity, but the key space is small and cannot effectively resist brute-force attacks. The control parameters, dynamic characteristics, and initial conditions of hyperchaotic systems are more complex than those of low-dimensional chaotic systems, which can expand the key space and possess stronger pseudo-randomness. Therefore, based on Equation (1), this paper constructs a 5D Lorenz hyperchaotic system by introducing two new variables u and w and nonlinear terms. The equation is shown in Equation (2):

$$\begin{cases} \frac{dx}{dt} = b(y-x) + yz - u + w \\ \frac{dy}{dt} = a(x+y) - xzu \\ \frac{dz}{dt} = -(c-a)z + xyu \\ \frac{du}{dt} = mu - xyz \\ \frac{dw}{dt} = -hx - hy \end{cases} \tag{2}$$

where x, y, z, u and w is the state variable of the constructed 5D Lorenz hyperchaotic system. a, b, c, m, and h are used to represent system parameters.

When $a = 27.02$, $b = 48$, $c = 33$, $m = 24$, $h = 9.48$, the Lyapunov exponents are $L_1 = 5.18340 > 0$, $L_2 = 3.00559 > 0$, $L_3 = 0.00054 \approx 0$, $L_4 = -17.74081 < 0$, $L_5 = -29.44874 < 0$,the chaotic system of Equation (2) is in a hyperchaotic state. Figure 1 shows the Lyapunov exponent diagram of 5D Lorenz hyperchaotic system. Figure 2 shows the chaotic attractor in each plane of the 5D Lorenz hyperchaotic system.

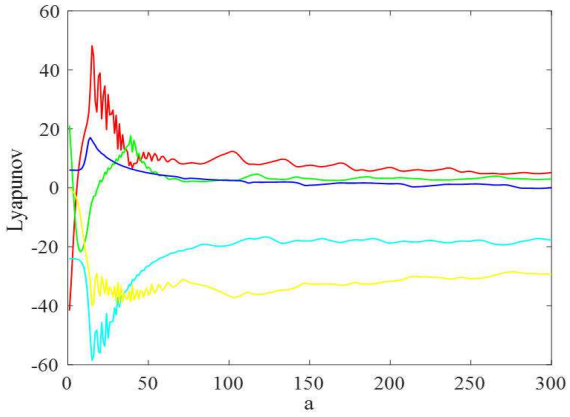
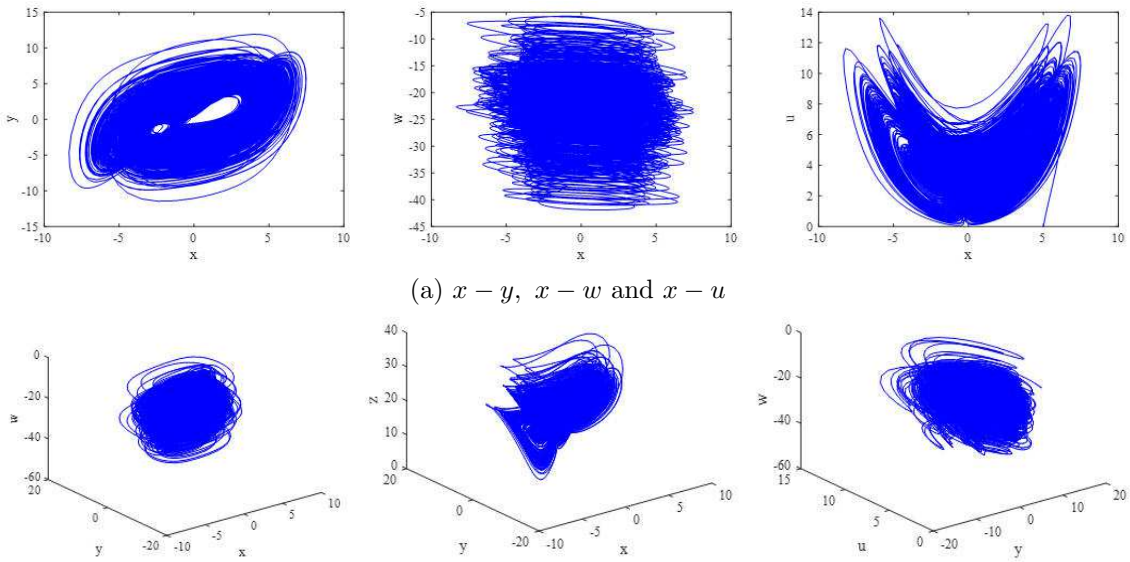


Figure 1: The Lyapunov exponent diagram of 5D Lorenz hyperchaotic system

2.2 Mixed Hash Function

During the process of image encryption, the hash function is usually used to increase the algorithm's resistance to plaintext attacks [9]. Therefore, hash functions such as MD5 and SHA-256 are widely used in the information security field. When any length of data is input, the output of a hash function is fixed in size. If the input of



(a) $x-y$, $x-w$ and $x-u$

(b) $x-y-w$, $x-y-z$ and $y-u-w$

Figure 2: The chaotic attractor in each plane of the 5D Lorenz hyperchaotic system

the hash function is modified slightly, the output will be completely different. Therefore, the attacker cannot infer the hash sequence to obtain plaintext image data [22]. Although the used hash function is pre-image resistance, in some encryption systems, the key is set very simply, and the attacker can successfully crack some messages through brute-force attacks [9]. To solve this problem, this paper combines MD5 and SHA-256 hash functions and adding a control parameter $KeyHex$ to encrypt the key multiple times to increases the security of the encryption algorithm.

Calculate the sum of the pixel values of all rows and columns of the plaintext image I with a size of $M \times N$, and record as $Sumrow$ and $Sumcol$ respectively. First, MD5 is used to calculate $Sumrow$, $SumCol$, and $KeyHex$, respectively. Then, the results are combined and calculated using SHA-256. Where $Keyhex$ = 6b679b3c71108d30a79e610526a8c18ef974c176f4e529f684748ac019931209. The method of calculating the mixed hash function is shown in Equation (3).

$$D = SHA-(256(MD5(SumRow)MD5(SumCol)MD5(KeyHex)). \quad (3)$$

2.3 DNA Coding and Operations

2.3.1 DNA coding

The four nucleic acid bases form the deoxyribonucleic acid (DNA) sequence [8]. The four binary numbers from 00 to 11 are used to represent A, C, G, and T, and only 8 of the 24 DNA coding schemes obtained to meet the conditions [21]. Table 1 shows the DNA coding rules. Table 2 shows examples of the application of DNA coding rules.

In the process of grayscale image encryption, the length of the binary sequence of each pixel is 8, and the corresponding DNA sequence length is 4. As shown in Table 2, the pixel grayscale value is 231 and its binary sequence is 11100111. If encoding and decoding according to different rules, completely different results will be obtained. In this paper, eight kinds of DNA coding rules are dynamically selected through the encryption key L_k. Coding operations are performed from top to bottom, left to right, and the random number obtained is 1 to 8. The calculation method is shown in Equation (4).

Table 1: DNA coding rules

Rules	1	2	3	4	5	6	7	8
00	A	A	C	C	G	G	T	T
01	G	C	T	A	T	A	C	G
10	C	G	A	T	A	T	G	C
11	T	T	G	G	C	C	A	A

$$E_{DC}^{(k)} = floor\,(8 * (L_k/3)) + 1 \quad (4)$$

where the function $floor(\cdot)$ indicates round down.

2.3.2 DNA Operations

DNA sequence operations are the same as binary operations [8, 21]. According to eight different DNA coding schemes, eight different DNA operation rules can be obtained. Each DNA coding rule corresponds to a DNA operation.

In this paper, the choice of DNA operation rules is determined by the encryption key L_k. The random selection rule of DNA operation is shown in Equation (5):

$$L_{operation} = floor\,(7 * (L_K/3)) + 1 \quad (5)$$

The corresponding relationship between the value of $L_{operation}$ and the operation rules is shown in Table 3.

Table 2: The examples of the application of DNA coding rules

Rules	1	2	3	4	5	6	7	8
231	TCGT	TGCT	GATG	GTAG	CATC	CTAC	AGCA	ACGA

Table 3: $L_{operation}$ and DNA operation rules comparison table

$L_{operation}$	1	2	3	4	5	6	7
operation rules	Addition	subtraction	multiplication	XNOR	XOR	left shift	right shift

The calculation results of Equation (5) are random numbers with values from 1 to 7, and corresponding operation rules can be carried out by referring to Table 3.

3 The Proposed Algorithm

Figure 3 shows the processing flow chart of the JPEG image encryption algorithm. The encryption algorithm is mainly realized through 5D Lorenze hyperchaotic system, mixed hash function (MD5 and SHA-256), DNA encoding/decoding and DNA operation.

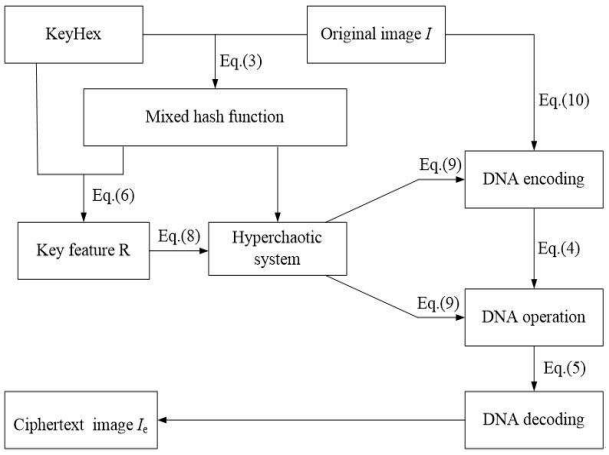


Figure 3: The processing flow chart of the JPEG image encryption algorithm

3.1 Generation of Key

In order to improve the resistance of image encryption algorithm to choice plaintext attack and known-plaintext attack, this paper adopts a mixed hash function to calculate the hash value of plaintext image as the initial parameter of a chaotic system. The specific steps of the key generation are as follows.

Step 1: Hash sequence D is calculated using the mixed hash function of Equation (3).

Step 2: Determine the value of R from the hash sequence D, and then determine the initial value of the hyperchaotic system by R. The calculation formula is shown in Equation (6).

$$R = hex2dec(D) \oplus hex2dec(KeyHex) \tag{6}$$

where $\oplus$ is XOR operation, $hex2dec(\cdot)$ reepresents the conversion from hexadecimal to decimal.

Step 3: Divide $hex2dec$ (D) into a group of 8 bits and get 32 segmented key d_i. Therefore, D' can also be expressed as:

$$D' = d_1, d_2, d_3, \cdots, d_{32} \tag{7}$$

Step 4: The initial value of chaos is obtained by Equation (8). The Equation (2) is continuously cycled $d_{31} + d_{32} + R$ times to obtain the chaotic sequence L with good randomness.

$$\mathrm{L} = \begin{cases} x_{(1)} = ((((d_3 \oplus d_4) \oplus (d_2 \oplus d_5)) \oplus (d_1 \oplus d_6)) \oplus R)/256 \\ y_{(1)} = ((((d_y \oplus d_{10}) \oplus (d_8 \oplus d_{11})) \oplus (d_7 \oplus d_{12})) \oplus R)/256 \\ z_{(1)} = ((((d_{15} \oplus d_{16}) \oplus (d_{14} \oplus d_{17})) \oplus (d_{13} \oplus d_{18})) \oplus R)/256 \\ u_{(1)} = ((((d_{21} \oplus d_{22}) \oplus (d_{20} \oplus d_{23})) \oplus (d_{19} \oplus d_{24})) \oplus R)/256 \\ w_{(1)} = ((((d_{27} \oplus d_{28}) \oplus (d_{26} \oplus d_{29})) \oplus (d_{25} \oplus d_{30})) \oplus R)/256 \end{cases} \tag{8}$$

Step 5: Chaotic sequence L can be calculated by Equation (9) to obtain the encryption key L_k.

$$L_k = \mod(floor(4 * L), 4) \tag{9}$$

The key generated by the above equation is closely related to the plaintext image. It can guarantee the unique key of each encryption process.

3.2 The Encryption Algorithm

Assume that the encrypted object is a grayscale image I of size $M \times N$. First, DNA encoding of the plaintext image which is controlled by the encryption key L_k. L_k is generated by a mixed hash algorithm (SHA-256 and MD5) and 5D Lorenz hyperchaotic system. Then, the encoded image is rearranged in the order of L_k to achieve pixel permutation. Finally, the encryption key L_k is used to dynamically select DNA operation rules, Perform relevant calculations, the results were decoded according to DNA coding rules to obtain the ciphertext image I_e.

The specific encryption processing steps are as follows:

Step 1: Obtain L_k through the key generation, and Two-dimensional matrix P is obtained from the original JPEG image.

Step 2: Divide P into 4 sub-blocks. The calculation method is shown in Equation (10).

$$E_{enc} = fix(P/4) \tag{10}$$

where the function $fix(\cdot)$ represents the rounding of the zero direction of the element in $P/4$.

Step 3: Encode each pixel value in E_{enc} using Equation (4) and the rules in Table 1.

Step 4: The L_k generated by the hyperchaotic system is sorted to obtain the sorted sequence L_p. The calculation method is shown in Equation (11):

$$(B, L_p) = sort(L_K) \tag{11}$$

where L_p is an array whose Size is equal to size(L_k). Each column of L_p is a permutation vector corresponding to the elements of column vectors in L_k.

Step 5: Transpose the subscript array L_p to get L'_p, and obtain the permutation image I_p through Equation (12).

$$I_p = image(L'_p) \tag{12}$$

where the function image($\cdot$) means to display L'_p as an image.

Step 6: Perform DNA operation on I_p and L_k according to the DNA operation rules in Equation (5) and Table 3.

Step 7: The DNA encoding and decoding rules in Equation (4) and Table 1 were used to decode the DNA operation results and obtain ciphertext image I_e.

3.3 The Decryption Algorithm

The receiver owns the ciphertext image I_e and the control parameter *KeyHex*. The decryption process of ciphertext image is as follows:

Step 1: Perform the key generation method in Section 3.1 to obtain R and L_k.

Step 2: Obtain the DNA coding sequence by executing Equation (10) and Equation (4).

Step 3: According to Equation (5) and the rules in Table 3, using the inverse operation to obtain the sequence L_p.

Step 4: Perform reverse permutation operation on L_p to get the decrypted image. Equation (13) is the calculation formula of permutation inverse operation.

$$I_{pi} = image(:) \tag{13}$$

4 Experimental Results and Analysis

The software simulation environment is MATLAB R2017a and the experimental platform is Intel (R)Core (TM) I5-2410m CPU @2.30ghz, 4.00GB RAM, Windows 7(64-bit) operating system. Four images Lena, Barbara, Baboon and Boat with a size of 512×512 were selected from USC-SIPI [20] image database as test images. To carry out the follow-up experimental work, OpenCV [15] is used to convert the image format to JPG. Figure 4 shows the original image, ciphertext image and decrypted image of test images.

4.1 Statistical Analysis

4.1.1 Histogram Analysis

The more even the distribution of the ciphertext image histogram, the higher the security of the encryption algorithm [5]. Figure 5 shows the histograms of plaintext images and ciphertext images.

As shown in Figure 5, the pixel values of the plaintext image are concentrated in some areas, while the pixel values in the ciphertext image histogram are uniformly distributed. This indicates that the statistical characteristics of digital images are destroyed and attackers cannot obtain any valid information related to plaintext through statistical attacks. Therefore, the proposed algorithm can resist the attack of statistical analysis.

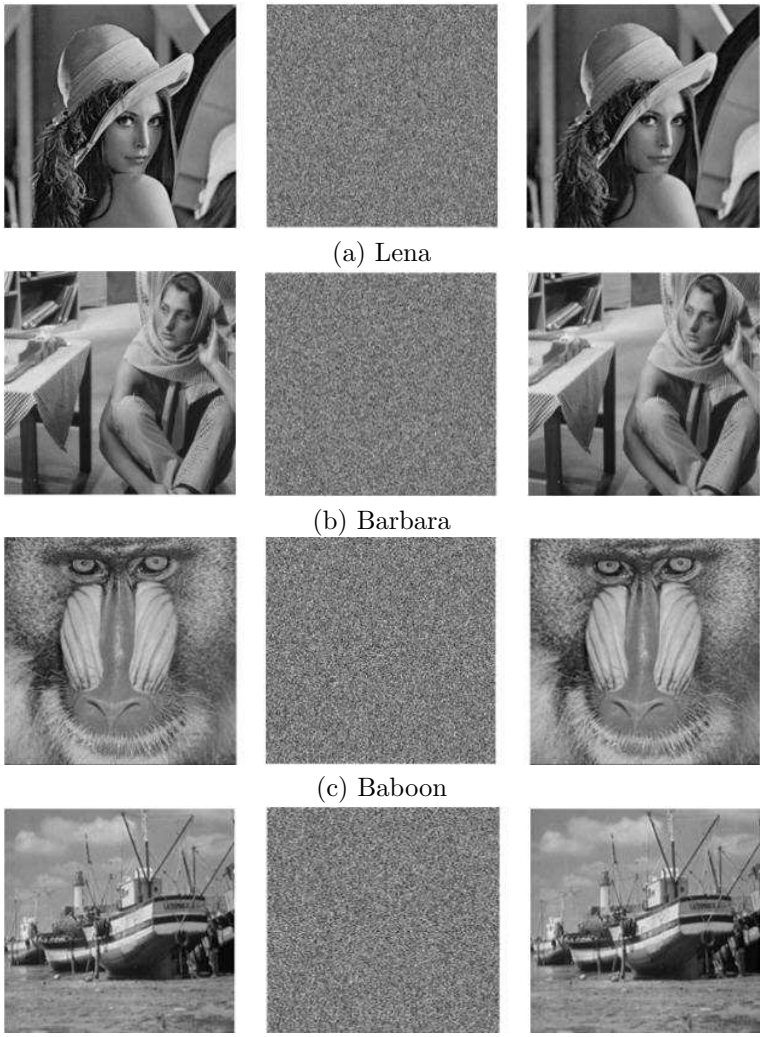
4.1.2 Correlation Analysis

It is necessary to weaken the strong correlation between adjacent pixels of original images to improve the security of multimedia data [26]. Three thousand pairs of pixels were randomly selected from three directions, and the correlation coefficient was calculated by the following equations:

$$\begin{aligned}
\rho_{xy} &= \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)} \times \sqrt{D(x)}} \\
\operatorname{cov}(x,y) &= \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)\left(y_i - E(y)\right) \\
E(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i \\
D(x) &= \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2
\end{aligned}$$

where x and y respectively represent the gray value of two adjacent pixels, the total number is represented by N, the covariance of the variables x and y is represented by $\operatorname{cov}(x,y)$. The expectation and variance of the variable are represented by $E(\cdot)$ and $D(\cdot)$ respectively.

Table 4 shows the correlation between adjacent pixels of the plaintext image and ciphertext image. Table 5



(a) Lena

(b) Barbara

(c) Baboon

(d) Boat

Figure 4: The encryption/decryption results of test images

Table 4: Correlation between adjacent pixels of plaintext image and ciphertext image

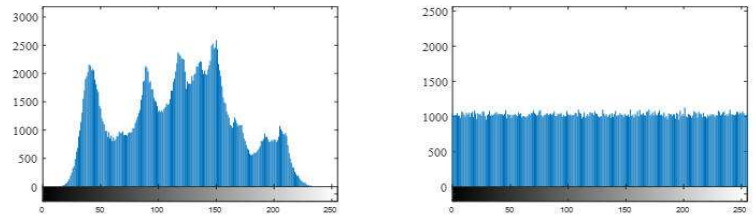
Test image	Plaintext image				Ciphertext image		
	Horizontal	Vertical	Diagonal		Horizontal	Vertical	Diagonal
Lena	0.9703	0.9841	0.9557		-0.0010	-0.0009	0.0007
Barbara	0.8899	0.9600	0.8859		0.0044	-0.0053	-0.0026
Baboon	0.8844	0.7633	0.7365		0.0079	-0.0001	-0.0009
Boat	0.9576	0.9801	0.9444		-0.0097	-0.0004	0.0060

shows the comparative analysis of the correlation between the proposed algorithm [6, 10, 11, 14]. Figure 6 shows the correlation between Lena plaintext image and its corresponding ciphertext image in all directions.

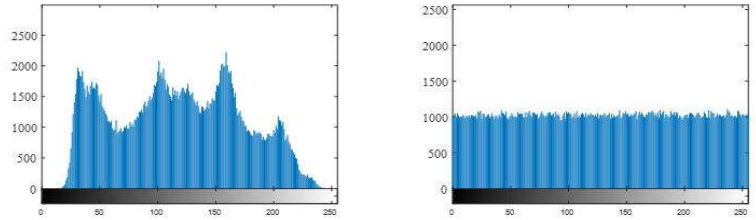
As shown in Table 4, the correlation coefficient of the plaintext image is close to 1, while the ciphertext image is closer to 0, and its value is much smaller than that of the plaintext image. As shown in Table 5, the resistance of the proposed algorithm to statistical attacks is significantly better than the encryption algorithm in [6, 10, 11, 14]. As shown in Figure 6, the pixels of the Lena plaintext image are clustered together, while the pixels of the ciphertext

Table 5: Correlation analysis between adjacent pixels of Lena ciphertext image

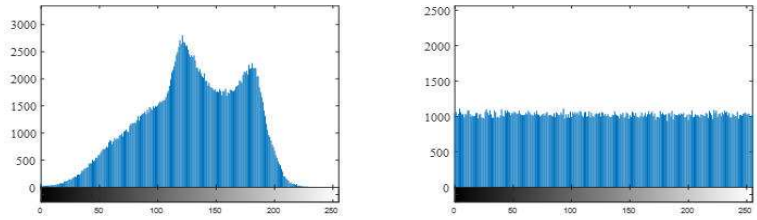
Methods		Horizontal	Vertical	Diagonal
[10]		-0.0025	0.0006	0.0072
[6]		-0.0033	-0.0040	-0.0002
[14]		0.0015	-0.0043	-0.0023
[11]		-0.3458	-0.0027	0.0033
Proposed		-0.0010	-0.0009	0.0007



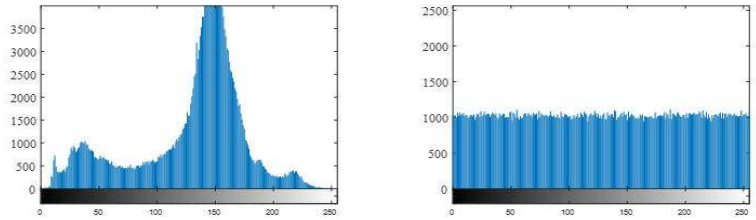
(a) Histogram of Lena



(b) Histogram of Barbara

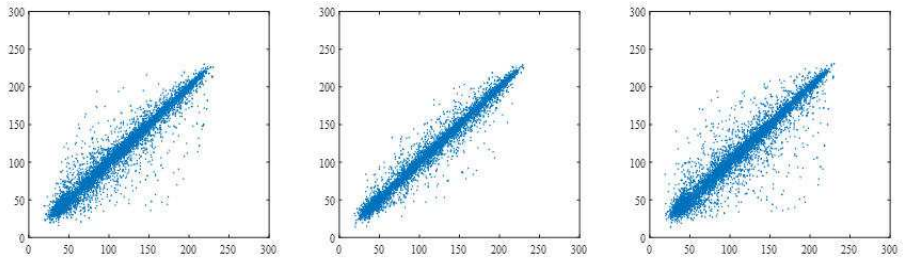


(c) Histogram of Baboon

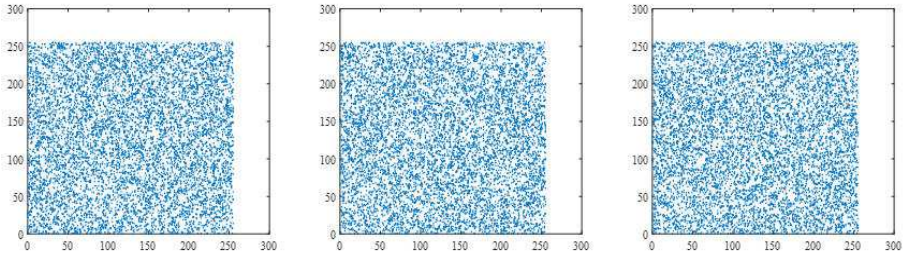


(d) Histogram of Boat

Figure 5: The histograms of plaintext images and ciphertext images of test images



(a) Horizontal, vertical, and diagonal directions of plaintext images



(b) Horizontal, vertical and diagonal directions of ciphertext images

Figure 6: Correlation comparison between plaintext image and ciphertext image of Lena image

Table 6: The entropy of the ciphertext image

	Lena	Barbara	Baboon	Boat
Original image	7.598324	7.632119	7.374090	7.203873
Ciphertext image	7.999314	7.999336	7.999262	7.999293

Table 7: Comparative analysis of information entropy of Lena ciphertext image

Methods	[7]	[6]	[14]	[11]	Proposed
Information entropy	7.80	7.9990	7.9993	7.8232	7.999314

Table 8: NPCR and UACI values of ciphertext images (100%)

Test image	Pixel position	Pixel value change	NPCR(%)	UACI(%)
Lena	(183,95)	$67 \rightarrow 68$	99.6117	33.4469
Barbara	(504,108)	$104 \rightarrow 105$	99.6159	33.4252
Baboon	(24,11)	$200 \rightarrow 201$	99.6315	33.4667
Boat	(368,180)	$35 \rightarrow 36$	99.6189	33.4474

image are more scattered. Therefore, the proposed algorithm can resist statistical analysis attacks.

4.2 Information Entropy Analysis

The information entropy [1] is calculated by Equation (14).

$$H = -\sum_{i=0}^{2^n-1} p_{(m_i)} \log_2 \frac{1}{p_{(m_i)}} \tag{14}$$

where the probability of occurrence of m_i is represented by $p(m_i)$.

Theoretically, the pixel values of the ciphertext image will have very good randomness when the information entropy is closer to 8. The information entropy of 4 test images is shown in Table 6. Table 7 compares and analyzes the entropy of Lena ciphertext image in the proposed algorithm [6,7,11,14].

As shown in Table 6, the information entropy of all test images is above 7.9992, close to the ideal entropy value of 8. Through the comparative analysis of information entropy in Table 7, it can be seen that the information entropy of the proposed algorithm is superior to the [6, 7, 11, 14]. Therefore, the proposed algorithm has good security to resist entropy attacks.

4.3 Differential Attack Analysis

The encryption algorithm is more resistant to differential attacks when the ciphertext image is more sensitive to the plaintext image, the number of pixels changes rate (NPCR) and the unified average changing intensity (UACI) [4,17] are often used as evaluation criteria. The calculation method is detailed in the following equations:

$$\begin{aligned} NPCR &= \frac{\sum_{i=0}^{M=1}\sum_{j=0}^{N=1} D(i,j)}{M \times N} \\ D_{(i,j)} &= \begin{cases} 0, c_1(i,j) = c_2(i,j) \\ 1, c_1(i,j) \neq c_2(i,j) \end{cases} \\ UACI &= \frac{1}{M \times N} \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \frac{|c_1(i,j) - c_2(i,j)|}{255} \end{aligned}$$

where the pixel values of the ciphertext image before and after changing the plaintext image are represented by $c_1(i,j)$ and $c_2(i,j)$ respectively, the length and width of the image are represented by M and N respectively.

Table 8 shows the NPCR and UACI values of the ciphertext image. Table 9 shows the comparison and analysis of NPCR and UACI in Lena ciphertext images of the proposed algorithm and [6,7,10,11,14].

Table 9: Different encryption algorithms NPCR and UACI comparison (100%)

Method	NPCR(%)	UACI(%)
[10]	95.43	21.34
[7]	99.45	27.03
[6]	99.57	33.35
[14]	99.6881	37.5600
[11]	99.6827	33.3781
Proposed	99.6117	33.4469

As shown in Table 8, the UACI values of different images are close to 33.46%, indicating that the pixel values have changed. The NPCR of different images is close to 100%, indicating that the position where the ciphertext image has changed. Therefore, the proposed algorithm can resistance differential attacks. As shown in Table 9, the proposed algorithm is compared with Lena ciphertext images of NPCR and UACI in the [6,7,10,11,14], the analysis shows that the proposed algorithm has better UACI and PSNR values, close to the optimal value, compared to [6,7,10,11,14].

4.4 Exhaustive Attack Analysis

4.4.1 Key Space Analysis

The more effective against brute-force attacks when the key space is larger [17]. Valid keys in the proposed algorithm are as follows:

1) Hash values generated by plaintext images using the mixed hash function;

2) Initial values of chaotic systems x_0, y_0, z_0, u_0 and w_0;

3) DNA coding rules (8 types).

When the computer precision is set to 10^{-15}, then the result of $2^{256}\times 8\times 10^{15}\times 10^{15}\times 10^{15}\times 10^{15}\times 10^{15} = 9.248\times 10^{152} \approx 2^{507}$ is the key space of the proposed algorithm. Table 10 shows the comparative analysis of the key space.

As shown in Table 10, compared with the key space in the [6, 10, 11, 14], the proposed algorithm is significantly larger. In contrast with the [7], the key space of the proposed algorithm is much smaller than that of the encryption scheme in [7]. This is because in the [7], each nonzero AC coefficient corresponds to an ACC, the number of ACCs in the entropy-coded data is large, When the number of ACCs involved in scrambling is much larger than the number of DCCs, this encryption step will result has a large key space. The key space of the proposed algorithm is approximately 2^{507} which is much larger than in 2^{100}. Therefore, the proposed algorithm can resist brute-force attacks.

4.4.2 Key Sensitivity Analysis

After slightly changing the encryption key, the corresponding result should be completely different [22]. Similarly, even minor changes to the encryption key cannot be able to get the correct plaintext image. For example, if the parameter x_0 of this algorithm is changed to $x_0 + 10^{-15}$, the correct decrypted image will not be obtained. The key sensitivity test analysis results of Lena ciphertext images are shown in Figure 7.

As can be seen from Figure 7, a slight modification to the key, the decryption results obtained are completely different. Moreover, if the attacker decrypts the image with an incorrect key, any valid feature information related to the original image will not be obtained.

4.5 Robustness Analysis

The anti-interference ability of an encryption system can be tested with robustness [16, 24, 25]. In the actual communication process, it is often affected by the transmission environment. To verify the robustness of the proposed algorithm, noise attacks, cropping attacks, and JPEG compression attacks will be used.

4.5.1 Noise Attack Analysis

In the image transmission process, inevitable noise will interfere with image decryption results [16, 24]. The robustness of the encryption algorithm is tested by adding noise of different intensities to the ciphertext image and using the same method for encryption and decryption operations. Figure 8 shows a Lena ciphertext image with different intensities of salt and pepper noise or Gaussian noise with a mean value of 0 and different variances and its corresponding decrypted image.

As can be seen from Figure 8, both can recover plaintext images, no matter how different kinds of noise are added. The image information is still identifiable, after adding salt and pepper noise with a noise intensity of 0.2 or adding Gaussian noise with a mean of 0 and a variance of 0.001. However, the image quality of the decrypted image after adding Gaussian noise is lower than that of adding salt and pepper noise.

4.5.2 Cropping Attack Analysis

Test the robustness of the algorithm against cropping attacks by cropping part of the encrypted image [25]. Fig.9 shows the ciphertext image and the decrypted image after cropping 1/16, 1/8, 1/4 and 1/2.

As can be seen from Figure 9. The decrypted image can still be identify after cropping the ciphertext image 1/2. Therefore, the proposed algorithm has better robustness for cropping attacks, and the image quality of decrypted images will be affected when the crop size is too large.

4.5.3 JPEG Compression

Perform JPEG compression on the ciphertext image to verify the robustness of the algorithm under JPEG compression. The measurement of JPEG compression adopts a quality factor, the range of quality factor is between 1 $\sim$ 100 [18]. Generally, the change of pixel value between plaintext and ciphertext image is measured by PSNR. The calculation method is shown as follows:

$$PSNR = 10 \times \log_{10}\left(\frac{M \times N \times 255^2}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(P(i,j) - C(i,j))^2}\right)$$

where the expected value in the plaintext image is represented by $P(i, j)$; the error value at the same position in the ciphertext image is represented by $C(i, j)$.

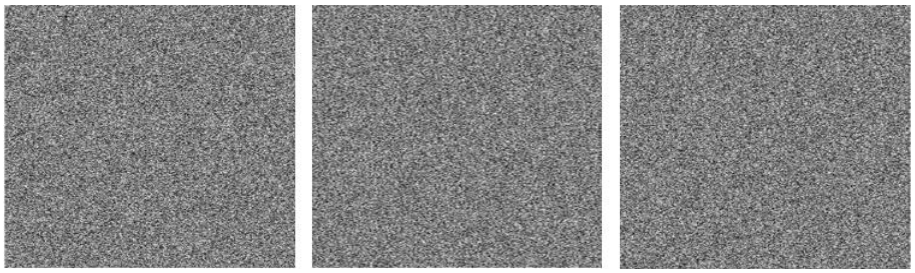
The calculation method for SSIM is defined as follows:

$$SSIM = \frac{\left(2u_p u_c + (0.01L)^2\right)\left(2\sigma_{pc} + (0.03L)^2\right)}{\left(u_p^2 + u_c^2 + (0.01L)^2\right) + \left(\sigma_p^2 + \sigma_c^2 + (0.03L)^2\right)}$$

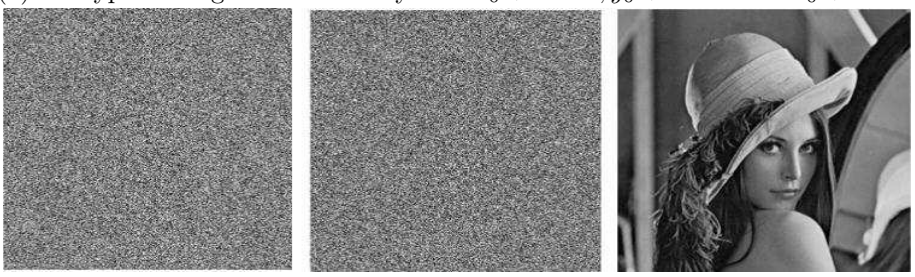
where the plaintext image and the restored image are represented by u_p and u_e respectively. The variance of the original image and the restored image is represented by δ_p and δ_c. δ_{pc} represents the covariance of the original and recovered images, and L represents the dynamic range of pixel values.

Table 10: Comparative analysis of key space

Method	[10]	[7]	[6]	[14]	[11]	Proposed
key space	$2^{256} \times 10^{42}$	3.15×10^{237206}	2^{224}	$2^2 56$	$2^2 56$	9.248×10^{152}

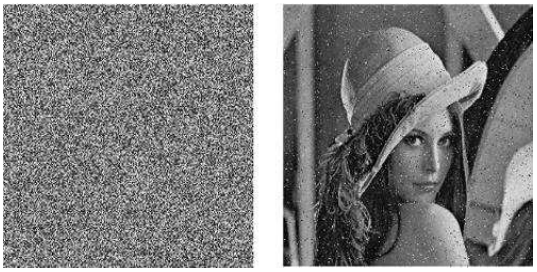


(a) Decrypted images when the keys are $x_0 + 10^{-15}, y_0 + 10^{-15}$ and $z_0 + 10^{-15}$

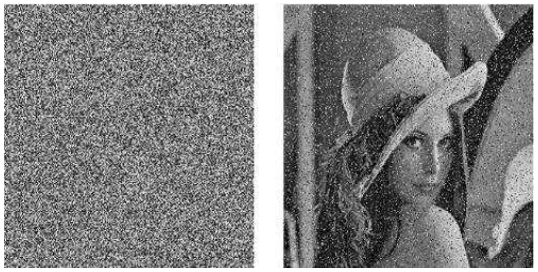


(b) Decrypted images when the keys are $u_0 + 10^{-15}, w_0 + 10^{-15}$ and correct key

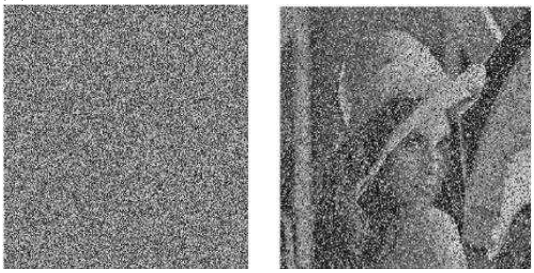
Figure 7: Key sensitivity analysis



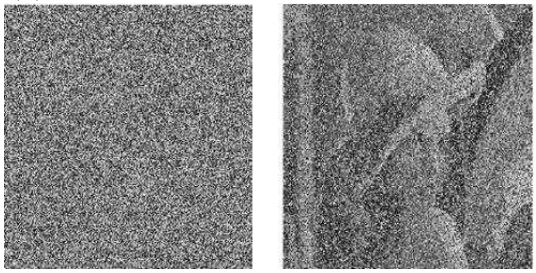
(a) The salt and pepper noise intensity of 0.05



(b) The salt and pepper noise intensity of 0.2

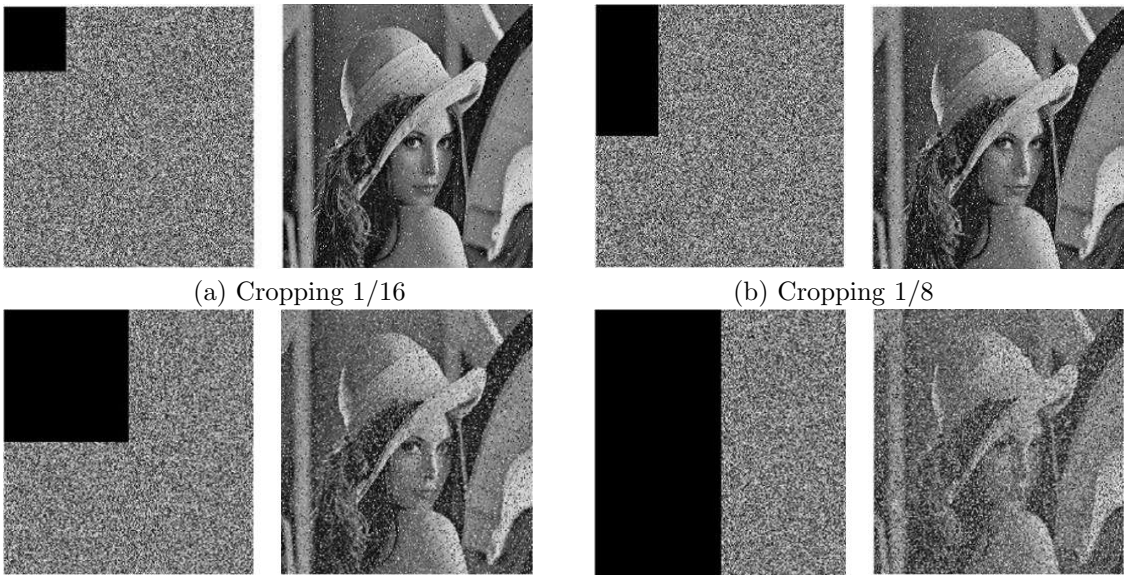


(c) The Gaussian noise variance 0.0001



(d) The Gaussian noise variance 0.001

Figure 8: Robustness analysis of Lena image encryption and decryption after adding Noise



(a) Cropping 1/16 (b) Cropping 1/8

(c) Cropping 1/4 (d) Cropping 1/2

Figure 9: Robustness analysis of Lena image encryption and decryption after cropping

Encrypts 4 images of Lena, Barbara, Baboon and Boat under different QF values with the proposed algorithm, and compared with the algorithm in Li's (2019) [10] and the algorithm in Li's (2017) [9]. Figure 10 shows the comparison result of average PSNR and SSIM of ciphertext image under different QF values.

As can be seen in Figure 10, the average PSNR values of the ciphertext image obtained by the proposed algorithm is maintained above 20 dB. Therefore, the security of encryption algorithm can be guaranteed. The average PSNR and SSIM values of [9] are lower than the proposed scheme and Algorithm 1 of [10]. This is because Algorithm 1 of [10] only performs encryption in the conversion phase and quantization phase. However, all the coefficients in [9] are encrypted, which sacrifices compression efficiency. Therefore, the proposed algorithm can resist JPEG compression attacks.

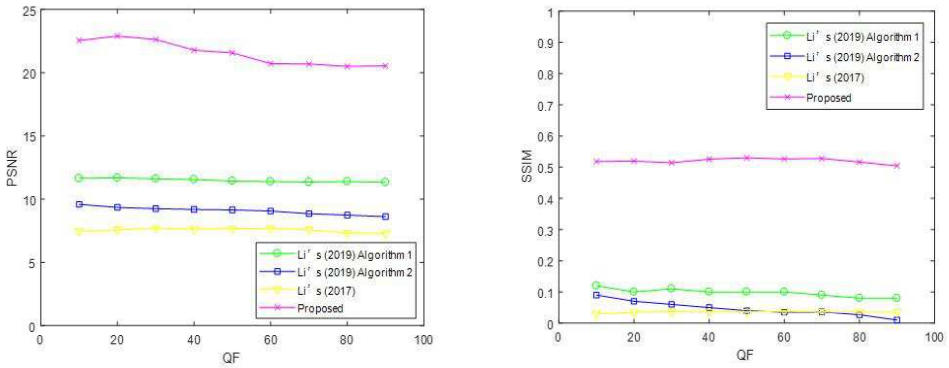
In order to further evaluate noise, cropping, JPEG compression attack after the decrypted image quality, it will have salt and pepper noise (0.05), Gaussian noise (0.0001), cropping (1/16), JPEG compression (QF = 50) after four kinds of attack the decrypted image PSNR and SSIM values and not subject to any attacks, comparing the decrypted image PSNR and SSIM values to evaluate the difference between different decrypted image, the results are shown in Table 11. Table 12 shows the comparison result of the average PSNR value of the decrypted image after the attack between the proposed algorithm and the method of [14].

As shown in Table 11, the decrypted image after being attacked by noise, cropping, and JPEG compression is compared with the decrypted image without any attack. It is found that the SSIM of the decrypted image before the attack is greater than 0.7300 and the PSNR remains above 40 dB. After adding noise, cropping and JPEG compression attacks, the values of SSIM and PSNR decrease significantly.

As can be seen in Table 12, the average PSNR values of the decrypted image after the noise attack of the proposed algorithm is close to or better than [14]. After the proposed algorithm suffers from cropping and JPEG compression attacks, the average PSNR values of the decrypted image is slightly weaker than the [14]. The results show that the decrypted image can still recover and identify when the ciphertext image is severely distorted. Therefore, the proposed algorithm has the ability to resist noise, cropping and JPEG compression attacks.

4.6 Time Complexity Analysis

The computational complexity of the proposed algorithm is mainly related to the encryption algorithm steps in Section 3.2. The time consumption of Step 1 is $O(4\times M\times N)$ because the time consumption is mainly the floating-point operation of generating chaotic sequences. Steps 2, 3, 4, and 5, the time complexity of each step is $O(4\times M\times N)$. This is because the time consumption is mainly the number of operations for image pixel conversion. The time complexity of Step 6 is $O(16\times M\times N)$. The time consumption is mainly the number of DNA operations. The time complexity of Step 7 is $O(16\times M\times N)$ because the time consumption is mainly the number of conversion operations of the DNA matrix. Therefore, the total time complexity of the proposed algorithm is $O(16\times M\times N)$. By testing the encryption algorithm of the proposed, the average encryption time of Lena image with a size of 512×512 is 2.573 s, which is better than the encryption time of 3.007 s in [14].



(a) Average PSNR values (b) Average SSIM values

Figure 10: Comparison of average PSNR and SSIM of ciphertext image under different QF values

Table 11: SSIM and PSNR values of ciphertext images before and after different attacks

decrypted image	Attack types	evaluation index	Lena	Barbara	Baboon	Boat
Before the attack	-	PSNR	48.501259	47.773492	46.993609	48.892712
		SSIM	0.8249	0.7403	0.7339	0.7323
After the attack	pepper and salt noise (0.05)	PSNR	27.943090	27.285805	26..670474	28.331530
		SSIM	0.5482	0.5280	0.6783	0.5264
	Gaussian noise (0.0001)	PSNR	26.946095	26.374115	26.046747	27.296853
		SSIM	0.5228	0.5075	0.5338	0.5040
	cropping (1/16)	PSNR	21.095905	20.075935	21.513170	21.406337
		SSIM	0.5398	0.5318	0.5881	0.5256
	JPEG compression (QF=50)	PSNR	22.249904	23.339515	20.403879	20.323898
		SSIM	0.5193	0.5042	0.5143	0.5301

Table 12: Comparison of average PSNR values of ciphertext images against noise, cropping and JPEG compression

Attack types	[14]	Proposed
Pepper and salt noise	26.5911	27.557725
Gaussian noise	27.2038	26.665953
Cropping	26.4020	21.022836
JPEG compression	27.9637	21.579299

5 Conclusions

In this paper, we have proposed a JPEG image encryption algorithm based on hyperchaotic, mixed hash and dynamic DNA, which improves the anti-attack ability and key space of the encryption algorithm and improves the efficiency and robustness of the existing encryption algorithm. Firstly, the encryption key is generated from the 5D hyperchaotic system, the mixed hash function and the original plaintext image. Secondly, the plaintext image is converted into a DNA matrix according to the DNA coding rules. The DNA encoded image is rearranged in order of the encryption key to achieving pixel permutation. Finally, the encryption key is used to dynamically select DNA operation rules. After relevant calculation, ciphertext images are obtained by decoding the results through DNA coding rules. The experimental results show that the proposed algorithm has large key space, high key sensitivity, high security and robustness, can resist noise, cropping , JPEG compression and other common attacks, and can be applied to secure image communication.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61862041, 61363078). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] A. Babaei, H. Motameni and R. Enayatifar, "A new permutation-diffusion-based image encryption technique using cellular automata and DNA sequence," *Optik*, vol. 203: 164000, 2020.

[2] P. Chaudhary, R. Gupta, A. Singh, P. Majumder and A. Pandey, "Joint image compression and encryption using a novel column-wise scanning and optimization algorithm," *Procedia Computer Science*, vol. 167, pp. 244–253, 2020.

[3] T. Chuman, K. Kurihara and H. Kiya, "On the security of block scrambling-based etc systems against jigsaw puzzle solver attacks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17)*, pp. 2157–2161, June 2017.

[4] G. Z. Cui, L. F. Wang, X. C. Zhang and Z. Zhou, "An image encryption algorithm based on dynamic DNA coding and hyper-chaotic lorenz system," in *International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 226–238, 2018.

[5] R. Enayatifar, A. H. Abdullah, I. F. Isnin, A. Altameem and M. Lee, "Image encryption using a synchronous permutation-diffusion technique," *Optics and Lasers in Engineering*, vol. 90, pp. 146–154, 2017.

[6] M. Ghazvini, M. Mirzadi and N. Parvar, "A modified method for image encryption based on chaotic map and genetic algorithm," *Multimedia Tools and Applications*, vol. 79, no. 37, pp. 26927-26950, 2020.

[7] J. He, S. Huang, S. Tang and J. Huang, "JPEG image encryption with improved format compatibility and file size preservation," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2645–2658, 2018.

[8] M. Kar, A. Kumar, D. Nandi and M. K. Mandal, "Image encryption using DNA coding and hyper-chaotic system," *IETE Technical Review*, vol. 37, pp. 12–23, 2020.

[9] P. L. Li and K. T. Lo, "A content-adaptive joint image compression and encryption scheme," *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 1960–1972, 2017.

[10] P. L. Li and K. T. Lo, "Joint image encryption and compression schemes based on 16×16 DCT," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 12–24, 2019.

[11] Y. P. Li, C. H. Wang and H. Chen, "A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Optics and Lasers in Engineering*, vol. 90, pp. 238–246, 2017.

[12] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[13] Z. Man, J. Li, X. Di and O. Bai, "An image segmentation encryption algorithm based on hybrid chaotic system," *IEEE Access*, vol. 7, pp. 103047–103058, 2019.

[14] B. Mondal, S. Singh and P. Kumar, "A secure image encryption scheme based on cellular automata and chaotic skew tent map," *Journal of Information Security and Applications*, vol. 45, pp. 117–130, 2019.

[15] Open Source Computer Vision Library[CP/OL]. (`https://opencv.org/`)

[16] A. U. Rehman and X. F. Liao, "A novel robust dual diffusion/confusion encryption technique for color image based on chaos, DNA and SHA-2," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 2105-2133, 2019.

[17] M. R. Salman and A. K. Farhan, "Color image encryption depend on DNA operation and chaotic system," in *First International Conference of Computer and Applied Sciences (CAS'19)*, pp. 267–272, 2019.

[18] W. Sirichotedumrong, T. Chuman, S. Imaizumi and H. Kiya, "Grayscale-based block scrambling image encryption for social networking services," in *IEEE International Conference on Multimedia and Expo (ICME'18)*, pp. 1-6, 2018.

[19] V. Thanikaiselvan, S. Patel and S. Sivanantham, "Secured data transmission through dual domain reversible data hiding and encryption in images," in *International Conference on Inventive Computation Technologies (ICICT'20)*, pp. 840–847, Feb. 2020.

[20] USC-SIPI Image Database[DB/OL]. (`http://sipi.usc.edu/database/`)

[21] X. Y. Wang, Y. Wang, X. Q. Zhu and S. Unar, "Image encryption scheme based on chaos and DNA plane operation," *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26111–26128, 2019.

[22] X. Y. Wang, X. Q. Zhu, X. J. Wu and Y. Q. Zhang, "Image encryption algorithm based on multiple mixed hash functions and cyclic shift," *Optics and Lasers in Engineering*, vol. 107, pp. 370–379, 2018.

[23] Z. R. Wang, W. Li, B. L. Zhu and X. Q. Li, "A joint image lossless compression and encryption method based on chaotic map," *Multimedia Tools and Applications*, vol. 76, no. 12, pp. 13995–14020, 2017.

[24] G. D. Ye and X. L. Huang, "A secure image encryption algorithm based on chaotic maps and SHA-3," *Security and Communication Networks*, vol. 9, no. 13, pp. 2015-2023, 2016.

[25] Q. Y. Zhang, J. T. Han and Y. T. Ye, "Image encryption algorithm based on image hashing, improved chaotic mapping and DNA coding," *IET Image Processing*, vol. 13, no. 14, 2019.

[26] X. C. Zhang, Z. Zhou, Y. Niu, Y. F. Wang and L. F. Wang, "An image encryption algorithm based on chaotic system using DNA sequence operations," in *International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 213–225, 2018.

Biography

Zhang Qiu-yu. Researcher/Ph.D. supervisor, graduated from Gansu University of Technology in 1986, and then worked at school of computer and communication in Lanzhou University of Technology. He is vice dean of Gansu manufacturing information engineering research center, a CCF senior member, a member of IEEE and ACM. His research interests include network and information security, information hiding and steganalysis, multimedia communication technology.

Ye Yu-tong. received the BS degrees in communication engineering from Shanghai Normal University, Shanghai, China, in 2016. Her research interests include network

and information security, information hiding and steganalysis.

Blockchain-based Privacy-Preserving Electronic Voting Protocol

Wenqiang Chai[1], Momeng Liu[1], Zeyu Zhang[1], and Liping Lv[2]
(Corresponding author: Momeng Liu)

Shaanxi Key Laboratory of Clothing Intelligence, State and Local Joint Engineering Research Center for Advanced Networking and Intelligent Information Services, School of Computer Science, Xi'an Polytechnic University[1]
58 Shan-gu-road, Lintong, Xi 'an 710600, China
Email:liumomeng@163.com
College of Information Engineering, Zhengzhou Shengda University[2]
No.1, Wen-chang-road, Longhu Town, Zhengzhou City, Henan Province 451191, China
(Received July 11, 2021; Revised and Accepted Jan. 5, 2022; First Online Feb. 26, 2022)

Abstract

The beneficial properties of the blockchain, anonymity, decentralization, and traceability, boost the rapid development of the study on electronic voting schemes. However, with the increase in the transaction fee price, the electronic voting scheme based on blockchain has been gradually unable to meet the demand of voter scale and voting information privacy simultaneously. To solve this problem, we propose a tallying algorithm based on *non-interactive zero-knowledge proof* (NIZK) in this paper, which uses *Groth16 zero-knowledge proof* (ZKP) to hide voters's *identity information* (ID) and *ballot information* (BI). In addition to providing privacy-preserving guarantees, our solution only requires low complexity on the chain so that it can be applied to large-scale voting scenarios. At the end of this paper, we also conducted experiments on Ethereum, and the result shows that our work meets privacy-preserving, low complexity, and can be applied to large-scale elections.

Keywords: Blockchain; Electronic Voting; Privacy-Preserving; Tallying Algorithm; Zero-Knowledge Proof

1 Introduction

Compared with traditional voting, electronic voting has the characteristics of convenience and economy, and has gradually developed into the mainstream of voting. Since *Chaum* proposed an electronic voting protocol based on cryptography in 1981 [4], cryptographic electronic voting schemes have emerged continuously [14, 15]. Electronic voting schemes can be mainly built relying on mixnet [1, 13], full homomorphism [16, 18, 21], blind signature [8, 9, 12] and ring signature [5, 11, 22]. But these schemes cannot meet the demand of voter scale and voting privacy at the same time. The electronic voting scheme based on the mixnet is computationally complex, inefficient, and insecurity. Although the electronic voting scheme based on full homomorphism can protect voting information, but the complexity is higher and unpractical. The electronic voting scheme based on ring signatures and blind signatures requires a trusted third-party central organization and lacks voting information privacy-preserving.

Due to the outstanding characteristics of blockchain, it can be naturally combined with voting applications. For the past few years, electronic voting schemes based on blockchain have been proposed [17, 24–27]. The work of McCorry proposed a blockchain electronic voting scheme that maximizes voter privacy in 2017 [17], but all encrypted votes must be submitted before votes can be tallied. Therefore, the system lacks certain robustness. The electronic voting scheme based on the ring signature proposed in [27] needs complex calculations and consumes high transaction fees for ring signature verification on the chain, which is unpractical. In [24], The paper proposed a secure electronic voting scheme based on Ethereum. However, the scheme generates and verifies the ZKP voting information on the chain, which requires same high transaction fees. The paper of [25], Publicly Verifiable Online Voting Protocol Without Trusted Tallying Authorities, hides BI based on ElGamal encryption that has the additive homomorphism that can be used to tallying. In addition, the author adopts group encryption to reduce tallying time-consuming but ignored verification consuming-time. Experiments show that in the voting scenario of about a thousand voters, the verification time will takes nearly one minute. In [26], an electronic voting system is implemented based on Ethereum. However, in the scenario of 6 candidates and 5 voters, the experiment shows that nearly 2.25 million gas expenses even though without the hiddings of BI and ID. This is inappropriate in current high transaction fees.

Thus, this papper is motivated by this: Can we design a electronic voting protocol with privacy-preserving, low complexity? To answer this question, we propose a blockchain-based privacy-preserving electronic voting protocol and our specific contributions are in the following:

1) We propose a tallying algorithm based on homomorphic encryption and can greatly reduce the complexity for tallying and calculations.
2) We built a ZKP voting protocol based on the Zokrates toolbox, which can protect the voter's BI and ID.
3) Experiments show that the protocol meets privacy-preserving, low complexity, and can be applied to large-scale elections.

2 Preliminaries

In this section, we will introduce some backgrounds including Blockchain, smart contract, ZKP, Zokrates toolbox and Paillier public key cryptosystem.

2.1 Blockchain

Since *Nakamoto* released the Bitcoin white paper in 2009 [19], electronic money has had a great impact on the traditional financial industry. Blockchain is a peer-to-peer electronic cash system that does not rely on third-party institutions. It uses cryptography and consensus mechanisms to achieve immutability and integrity on the chain. Anyone can participate in blockchain transactions, and every transaction is transparent and traceable.

Electronic voting has a strong dependence on third-party trust institutions, and single point of failure and the reveal of voting information are existing. The characteristics of the blockchain can be naturally combined with electronic voting, which makes up for some shortcomings for traditional electronic voting. The blockchain relies on the consensus protocol to ensure the integrity of the vote information. The decentralized system guarantees the transparency of the vote information. The distributed architecture guarantees that the system can operate normally even encountering single point of failure.

2.2 Smart Contract

Smart contract is also called Blockchain 2.0, which is similar to traditional programs, but different from traditional programs. It is deployed on the blockchain and executed spontaneously by event-driven and the triggering process is accompanied by transactions. Some entity operation processes under the smart contract are visible to people, and the underlying consensus mechanism ensures the consistency and correctness of the transaction results [10]. Therefore, smart contracts can be applied to some more complex application scenarios, not just limited to currency transactions, and can deploy and run some distributed applications, such as crowdfunding and medical care. At present, the two blockchain platforms that are widely used for smart contracts are Ethereum [23] and IBM's Hyperledger [2] . The bottom layer of Ethereum is a Turing-complete Ethereum virtual machine that supports the Solidity language to complete smart contracts compile and run.

2.3 ZKP

ZKP was proposed by [6] in the 1980s. In cryptography, ZKP is a method that one party can prove to the other party that they know the value of x without revealing any information. The essence of ZKP is that it is easy by simply revealing the knowledge that someone has certain information. The challenge is to prove this possession without revealing the information itself or any other information. NIZK also known as Zero-Knowledge Sufficient *Non-interactive Arguments of Knowledge*(zk-SNARK) proposed by [3] in 2014 or *Zero-Knowledge Scalable Transparent ARguments of Knowledge* (zk-STARK) proposed in 2018. Facts have proved that ZKP has a broad application space and can effectively solve many problems. For example, ZKP can be applied to the Zcash transaction process that miners believe that the transaction between the two parties is valid without revealing the addresses and transaction amount of both parties.

ZKP are divided into two types, interactive and non-interactive based on whether there is an interactive process between the verifier and the prover. zk-SNARK have been used on Zcash and this paper will use zk-SNARK to protect the voting information.

2.4 Zokrates Toolbox

Zokrates is a model proposed by [7] in 2018 based on the Groth16 algorithm to calculate online verification on Ethereum. It is a ZKP toolbox composed of *Domain-Specific Language* (DSL), compiler, proof and witness generator. In addition, it is worth nothing that this toolbox can directly export the verification information as a smart contract for deployment on the blockchain.

Therefore, we can use the Zokrates toolbox to complete the ZKP work of this paper. The workflow is shown in Figure 1. The specific details are as follows: First, We convert the problem that needs to be proved into a specific calculation formula and write it into the DSL file. Then we compile the DSL file into flattened code, which is an abstract form of limiting the circuit. The purpose of this compilation process is to convert DSL files into *Rank-1-Constraint-System*(R1CS). Therefore, ZoKrates can be combined with the zk-SNARK verification stage. Similar to the zk-SNARK algorithm, *Common Reference String*(CRS) is shared during the establishment phase. Then flattened code generate a long proving key and a short verification key. The verification key will be deployed on the blockchain along with the verifica-

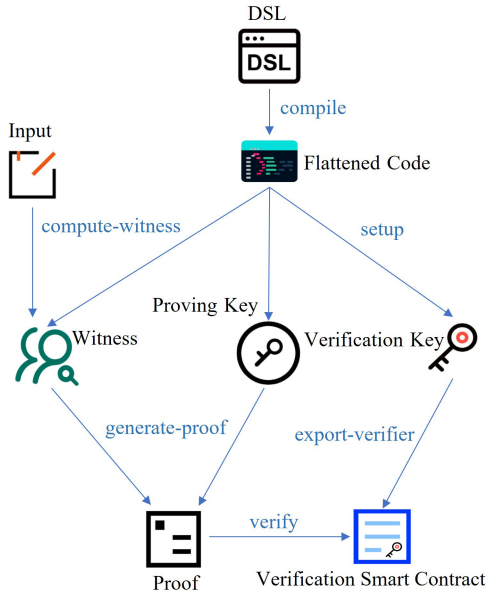


Figure 1: The workflow of Zokrates

tion smart contract, and the verification key will be sent to every prover. Before generating the ZKP, the prover needs to fill in the relevant input to satisfy the flattened code and generate the relevant witness.

Finally, Zokrates calculates proof based on the verification key and witness and the proof can be verified by the verification information stored in the smart contract.

2.5 Paillier Public Key Cryptosystem

Paillier cryptosystem was proposed by [20]. It is a semantically secure probabilistic asymmetric algorithm. The algorithm is based on complex residual difficult problems and satisfies the homomorphism of addition and multiplication. In this paper, only addition homomorphisms are used for tallying, precisely:

$$E(x+y) = E(x) \oplus E(y)$$

Paillier key generation algorithm:

1) Randomly select two large prime number p and q, and satisfy $gcd(p \cdot q, (p-1)(q-1)) = 1$, this property guarantees that p and q are equal in length.
2) Calculate $n = p \cdot q$ and $\lambda = lcm(p-1, q-1)$.
3) Choose a random integer $g(g \in Z_{n^2}^*)$, such that the order of n divided by g is satisfied.
4) Definition $L(x) = \frac{x-1}{n}$.
5) Calculate $\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n$.
6) The public key is (n, g).
7) The private key is (λ, μ).

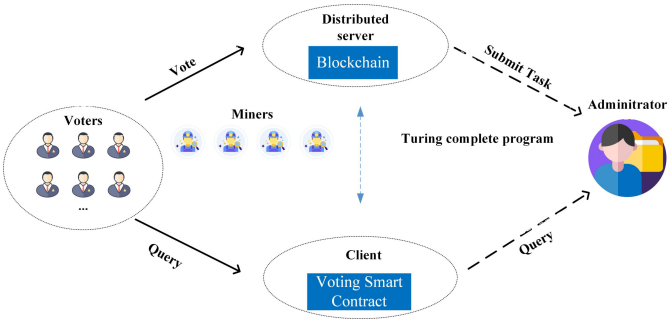


Figure 2: System role model

Paillier encryption algorithm:

1) m is plaintext $(0 \leq m < n)$.
2) Choose random number $r(0 < r < n)$, such that $gcd(r, n) = 1$.
3) $c = g^m * r^n \bmod n^2$.

Paillier decryption algorithm:

1) $m = L(c^\lambda \bmod n^2) * \mu \bmod n$.

3 System Model

This section will describe the system model of electronic voting based on blockchain. Based on this model we will assume some threat models and some technical challenges. Before we start, let us look at the related roles and explanation of notations, as shown in Figure 2 and Table 1.

As shown in Figure 2, there are four entities in the electronic voting system: voting administrators, voters, candidates and miners.

Administrator. Responsible for initialization of the elections, deployment of smart contracts, submission of voting tasks to the blockchain.

Voters. Responsible for submission of personal vote information to the blockchain, represented by $V = V_1, \cdots, V_i, \cdots, V_s$.

Candidates. Indicated by $C = C_1, \cdots, C_i, \cdots, C_m$.

Miners. Responsible for package of transactions on the blockchain.

3.1 Threat Model

Potential malicious administrators and voters will use special methods to maximize their interests. First, we introduce related threat models, including potential threats and malicious behaviors.

Malicious administrators. Since malicious administrators have the authority to deploy and maintain smart contracts, they can obtain more ID and BI of voters compared to other roles. Moreover, the administrator has the

Table 1: The explanation of notations

Notation	Explanation
$V = V_1, \cdots, V_i, \cdots, V_s$	Set of voters,$i = 1, \cdots, s$
$C = C_1, \cdots, C_j, \cdots, C_m$	Set of Candidates,$j = 1, \cdots, m$
$M = M_1, \cdots, M_i, \cdots, M_s$	Set of voting plaintext informations
$E = E_1, \cdots, E_i, \cdots, E_s$	Set of voting ciphertext informations
$H(M_x)$	The hash value of plaintext M_x
(n, g),(λ,μ)	Paillier public key , private key
$S_m^n(1 \leq n)$	Select m from n candidates
$proofG_{ID}$	The ID proof
$proofV_{ID}$	The contract of ID verification
$proofG_{BI}$	The BI proof
$proofV_{BI}$	The contract of BI verification

Paillier private key to decrypt the voting information, and has a certain channel to intercept and view the voting information.

Malicious voters. Voters have voting rights. In order to maximize their own interests, voters's malicious behavior is that the total number of votes is greater than the number of candidates in $S_m^n(1 \leq n)$.

3.2 Security Assumption

Groth16. We assume that the Groth16 algorithm is secure. In the stage of generate-proof according Figure 1, it is difficult to obtain any useful information about witness from the proof information and the administrator is no exception.

Paillier. We assume that the Paillier homomorphic encryption algorithm is secure. The voter encrypts the BI and transmits it on the public channel. Except for the administrator with the Paillier private key, which can decrypt and view the BI. It is difficult for others to see the plaintext of the BI without the Paillier private key.

Majority honest miners. Voting information is stored on the blockchain. It is important to ensure the security of the blockchain. The security of the blockchain depends on the miners. The blockchain is decentralized, and the attacker can control the entire blockchain network only if he has more than 50% of the blockchain computing power. Here, we assume that most miners are honest.

3.3 Technical Challenges

The BI of voters is malicious. For example, the number of votes of a certain voter for a certain candidate is greater than the specified value, but the voting information has been encrypted. In this case, it is a technical challenge to perform ZKP on the voting information to ensure the qualification of the voting information.

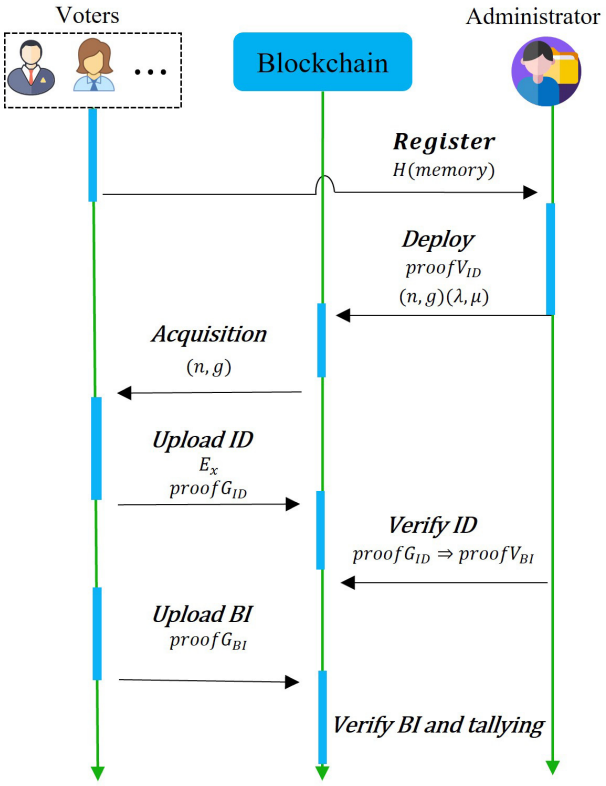


Figure 3: Election workflow

4 The Proposed Privacy-preserving Electronic Voting Protocol

The design of the privacy-preserving protocol in this paper is based on the technical challenges proposed above, and is mainly divided into three parts. First, tallying algorithm based on homomorphic encryption is proposed to deal with the general voting scenario $S_m^n(1 \leq n)$. Secondly, the ZKP protocol is designed to hide the ID and BI of voters. Finally, the specific voting process is described, as shown in Figure 3.

4.1 Tallying Algorithm Based on Homomorphic Encryption

The tallying algorithm proposed in this paper is mainly to design candidate index, which are based on the number of voters. Assuming that the number of voters is s, our structure is as follows:

$$\begin{aligned} C_1 &= 1 \\ C_2 &= s+1 \\ C_3 &= s^2+1 \\ &\vdots \\ C_m &= s^{m-1}+1. \end{aligned}$$

Table 2: Comparison of gas costs [17] and this paper

Plan	Ballot type	Gas cost(unit)				
	Timeline	Create ballot	Register	Vote	Total	Initial contract to the blockchain
[17]	3 Candidates and 3 voters	261340	424536	380176	1066052	2240889
	3 Candidates and 5 voters	261340	677560	805127	1744027	
this paper	3 Candidates and 3 voters	0(off-chain)	510084	282300	792384	3340684
	3 Candidates and 5 voters	0(off-chain)	740023	403400	1143423	

Suppose the voter is V_x and he supports all the candidates, his plaintext BI can be expressed as:

$$M_x = \sum_{i=1}^{m} C_i = s^{m-1} + \cdots + s^2 + s + m$$

Using Paillier encryption:

$$E_x = g^{s^{m-1}+\cdots+s^2+s+m} * r^n \bmod n^2$$

When $E_1, E_2, \cdots, E_s$ uploaded to the blockchain, according to the property of homomorphic encryption:

$$\sum_{x=1}^{s} E_x = g^{\sum_{x=1}^{s} M_x} * r^n \bmod n^2$$

Using Paillier decryption:

$$\sum_{x=1}^{s} M_x = g^{\sum_{x=1}^{s} M_x} * r^n \bmod n^2$$

Now, assuming $x = 1$ and he supporting all candidates, then M_x is:

$$\sum_{x=1}^{s} M_x = M_1 = \sum_{i=1}^{m} C_i = s^{m-1} + \cdots + s^2 + s + m$$

According to M_x, we can get the results of V_1 voting for all candidates, supposing the number of votes each candidate gets is C_{by}, as shown in the Algorithm 1.

Algorithm 1 Tallying

Input $M_x, C_1, C_2, \cdots, C_m$
Output $C_{b1}, C_{b2}, \cdots, C_{bm}$
if $M_x \neq 0$ **then**
  $y \leftarrow 1$
  $list \leftarrow$ null
  while $M_x \neq 0$ and $y \leq m$ **do**
    $C_{by} \leftarrow M_x$ divide by C_y
    $M_x \leftarrow M_x$ mod C_y
    $y \leftarrow y + 1$
    $list(y) \leftarrow C_{by}$
  end while
  return *list*
end if

4.2 The Hiding of Voting Information Based on ZKP

Here voting information includes ID and BI, the specific ZKP hidden information is as follows:

4.2.1 The Hiding of ID Based on ZKP

Zokrates needs to transform the actual problem into mathematical model, and compile and process it in a DSL form. In order not to reveal personal information, voters can prove that they are a member V_i in $V_1, \cdots, V_i, \cdots, V_s$, the DSL language can be expressed as $\prod_{j=1}^{s} V_i - V_j \equiv 0$. In this paper, the voters upload $H(memory_x)$ to verify their identity, so they need to input the correct $memory_x$ satisfy the following equation.

$$\prod_{j=1}^{s} H(memory_x) - H(memory_j) \equiv 0$$

4.2.2 The Hiding of BI Based on ZKP

Voters must prove that their votes M_x are within the effective range without revealing their BI. Suppose there are two candidates C_1, C_2 and each voter can support two candidates at most. For M_x, it should be satisfy:

$$(\prod_{i=1}^{2} M_x - C_i)(M_x - \sum_{i=1}^{2} C_i) \equiv 0$$

And it's necessary to prove whether the input M_x is the upload ID in Figure 3. So the voter needs to input the correct M_x, r_x to satisfy:

$$\prod_{i=1}^{s} g^{M_x} * r_x^n \bmod n^2 - E_i \equiv 0$$

The mathematical model is shown in Algorithm 2.

Algorithm 2 The Hiding of BI Mathematical Model

Input M_x, r_x
Output *true or false*
$i \leftarrow 1$
$BI_{flag} \leftarrow false$
while $i \neq$ s **do**
  if $g^{M_x} * r_x^n \bmod n^2 = E_i$ ***then***
    $BI_{flag} \leftarrow true$
  end if
end while
$M_{flag} \leftarrow false$
if $M_x(\prod_{i=1}^{2} M_x - C_i)(M_x - \sum_{i=1}^{2} C_i) = 0$ **then**
  $M_{flag} \leftarrow true$
end if
return BI_{flag} & M_{flag}

4.2.3 The Process of Electronic Voting Protocol

Registration. Voters submit voting applications off-line. If the administrator records the voter's Ethereum account address to prove voter's identity, the ID may be leaked during the channel transmission of the voting results, which is not secure against channel attacks. In order to reduce the complexity of the system, we adopt the way of memonics. Voters can write down some fixed words that they can remember, and then get hash for it, and submit the Hash value to the administrator off-line as a proof of voting identity.

Deployment. After obtaining all $H(memory)$, the administrator generates flattened code ,*proving key*, $proofV_{ID}$, according to the Zokrates tool. The $proofV_{ID}$ smart contract and Paillier public key (n, g) and private key (λ,μ) will be deployed to the blockchain by the administrator. The flattened code and proving key will also be uploaded to the blockchain. But it should be noted that the Paillier private key is not available on the blockchain by voters.

Acquisition. Voters obtain Paillier public key (n, g) , flattened code and proving key from the blockchain.

Upload ID. Voters generate E_x according to the plaintext information M_x of the ballot and the Paillier public key (n, g), and generate $proofG_{ID}$ according to the flattened code, proving key and $memory_x$ and upload it to the blockchain.

Verify ID. After the blockchain has received all E_x and verifies $proofG_{ID}$ by $proofV_{ID}$, the administrator will generate $proofV_{BI}$ that is deployed on the blockchain according Algorithm 2. Then upload the flattened code and proving key to the blockchain.

Upload BI. Voters obtain the flattened code and proving key. Then generate $proofG_{BI}$ according to flattened code and upload it to the blockchain.

Verify BI. The blockchain verifies the $proofV_{BI}$ according to $proofG_{BI}$ .If the verification is passed, it will automatically tally the ticket and publish the result of the vote on the blockchain.

5 Experiment and Analysis

This paper uses the local Ganache blockchain as a decentralized database, and uses the Truffle framework and Web.js to interact with smart contracts on the chain. The experimental environment are the 64-bit operating system Mac OS. The memory is 16GB 2400MHz, and the CPU is Intel Core i7 2.2GHz.

The purpose of the experiment is to test the efficiency of generation and verification of voter ID and BI and tallying in the scene of large-scale elections.

Before this experiment, there are 3 candidates who use the blockchain to deploy smart contracts to assess the generation and verification time of the ZKP of ID and BI, and the time for ballot tallying. The experiment takes the number of voters as an argument, and tests the time-consuming generation and verification. Finally, the average tallying time was calculated in the case of different numbers of voters.

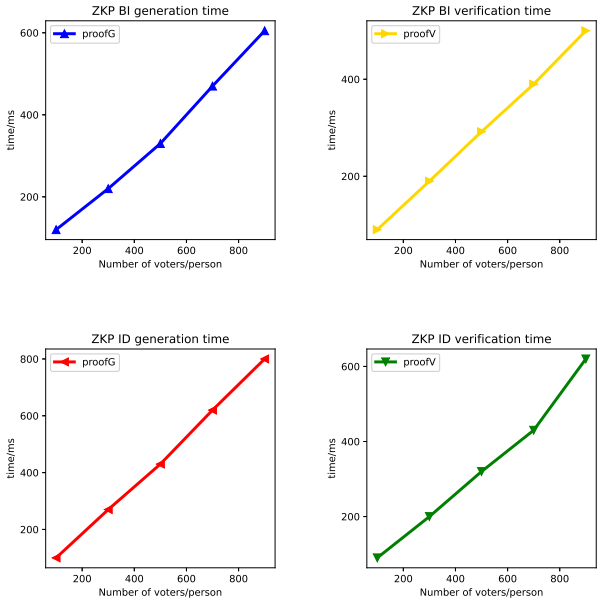


Figure 4: the time of BI,ID generation and verification

In the experiment, the smart contract complied with the scheme proposed in this paper, and realized the voting process without a trusted third-party. The experimental results are shown in Figure 4. With the increase in the scale of the election, there is a related increase in ID and BI generation time and verification time. However, the tallying time is shown in Figure 5. The tallying algorithm has greatly reduced the tallying time-consuming compared to [24], the increment of time-consuming always in the low speed range. The ZKP verification time-consuming is shown in Figure 6. For the merits of off-chain computing of Zokrates, with the increase of voters, the average verification time of ZKP's ID and BI is greatly reduced compared to [25]. In terms of gas-consuming, the paper will be compared with [26]. As shown in Table 2, in the case of 3 candidates, 3 voters and 5 voters. Due to the advantages of Zokrates off-chain computing, the gas-consuming is 0 in ballot creation stage. In registration stage and the contract initialization stage, as this paper needs to do some preparations for ZKP, the gas-consuming is slightly larger than that of [26]. In the voting stage, on account of the low-complexity verification of $proofG_{ID}$ and $proofG_{BI}$ and tallying algorithom based on homomorphic encryption, the amount of gas-consuming used is less than [26].

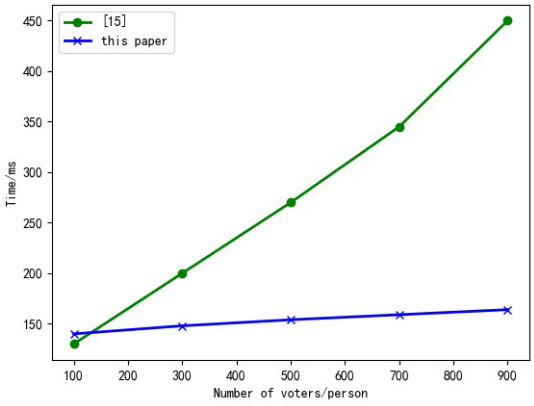


Figure 5: Comparison of tallying time-consuming [15] and this paper

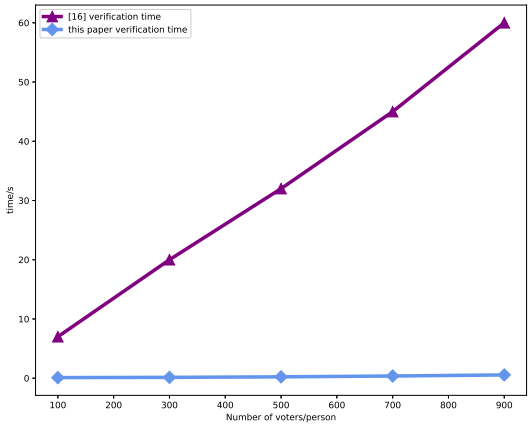


Figure 6: Comparison of ZKP verification time-consuming [16] and this paper

6 Conclusions

In response to the two technical challenges, this paper designs a privacy-preserving electronic voting scheme based on blockchain. Through the ZKP Groth16 algorithm and the off-chain computing on-chain verification framework Zokrates and Paillier homomorphic encryption, the ZKP for the hiding of voter ID and BI is realized. Experiments shows that the proposed tallying algorithm greatly reduces the calculation complexity on blockchain and can be applied to large-scale elections.

Acknowledgments

Acknowledgements: This work is supported by the National Natural Science Foundations of China (Grant No.61902303), the Natural Science Basic Research Program of Shaanxi (Program No.2020JQ-832), the Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No.21JK0651), the Young Talent fund of University Association for Science and Technology in Shaanxi, China (Program No.20210116), the Shaanxi Key Laboratory of Blockchain and Secure Computing, and the Research Foundation of Xi'an Polytechnic University for the Doctoral Scholars (Program No.BS201848).

References

[1] R. Aditya, B. Lee, C. Boyd, and E. Dawson. An efficient mixnet-based voting scheme providing receipt-freeness. In *International Conference on Trust, Privacy and Security in Digital Business*, pages 152–161. Springer, 2004.

[2] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, and J. Yellick. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*, pages 1–15, 2018.

[3] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza. Succinct non-interactive zero knowledge for a von neumann architecture. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 781–796, 2014.

[4] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[5] E. Fujisaki and K. Suzuki. Traceable ring signature. In *International Workshop on Public Key Cryptography*, pages 181–200. Springer, 2007.

[6] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.

[7] J. Groth. On the size of pairing-based non-interactive arguments. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 305–326. Springer, 2016.

[8] M. S. Hwang, C. C. Lee, and Y. C. Lai. Traceability on stadler et al.'s fair blind signature scheme. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, 86(2):513–514, 2003.

[9] M. S. Hwang, C. C. Lee, and Y. C. Lai. An untraceable blind signature scheme. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 86(7):1902–1906, 2003.

[10] S. Jani. Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought,(16)*, 18(2), 1996.

[11] A. Kugusheva and Y. Yanovich. Ring signature-based voting on blockchain. In *Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*, pages 70–75, 2019.

[12] M. Kumar, C. Katti, and P. C. Saxena. A secure anonymous e-voting system using identity-based blind signature scheme. In *International conference on information systems security*, pages 29–49. Springer, 2017.

[13] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. Providing receipt-freeness in mixnet-based voting protocols. In *International conference on information security and cryptology*, pages 245–258. Springer, 2003.

[14] C. T. Li and M. S. Hwang. Security enhancement of chang-lee anonymous e-voting scheme. *International Journal of Smart Home*, 6(2):45–52, 2012.

[15] C. T. Li and M. S. Hwang. A secure and anonymous electronic voting scheme based on key exchange protocol. *International Journal of Security and Its Applications*, 7(1):59–70, 2013.

[16] G. Liao. Multi-candidate electronic voting scheme based on fully homomorphic encryption. In *Journal of Physics: Conference Series*, volume 1678, page 012064. IOP Publishing, 2020.

[17] P. McCorry, S. F. Shahandashti, and F. Hao. A smart contract for boardroom voting with maximum voter privacy. In *International Conference on Financial Cryptography and Data Security*, pages 357–375. Springer, 2017.

[18] L. Morris. Analysis of partially and fully homomorphic encryption. *Rochester Institute of Technology*, pages 1–5, 2013.

[19] S. Nakamoto and A. Bitcoin. Bitcoin: a peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[20] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer, 1999.

[21] P. B. Rønne, A. Atashpendar, K. Gjøsteen, and P. Y. Ryan. Coercion-resistant voting in linear time via fully homomorphic encryption: towards a quantum-safe scheme. *arXiv preprint arXiv:1901.02560*, 2019.

[22] B. Wang, J. Sun, Y. He, D. Pang, and N. Lu. Large-scale election based on blockchain. *Procedia Computer Science*, 129:234–237, 2018.

[23] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[24] Z. Wu, Z. Cui, T. Liu, and H. Pu. Secure electronic voting scheme based on blockchain. *Journal of Computer Applications in Chinese*, 40(7):1989–1995, 2020.

[25] X. Yang, X. Yi, S. Nepal, A. Kelarev, and F. Han. Blockchain voting: publicly verifiable online voting protocol without trusted tallying authorities. *Future Generation Computer Systems*, 112:859–874, 2020.

[26] X. Yang, X. Yi, S. Nepal, A. Kelarev, and F. Han. Votereum: an ethereum-based e-voting system. In *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–6. IEEE, 2019.

[27] J. Zhengn and H. Lai. Blockchain e-voting scheme based on one-time ring signature. *Application Research of Computers in Chinese*, 37(11):187–197, 2020.

Biography

Wenqiang Chai is a master student of the School of Computer Science, Xian Polytechnic University, China. His current research interests is designing privacy-preserving protocols based on Blockchain.

Momeng Liu is a lecturer in the School of Computer Science, Xi'an Polytechnic University, China, and a member of Shanxi key Laboratory of Clothing Intelligence. In 2018, she earned her Ph.D. degree in cryptography from Xidian University, China. Her research interests mainly focus on designing protocols built upon lattice-based cryptography and providing privacy-preserving solutions in Blockchain-based scenarios.

Zeyu Zhang is a master student of the School of Computer Science, Xian Polytechnic University, China. Her current research interests is designing privacy-preserving protocols based on Blockchain.

Liping Lv received the bachelor degree in computer science from Qingdao Technological University in 2001, the mater degree in computer science from Qingdao University of Science and Technology in 2005. She is currently a full professor with the College of Information Engineering from Zhengzhou Shengda University, China. Her research directions include internet of things, artificial intelligence and automatic control.

Analysis of One Secure Key Agreement and Key Protection for Mobile Device User Authentication

Lihua Liu[1], Leming Hong[1], and Zhengjun Cao[2]
(Corresponding author: Zhengjun Cao)
Department of Mathematics, Shanghai Maritime University[1]
No.1550, Haigang Ave, Shanghai, China
Department of Mathematics, Shanghai University[2]
No.99, Shangda Road, Shanghai 200444, China
Email: caozhj@shu.edu.cn
(Received Oct. 30, 2020; Revised and Accepted July 6, 2021; First Online Feb. 26, 2022)

Abstract

We show that the scheme [IEEE TIFS, 14(2), 319-330, 2019] has some shortcomings. (1) The running modulus is too big (4096 bits at least) for most mobile devices to finish the related computations. (2) The auxiliary device acts negatively, which does not reduce the master device's computational burden but instead incurs more cost. (3) Its security argument is flawed because the user's public key P_{ID_i} is never used in the scheme, while the parameter repeatedly appears in its latter security argument.

Keywords: Anonymity; Authentication; Auxiliary Device; Master Mobile Device

1 Introduction

There are many authentication and key agreement protocols for mobile devices. In 2015, Tsai *et al.* [14, 21] presented an authentication scheme for distributed mobile cloud computing services. Gope and Hwang [8, 9] discussed the techniques for lightweight and energy-efficient mutual authentication and key agreement with user anonymity in global mobility networks. Cahyani *et al.* [2,11,18] investigated the problems about forensic data acquisition from cloud-of-things devices and anonymous authentication for wireless body area networks.

In 2019, Chiou *et al.* [7] pointed out that a mutual authentication scheme was insecure. Hsien *et al.* [4,5] have presented some surveys on public auditing for secure data storage in cloud computing. Wang *et al.* [13,22] presented a survey for reversible data hiding for VQ-compressed images. Pan *et al.* [16,17,20] put forth some batch verification schemes for identifying illegal signatures, smart card-based password authentication schemes, and data collaboration scheme with hierarchical attribute-based encryption in cloud computing. In 2020, Huang *et al.* [12] carried out the work on malware detection and classification based on artificial intelligence. Cao and Markowitch [3] proved that a general circuit ciphertext-policy attribute-based hybrid encryption with verifiable delegation in cloud computing was insecure. Chen *et al.* [6] discussed the topic of secure financial surveillance blockchain systems.

Very recently, Wu *et al.* [23] have presented a key agreement scheme for mobile device user authentication. It combines the Paillier encryption [15], a variation of Schnorr zero knowledge proof [19], the key generation techniques [1], the hashed Diffie-Hellman key agreement, and a default message authentication code (MAC). In this note, we want to stress that the auxiliary device does not function positively. The underlying modulus is too big to run for most mobile devices. Besides, the proposed security argument fails to keep its logical consistency.

2 Review of the Scheme

The scheme needs to use the Paillier encryption [15], which can be described as follows. Let $\bar{p}, \bar{q}$ be two primes, $n = \bar{p}\,\bar{q}$, and $\lambda = \text{lcm}(\bar{p}-1, \bar{q}-1)$. Pick $g \in \mathbb{Z}_{n^2}$ such that $n \,|\, \text{ord}_{n^2}(g)$. Set n, g as the public key, and λ as the private key. For $m \in \mathbb{Z}_n^*$, pick $r \in \mathbb{Z}_n$ to compute the ciphertext $c = g^m r^n \bmod n^2$. Given the ciphertext c, compute its plaintext

$$m = \left(\frac{c^\lambda - 1 \bmod n^2}{n}\right) / \left(\frac{g^\lambda - 1 \bmod n^2}{n}\right) \bmod n.$$

Let $\text{Enc}_{pk}(\cdot), \text{Dec}_{sk}(\cdot)$ be the encryption and decryption transformation, respectively. It is easy to check that $\text{Dec}_{sk}(\text{Enc}_{pk}(m_1) \cdot \text{Enc}_{pk}(m_2)) = m_1 + m_2 \bmod n$.

There are three entities: the master mobile device DA_i, the auxiliary device DB_i, and the server S (see Figure 1).

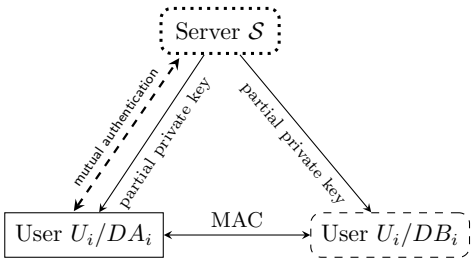


Figure 1: The system model

Setup. The server S with the identity ID_S chooses two primes p, q, and a generator P of a cyclic addition group $\mathbb{G}$ with the order q over a non-singular elliptic curve E. Choose five hash functions $h_1, \cdots, h_5$, and set a random $s \in \mathbb{Z}_q^*$ as the master key. The public parameters are set as $\{G, P, q, h_1, \cdots, h_5, P_{pub} = sP\}$.

Registration. The user U_i sends the true identity ID_i to S. The server S picks $r_{ID_i} \in \mathbb{Z}_q^*$ and computes

$$D_{ID_i}^{(1)} = \frac{r_{ID_i}}{s + h_1(ID_i)} P, \quad D_{ID_i}^{(2)} = r_{ID_i}^{-1} \bmod q.$$

Then generate a key pair (sk_i, pk_i). Load $(D_{ID_i}^{(1)}, sk_i, pk_i)$ on DA_i, and $(D_{ID_i}^{(2)}, pk_i)$ on DB_i. The server S computes $P_{ID_i} = (h_1(ID_i) + s)P$ and sets it as U_i's public key.

Authentication. See Figure 2.

3 Analysis

Though the scheme is interesting, we find it is flawed.

- *The practical modulus is very large, and most mobile devices cannot efficiently finish the related computations.* In the Paillier encryption, the RSA modulus n is generally set as a 2048-bit composite number. So, the running modulus n^2 should be of 4096 bits at least. Even for a PC, the modular reduction with such a big modulus is quite inefficient, because one modular exponentiation almost takes 0.0156 second (on PC, Intel(R) Core(TM) i7-479 CPU 3.60GHz, RAM 4.00GB).

  Note that the working parameter for RSA is of 2048 bits. Moreover, RSA is only used for encrypting session keys (invoked by the subsequent symmetric key encryption, such as AES), instead of any practical message.

- *The auxiliary device does not reduce the master device's computational burden, instead incurs more computational cost.* As usual, p, q are set as 512-bit, 160-bit primes, respectively (see [23]). In the authentication phase, the computational tasks are listed in Tables 1 and 2.

Table 1: Auxiliary device's computational cost

$(C_1^{r_2 D_{ID_i}^{(2)}}) \cdot \mathrm{Enc}_{pk}(\rho\, q)$	$O(\log^3 n)$
Checking π_1 and generating π_2	
requires three point multiplications,	$\hat{O}(\log q \log^2 p)$
one Paillier encryption and decryption	$O(\log^3 n)$

Table 2: Master device's computational cost

$\mathrm{Enc}_{pk}(r_1)$, $\mathrm{Dec}_{sk}(C_2) \cdot D_{ID_i}^{(1)}$	$O(\log^3 n)$
Generating π_1 and checking π_2	
requires three point multiplications,	$\hat{O}(\log q \log^2 p)$
one Paillier encryption and decryption	$O(\log^3 n)$
Other three point multiplications	$\hat{O}(\log q \log^2 p)$

Roughly speaking, $\hat{O}(\log q \log^2 p) \approx O(\log^2 n)$, because the elliptic curve discrete logarithm problem with parameters p (512-bit) and q (160-bit) has the same security level as the RSA problem with a 2048-bit modulus n. Thus, the introduced auxiliary device does not truly alleviate the master device's computational burden. Compared with the plain authentication scheme (Figure 3), the proposed mechanism is very inefficient.

- *The partial private key generation is just based on the simple observation:*

$$D_{ID_i} = \frac{1}{s + h_1(ID_i)} P \Longrightarrow \begin{cases} D_{ID_i}^{(1)} = \frac{r_{ID_i}}{s + h_1(ID_i)} P \\ D_{ID_i}^{(2)} = r_{ID_i}^{-1} \end{cases}$$

The resulting parameter dependent on the assigned key is eventually expressed as $\varphi = \frac{\hat{r}}{s + h_1(ID_i)} P$ for some random $\hat{r} \in \mathbb{Z}_q^*$. That means the Wu *et al.*'s scheme and the plain scheme are of the same security level. The plain scheme, however, needs only $O(\log^2 n)$ cost for the main mobile device, not $O(\log^3 n)$, and its running environment is equivalently restricted to the modulus n, not n^2.

- *Its security argument is not sound.* The user's public key P_{ID_i} is not invoked at all, while it appears repeatedly in the latter argument (see Section V, [23]). So, the security argument fails to keep its logical consistency.

  By the way, the resulting session key SK_U is generally forwarded to the powerful auxiliary device via the default message authentication code (MAC), and it will complete the subsequent data exchange with the server, not the master mobile device itself.

- *The scheme is not of the so-called two-factor security,* i.e, an adversary, who would control a single device of a target user U (whose master device is DA_i and auxiliary device is DB_i), cannot impersonate U to

Auxiliary device $\{ID_i, D_{ID_i}^{(2)}, pk\}$	Master device $\{ID_i, D_{ID_i}^{(1)}, pk, sk\}$	Server $\{ID_i, ID_S, s\}$
	Pick $r_1 \in \mathbb{Z}_q^*$, set $C_1 = \mathrm{Enc}_{pk}(r_1)$, $R_1 = r_1 P$, a proof π_1 for R_1, C_1.	
Check π_1. If true, pick $r_2, \rho \in \mathbb{Z}_q^*$.	$\xleftarrow{R_1, C_1, \pi_1}$	
Set $C_2 = (C_1^{r_2 D_{ID_i}^{(2)}}) \cdot \mathrm{Enc}_{pk}(\rho\, q)$, $R_2 = r_2 P$ and a proof π_2 for R_2, C_2.		
$\xrightarrow{R_2, C_2, \pi_2}$	Check π_2. Pick $r_3 \in \mathbb{Z}_q^*$, compute $\varphi = \mathrm{Dec}_{sk}(C_2) \cdot D_{ID_i}^{(1)}$, $R = r_1 R_2$, $\alpha = h_3(R, \varphi, ID_i, T_1)$, $R_3 = r_3 P$, $w = r_3 P_{pub}$, $AID_i = ID_i \oplus h_2(R_3, w)$.	
	$\xrightarrow{\textbf{Auth1}=(AID_i, R, R_3, \alpha, T_1)}$	Compute $w' = sR_3$, $ID_i' = AID_i \oplus h_2(R_3, w')$, $\varphi' = \frac{1}{s+h_1(ID_i')} R$, $\alpha' = h_3(R, \varphi', ID_i', T_1)$. If $\alpha' = \alpha$, pick $r_4 \in \mathbb{Z}_q^*$, and set $R_4 = r_4 P$, $\hat{R} = r_4 R_3$, $\beta = h_4(R, w', \varphi', T_2)$, $SK_S = h_5(ID_i', ID_S, w', \varphi', \hat{R})$.
	Compute $\beta' = h_4(R, w, \varphi, T_2)$. If $\beta' = \beta$, compute $\hat{R}' = r_3 R_4$, $SK_U = h_5(ID_i, ID_S, w, \varphi, \hat{R}')$.	$\xleftarrow{\textbf{Auth2}=(R_4, \beta, T_2)}$

Figure 2: The mutual authentication in Wu *et al.*'s scheme

Auxiliary device $\{ID_i\}$	Master device $\{ID_i, D_{ID_i} = \frac{1}{s+h_1(ID_i)} P\}$	Server $\{ID_i, ID_S, s\}$
	Pick $r_1, r_3 \in \mathbb{Z}_q^*$, set $R = r_1 P$, $R_3 = r_3 P$, $\varphi = r_1 D_{ID_i}$, $\alpha = h_3(R, \varphi, ID_i, T_1)$, $w = r_3 P_{pub}$, $AID_i = ID_i \oplus h_2(R_3, w)$.	
	$\xrightarrow{\textbf{Auth1}=(AID_i, R, R_3, \alpha, T_1)}$	Compute $w' = sR_3$, $ID_i' = AID_i \oplus h_2(R_3, w')$, $\varphi' = \frac{1}{s+h_1(ID_i')} R$, $\alpha' = h_3(R, \varphi', ID_i', T_1)$. If $\alpha' = \alpha$, pick $r_4 \in \mathbb{Z}_q^*$, and set $R_4 = r_4 P$, $\hat{R} = r_4 R_3$, $\beta = h_4(R, w', \varphi', T_2)$, $SK_S = h_5(ID_i', ID_S, w', \varphi', \hat{R})$.
	Compute $\beta' = h_4(R, w, \varphi, T_2)$. If $\beta' = \beta$, compute $\hat{R}' = r_3 R_4$, $SK_U = h_5(ID_i, ID_S, w, \varphi, \hat{R}')$.	$\xleftarrow{\textbf{Auth2}=(R_4, \beta, T_2)}$
With the session key SK_U, the auxiliary device can securely exchange data with the server.	$\xleftarrow{SK_U}$	

Figure 3: The plain mutual authentication

pass a server S's authentication. However, Han *et al.* [10] have proved the claim was false.

4 Conclusion

We show the Wu *et al.*'s key agreement scheme is flawed. We want to stress that Paillier encryption seems inappropriate for most mobile devices because its running modulus is very big. The modular reduction with such a big modulus is generally considered to outsource to the cloud.

Acknowledgements

We thank the National Natural Science Foundation of China (Project 61411146001), and are grateful to the reviewers for their valuable suggestions.

References

[1] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference*, pp. 41–55, 2004.

[2] N. D. W. Cahyani, B. Martini, K. K. R. Choo, AKBP M. N. Al-Azhar, "Forensic data acquisition from cloud-of-things devices: windows smartphones as a case study," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 14, 2017.

[3] Z. J. Cao and O. Markowitch, "Comment on "circuit ciphertext-policy attribute-based hybrid encryption with verifiable delegation in cloud computing"," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 392–393, 2021.

[4] W. Y. Chao, C. Y. Tsai, and M. S. Hwang, "An improved key-management scheme for hierarchical access control," *International Journal of Network Security*, vol. 19, no. 4, pp. 639–643, 2017.

[5] J. S. Chen, C. Y. Yang, and M. S. Hwang, "The capacity analysis in the secure cooperative communication system," *International Journal of Network Security*, vol. 19, no. 6, pp. 863–869, 2017.

[6] Y. H. Chen, L. C. Huang, I. C. Lin, and M. S. Hwang, "Research on the secure financial surveillance blockchain systems," *International Journal of Network Security*, vol. 22, no. 4, pp. 708–716, 2020.

[7] S. F. Chiou, H. T. Pan, E. F. Cahyadi, M. S. Hwang, "Cryptanalysis of the mutual authentication and key agreement protocol with smart cards for wireless communications," *International Journal of Network Security*, vol. 21, no. 1, pp. 100–104, 2019.

[8] P. Gope and T. Hwang, "An efficient mutual authentication and key agreement scheme preserving strong anonymity of the mobile user in global mobility networks," *Journal of Network and Computer Applications*, vol. 62, pp. 1–8, 2016.

[9] P. Gope and T. Hwang, "Lightweight and energy-efficient mutual authentication and key agreement scheme with user anonymity for secure communication in global mobility networks," *IEEE Systems Journal*, vol. 10, no. 4, pp. 1370–1379, 2016.

[10] Y. X. Han, C. X. Xu, D. B. He, and K. F. Chen, "On the security of a key agreement and key protection scheme," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3293–3294, 2020.

[11] D. He, S. Zeadally, N. Kumar, J. H. Lee, "Anonymous authentication for wireless body area networks with provable security," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2590–2601, 2017.

[12] L. C. Huang, C. H. Chang, and M. S. Hwang, "Research on malware detection and classification based on artificial intelligence," *International Journal of Network Security*.

[13] L. C. Huang, T. Y. Chang, and M. S. Hwang, "A conference key scheme based on the diffie-hellman key exchange," *International Journal of Network Security*, vol. 20, no. 6, pp. 1221–1226, 2018.

[14] M. Karuppiah and R. Saravanan, "A secure authentication scheme with user anonymity for roaming service in global mobility networks," *Wireless Personal Communications*, vol. 84, no. 3, pp. 2055–2078, 2015.

[15] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology - EUROCRYPT'99, International Conference on the Theory and Application of Cryptographic Techniques*, pp. 223–238, 1999.

[16] H. T. Pan, E. F. Cahyadi, S. F. Chiou, M. S. Hwang, "Research on batch verification schemes for identifying illegal signatures," *International Journal of Network Security*, vol. 21, no. 6, pp. 1062–1070, 2019.

[17] H. T. Pan, H. W. Yang, and M. S. Hwang, "An enhanced secure smart card-based password authentication scheme," *International Journal of Network Security*, vol. 22, no. 2, pp. 358–363, 2020.

[18] D. Quick and K. Choo, "Pervasive social networking forensics: Intelligence and evidence from mobile device extracts," *Journal of Network and Computer Applications*, vol. 86, pp. 24–33, 2017.

[19] C. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[20] W. L. Tai, Y. F. Chang, and W. H. Huang, "Security analyses of a data collaboration scheme with hierarchical attribute-based encryption in cloud computing," *International Journal of Network Security*, vol. 22, no. 2, pp. 212–217, 2020.

[21] J. L. Tsai and N. W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, vol. 9, no. 3, pp. 805–815, 2015.

[22] Y. L. Wang, J. J. Shen, and M. S. Hwang, "A survey of reversible data hiding for vq-compressed images," *International Journal of Network Security*, vol. 20, no. 1, pp. 1–8, 2018.

[23] L. Wu, J. Wang, K. K. R. Choo, D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 319–330, 2019.

Biography

Lihua Liu, associate professor with Department of Mathematics at Shanghai Maritime University, received her PhD degree in applied mathematics from Shanghai Jiao Tong University. Her research interests include combinatorics and cryptography.

Leming Hong is currently pursuing his M.S. degree from Department of Mathematics, Shanghai Maritime university. His research interests include numerical computation theory and information theory.

Zhengjun Cao, associate professor with Department of Mathematics, Shanghai University, received his PhD degree in applied mathematics from Academy of Mathematics and Systems Science, Chinese Academy of Sciences. He had served as a post-doctor in Computer Sciences Department, Université Libre de Bruxelles. His research interests include cryptography, discrete logarithms and quantum computation.

Research on Robustness of Deep Neural Networks Based Data Preprocessing Techniques

Hong Zhao, You-kang Chang, and Wei-jie Wang
(Corresponding author: You-kang Chang)

School of Computer and Communications, Lanzhou University of Technology, 36 Peng-jia-ping Road, Lanzhou, Gansu 730050, China
(Email: 2507576651@qq.com)

(Received Feb. 25, 2021; Revised and Accepted Dec. 25, 2021; First Online Feb. 26, 2022)

Abstract

Deep neural networks are an important method to achieve artificial intelligence. However, recent research has shown that deep neural networks are vulnerable to adversarial attacks, where small perturbations in the input data can cause the network to make incorrect judgments. In this paper, we proposed a defense method based on the combination of image restoration and image super-resolution reconstruction, which removed the interference from the image through image restoration and recovered some high-frequency information of the image through super-resolution reconstruction of the image to correct the feature extraction area of the deep neural network. Experimental results showed that the method could significantly improve the robustness of deep neural networks and effectively defend against the FGSM series of attacks.

Keywords: Adversarial Attack Defense; Deep Neural Networks; Image Restoration; Model Robustness; Super-Resolution Reconstruction

1 Introduction

Deep neural networks are an important implementation of artificial intelligence [28], have made significant progress in the fields of image recognition, network analysis, medical image analysis, autonomous driving, and natural language processing, but the problem of their low robustness has also come to the fore. In 2013, Szegedy [23] pointed out that deep neural networks are vulnerable to the attack of adversarial samples, that is the addition of some subtle perturbations, which are not easily observed by human vision, to the original image can make the classification network's classification results wrong. In 2014, Goodfellow [8] proposed the linearity hypothesis to explain why adversarial samples cause errors in the output of deep neural networks, and proposed the Fast Gradient Sign Method (FGSM) adversarial sample generation method. Then, Kurakin [11] pointed out that adversarial samples also exist in the real world, and the authors fed natural images obtained from cameras and other sensors into the Inception_v3 classification network and showed that the network suffered from adversarial samples.

In this paper, we proposed a defense method based on image restoration and image super-resolution reconstruction, which can well recover the original class of the adversarial sample, and our main contributions are:

1) Our proposed defense method outperforms other methods overall;
2) Super-resolution reconstruction of the adversarial samples is performed to remap them back into clean image space, enabling the classification network to correctly classify them;
3) The proposed method does not add artifacts when dealing with perturbations in the adversarial sample, preserves the key information of the image, minimizes the distance between the recovered image and the clean image, and reduces the impact of the adversarial sample on the classifier performance;
4) The proposed method can effectively defend against FGSM series of attacks, especially in the face of the more aggressive MI-FGSM and MDI^2FGSM.

Figure 1 shows the heat map distribution of the classification network when extracting features. The darker color in the figure indicates that the region has more influence on the classification results.

In Figure 1, the left is the clean image with its corresponding heat map; the middle is the adversarial sample with its heat map, and the right is the image with its heat map after image restoration and super-resolution reconstruction. In the heat map, it is observed that the key feature region is reduced when the classification network extracts the features of the adversarial sample, and other regions of the image are added as feature regions, which leads the network model to make wrong judgments; on the right is the recovered image of the adversarial sample after image restoration and super-resolution reconstruction, and it can be seen from the figure that although

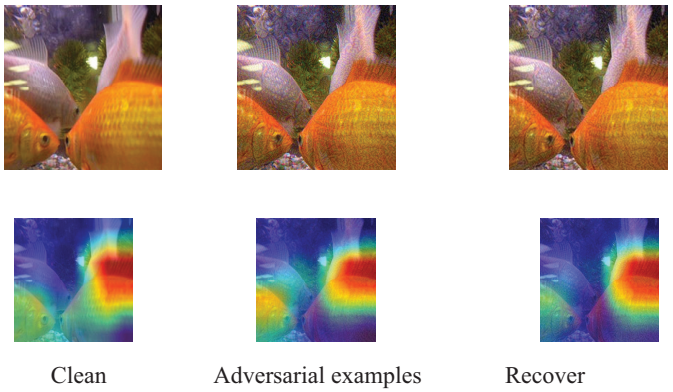


Figure 1: Feature regions of the image extracted by the network model during classification

there are still differences between the key feature region and the clean sample, the key feature region range is approximately the same. The final results showed that the adversarial sample was classified as 'brain_coral' with a confidence rate of 0.08, while the recovered image was classified as 'goldfish' with a confidence rate of 0.11, indicating that our proposed method could remap the adversarial sample back to the clean sample space, so that the classification network could make correct classification results and ensure the robustness of the network.

Section 2 gives an overview of related work, Section 3 proposes a defense method using image restoration and super-resolution reconstruction, Section 4 designs experiments and discusses the results, and Section 5 concludes the whole paper.

2 Related Work

Adversarial samples have been widely presented in text sentiment analysis [22], wireless audio technology [21], speech recognition [2], and other fields. In the following, common adversarial attacks and defense methods in the field of image classification are described.

2.1 Image Classification Adversarial Samples

Let f be a deep neural classification network with an input of a clean image x and an output of the correct class y of x, i.e., f(x)=y. The adversarial sample in image classification is the image xadv generated after adding a perturbation m to x, i.e., x_{adv}=x+m such that f(x_{adv})=y_{adv}≠y.

In the attack, with f(x_{adv})≠y as the purpose, it is a non-target attack; f(x_{adv})=y_{tar}≠y as the purpose, it is a target attack. The adversarial image-clean image relationship satisfies l_p the parametric number, i.e.$\|X_{adv} - X\|_p < \varepsilon$, $\| \bullet \|_p$ as shown in Equation (1), with p∈ $\{0, 1, ..., \infty\}$.

$$\|X_{adv} - X\|_p = \sqrt[p]{\sum_{i=0}^{m} |X_{adv_i} - X_i|^p} \tag{1}$$

where p is usually 0, 2 or ∞. When p=0, it indicates the number of pixels in the clean sample that are changed due to the adversarial attack, p=2 indicates the Euclidean distance between the clean sample and the adversarial sample, p=∞ indicates the maximum value of the modification of the phase element value at the corresponding position between the clean sample and the adversarial sample,ε is the perturbation strength that controls the difference between the adversarial sample and the clean image.

2.2 Image Classification Against Attack Algorithm

Since Szegedy first proposed adversarial samples, researchers have proposed different adversarial attack methods and their optimization methods. Currently, for white-box attacks, they can be mainly classified into gradient-based, optimization-based, and generative adversarial network (GAN) based attack methods.

2.2.1 Gradient-based Attack Methods

In 2014, Goodfellow [8] proposed the Fast Gradient Sign Method (FGSM) attack method, which first maximizes the loss function using the input image, then passes the loss value to the input image, calculates the gradient value and gradient direction, and finally adds the calculation result as a perturbation to the input image to generate an adversarial sample. FGSM is a single-step attack algorithm, and its advantage is that can generate adversarial samples quickly, and disadvantage is that some of the generated adversarial samples have weak attack capability.

In 2016, Kurakin [11] proposed the Basic Iterative Method (BIM) method. The BIM algorithm is similar to FGSM in that both compute perturbations based on gradients. The difference is that the BIM method iteratively computes perturbations along the direction of the gradient and recalculates the gradient direction after each iteration, adding the perturbations to the input image gradually to generate the adversarial samples attack capability is stronger.

In 2018, Dong [6] proposed the Momentum Iterative Fast Gradient Sign Method (MIFGSM) method based on the BIM method. This method incorporates the momentum iterative gradient method in the process of adversarial sample generation, i.e., the velocity vector is accumulated along the gradient direction of the loss function during the iterative process. This method accelerates the convergence and descent of the gradient and stabilizes the gradient descent direction.

Since single-step attacks produce underfitting and iterative attacks produce overfitting, to solve these two

problems, in 2019, Xie proposed the Diverse-Input-Iterative FGSM (DI^2FGSM) and Momentum-Diverse-Input-Iterative FGSM (MDI^2FGSM) [26].

2.2.2 Optimization-based Attack Methods

Gradient-based adversarial attack methods require that the gradients of the network model and the objective function are solvable and that the generated adversarial samples are not very similar to the input samples. In 2017, Carlini and Wagner proposed the Carlini and Wagner (C&W) [3] method: this method uses the Adam-Optimizer optimizer to generate adversarial samples. In the process of optimization, the authors map the adversarial samples to (-∞,+∞), and the distance between the generated adversarial samples and clean images is smaller; the adversarial samples generated by controlling the parameter k of the network model, the larger the value of k, the higher the confidence rate of classification error, but at the same time the optimization process will be more difficult and the adversarial samples will be difficult to find.

For the black-box attack cannot know the internal structure and parameter information of the classification network model and cannot know the gradient of the model. Chen proposed the Zeroth Order Optimization (ZOO) [4] method inspired by the C&W [3] method. By approximating the gradient information of the classification network model, the problem that the black-box attack cannot obtain the first- and second-order gradient information of the target model is solved. However, this method is proportional to the size of the input samples, and a large number of computations are required to estimate the gradient for input samples of large size. To solve this problem, the authors reduce the computational effort by dimensionality reduction and use the Adam method to speed up the convergence of the gradient.

2.2.3 GAN-based Attack Methods

Xiao used GAN networks to generate adversarial samples (AdvGAN) [7]. The AdvGAN framework consists of three parts: a generator G, a discriminator D, and a target attack network f. Firstly, an input sample x is fed into the generator G to generate a small perturbation G(x), and then an adversarial sample x_{adv} = x + G(x) is fed into the discriminator D to distinguish between a clean sample x and an adversarial sample xadv, while using the adversarial sample xadv to attack the target network f. The AdvGAN method can generate realistic and natural adversarial samples to achieve efficient attacks, and at the same time, it can directly generate adversarial samples without resorting to the transferability of adversarial attacks when facing black-box attacks.

2.3 Image Classification Against Attacks for Defense

Adversarial attacks have raised concerns about the problem of robustness of deep neural networks, and different methods have been proposed by researchers for the defense against adversarial attacks, which are mainly classified into three categories.

1) Modifying the input samples of the network. Such as adversarial training [24] denoising of the adversarial samples [25] and data compression [5]. The advantages of these methods are fast computation and no need to modify the network structure. The disadvantages of these methods are that denoising and data compression cause the loss of high-frequency information of the samples and the network extracts the wrong feature regions when extracting features, which makes the classifier make wrong judgments.

2) Adding more layers or adding sub-networks to the network and changing the loss or activation functions. Such as the defense distillation method [19], and the biologically inspired defense method [18]. This approach of modifying the network model increases the complexity of the network by improving the randomness and cognitive performance of the model, and requires retraining the network with high overhead, which is still ineffective in defending against specific attacks that are carefully designed.

3) Using other networks as additional networks to defend against adversarial attacks. Methods such as Generative Adversarial Network (GAN) network based defense methods, MagNet [15]. However, the training process of using GAN network as a defense mechanism has a large overhead and its defense capability is not significantly improved if it is not trained properly; MagNet has good defense capability against black-box and gray-box attacks when used as a defense method, but its performance still low in the case of white-box attacks.

3 Image Restoration and Super-resolution Reconstruction

Our proposed defense methods based on image restoration and image super-resolution reconstruction proposed in this paper belong to the first category of defense methods. Since the denoising methods in the first category remove high-frequency information from images, which affects the robustness of the network; and data compression methods have been shown to be ineffective in defending against C&W attacks [3] and universal interference attacks [1], which are due to the fact that larger data compression leads to a decrease in the classification accuracy of the original samples, while smaller compression cannot effectively eliminate interference. Compared

with them, image restoration removes the small perturbations added in the adversarial sample, reduces the influence of noise in the image, super-resolution reconstruction restores some of the high-frequency information of the image, compensates for the information lost during image restoration, and remaps the features of the adversarial sample back to the feature space of the clean image, which can finally output the correct classification results. The proposed method does not require extensive network training; and compared with the denoising method, it can effectively recover the high-frequency information of the image, improve the robustness of the model, and show better defense performance in the face of attack methods with stronger attack performance.

3.1 Image Restoration with Adaptive Features

Since the adversarial samples are produced by adding elaborate small perturbations to clean images, image denoising is performed using an adaptive feature modification network, a continuous-threshold image restoration method that produces images with different levels of smoothness by adjusting different levels of restoration parameters. The effect of this continuous thresholding method on the classification results will be described in detail in the later experiments.

3.1.1 Related Theories

The CNN model is first trained using the initial image training set with noise level $\sigma = 15$ to obtain the convolution kernel f_{15}; then the same model is trained using the training set with noise level $\sigma = 50$ with the convolution kernel f_{50}. It is observed that the difference between the two convolution kernels is small [9]. It is thus thought that f_{50} can be obtained from f_{15} after transformation, as shown in Equation (2).

$$min_g \| f_{50} \odot x - g \odot (f_{15} \odot x) \|^2 \tag{2}$$

where x denotes the feature map, $\odot$ denotes the convolution operation, and g denotes the convolution kernel of the f_{15} to f_{50} approximation transform.

By changing the mean and variance of the convolution kernel g, the output of the network can be continuously variable. The function to obtain the intermediate convolution kernel is shown in Equation (3).

$$f_{mid} = f_{15} + \lambda(g - I) \odot f_{15}, 0 \leq \lambda \leq 1 \tag{3}$$

where I is the unit convolution kernel and λ is the interpolation factor. By controlling the range of λ, the range of f_{mid} will vary between f_{15} and g$\odot f_{15}$.

3.1.2 Adaptive Feature Modification Network

A CNN network is first trained using the training set of the initial noise level for parameter correction; then an Adaptive Feature Modification (AdaFM) layer is added to the CNN network to keep the network parameters unchanged and continue training on the final noise level to obtain the parameters of AdaFM for continuous image recovery, and the training process of the AdaFM layer is shown in Equation (4).

$$AdaFM(x_i) = g_i \odot x_i + b_i, 0 \leq i \leq N \tag{4}$$

where g_i and b_i denote the convolution kernel and bias, respectively, x_i is the input feature map, N is the number of feature maps, and g_i depends on the noise level of the input samples. the AdaFM layer is tested as shown in Equation (5).

$$g_i^* = I + \lambda(g_i - I), b_i^* = \lambda b_i \tag{5}$$

where I is the unit convolution kernel and λ is the interpolation coefficient.

By this method, the value of λ is adjusted to obtain recovered images with different noise levels.

3.2 Super-resolution Reconstruction

In deep learning-based image super-resolution methods, the model is usually trained using a large number of training samples to recover the local information of the image. However, this approach ignores the internal similarity features existing in the image itself, so researchers proposed the non-local attention mechanism [12]. By capturing the structural information between different regions of an image through the non-local attention mechanism, the feature similarity of the image itself is fully explored. Later, through further research, researchers proposed the cross-scale non-local attention (CS-NL attention) [14] mechanism based on the non-local attention mechanism, which combines the in-scale non-local prior and the cross-scale non-local prior to find cross-scale feature correlation. It has been able to demonstrate that cross-scale image block similarity is widely present in natural images.

In-scale non local attention: For a given image feature map X, the in-scale non-local attention mechanism searches the entire picture for regional structure information with similar characteristics,and mines the long-distance dependence between them. The non-attention mechanism is calculated as shown in Equation (6).

$$Z_{i,j} = \sum_{g,h} \frac{exp\phi(X_{i,j}, X_{g,h})}{\sum_{u,v} exp\phi(X_{i,j}, X_{u,v})} \psi(X_{g,h}) \tag{6}$$

where (i,ϕ), (g,h) and (u,v) are the coordinate pairs of X, $\psi()$ is the feature transformation function, and $\phi()$ is the similarity measure function, as shown in Equation (7).

$$\phi(X_{i,j}, X_{g,h}) = \theta(X_{i,j})^T \delta(X_{g,h}) \tag{7}$$

where θ, δ denote the feature transformation.

The in-scale non-local attention mechanism searches feature regions from the same feature map, capturing similar features within the same scale. Unlike the in-scale non-local attention mechanism, the cross-scale non-local attention mechanism focuses on the correlation between low-resolution pixels and high-resolution image regions, which can search high-frequency information directly from low-resolution images and recover high-frequency information in combination with high-resolution images, making the reconstruction results more natural and accurate.

Cross-scale non-local attention: Since the existing methods cannot measure the similarity of pixels to regions, the feature map X is first down sampled s scale, i.e., Y=X↓S, as shown in Equation (8).

$$Z_{si,sj}^{S\times S} = \sum_{g,h} \frac{exp\phi(X_{i,j}, Y_{g,h})}{\sum_{u,v} exp\phi(X_{i,j}, Y_{u,v})} \psi(X_{sg,sh}^{S\times S}) \quad (8)$$

where Y is the down sampled image, (i, j) denotes the coordinate pair of X, (g, h) denotes the coordinate pair of Y, and (u, v) denotes an arbitrary coordinate pair. $X_{sg,sh}^{s\times s}$ denotes the image block of size $s \times s$ corresponding to the position of Y. $Z_{si,sj}^{s\times s}$ denotes the feature image block of size $s \times s$ at position (si, sj).

Super-resolution reconstruction is able to recover some of the high-frequency information of an image after image restoration. Considering the super-resolution reconstruction method as a mapping function, the robustness of the classification network is improved by reprojecting the adversarial samples back to the space of clean samples through mapping.

3.3 Algorithm

Section 3.1 and 3.2 describe the process of image restoration and super-resolution reconstruction, respectively. Algorithm 1 describes the flow of the proposed defense method based on the above two algorithms. Firstly, the effects of noise are suppressed using an adaptive feature modification network; then the super-resolution reconstruction method is considered as a mapping function to remap the adversarial samples back to the natural image space; and finally, the recovered images are fed into the classification network model.

Algorithm 1 Adversarial sample defense based on image restoration and super-resolution reconstruction

/* Image Restoration */
Input: Adversarial sample X_{adv}=X+m
Output: Restored images $X_{denoising}$ =D(X_{adv})
1) Input the adversarial samples into the network and perform the convolution operation.
2) Followed by residual processing, adding an adaptive feature modification layer after the convolution layer in each residual block for image restoration.
3) Output the restored image $X_{denoising}$.
/* Image reconstruction */
Input: Restoration image $X_{denoising}$
Output: Super-resolution images X_{output}=S(·)
4) The restored image is converted from RGB space to YCbCr space, where Y denotes the luminance channel, Cb and Cr denote the blue and red chrominance channels, respectively, and super-resolution reconstruction is performed in the Y channel, which is because the human visual system is more sensitive to changes in luminance compared to changes in chromatic aberration.
5) The adversarial samples are reconstructed using a super-resolution reconstruction method to map the adversarial samples back to the clean image space.
6) Converts YCbCr space to RGB space and outputs the reconstructed image.
7) The images are fed into the classification network model for category determination.

4 Experimental Design and Analysis of Results

100 images were randomly selected on the ILSVRC dataset for defense model evaluation and compared using different classification networks, namely VGG19, Xception, Inception-v3, Inception ResNet-v2, Google Net, ResNet50, ResNet152. For the classification network model, the pre-trained model of ImageNet was used directly.

Attacks: Adversarial samples are generated using different attack methods, including FGSM, BIM, Deep Fool [16], C&W, MI-FGSM, DI^2FGSM and MDI^2FGSM. For FGSM, the adversarial samples are generated using ε = 2. All attacks are performed in a white-box environment.

Defense: The method proposed in this paper is compared with literature methods, including image compression [10], pixel deflection [20] and super-resolution reconstruction [17], respectively. All experiments are performed against the same attack and on the same set of images.

For super-resolution reconstruction, the deep unfolding super-resolution network (USRNet) [27] and Enhanced Deep Residual Network (EDSR) [13] were used for comparison, respectively. The experimental results show that our proposed method exhibits better results due to the addition of the cross-scale non-local attention module to the network.

4.1 Experimental Results and Analysis

Table 1 compares the proposed method with other three commonly used methods. In the proposed method, the image restoration parameter λ = 0.1, and the data in the table are top-1 accuracy rates, and the experiments is aimed at non-targeted attacks. No Defense in the table

Table 1: Performance comparison of various methods on the ILSVRC validation set

Classifier	clean Images	FGSM	BIM	DeepFool	C&W	MI-FGSM	DI^2FGSM	MDI^2FGSM
No Defense								
VGG19	60	43	59	58	60	19	45	26
Xception	80	53	73	73	76	14	51	21
Inception-v3	81	27	66	71	57	4	3	3
Inception ResNet v2	79	55	78	77	76	6	53	25
Image Compression								
VGG19	55	31	52	56	56	14	39	20
Xception	70	45	66	70	65	24	57	32
Inception-v3	71	**40**	63	65	60	**14**	31	**7**
Inception ResNet v2	70	45	70	71	70	19	57	40
Pixel Deflection								
VGG19	**63**	43	55	**67**	53	20	49	29
Xception	**78**	53	**74**	72	**74**	15	62	22
Inception-v3	**77**	31	67	70	58	4	3	3
Inception ResNet v2	**78**	57	72	69	**74**	7	62	28
Wavelet Denoising + Image Super Resolution								
VGG19	58	28	55	60	56	18	48	**34**
Xception	77	50	**74**	**74**	67	14	50	21
Inception-v3	**77**	33	**70**	**73**	61	4	3	3
Inception ResNet v2	**78**	53	**74**	**75**	73	6	558	25
Our work:AdaFM+ CS-NL attention								
VGG19	55	**45**	**58**	56	**57**	**31**	**52**	32
Xception	75	**58**	67	67	66	**26**	**63**	**34**
Inception-v3	77	30	67	66	**66**	**14**	**47**	4
Inception ResNet v2	74	**60**	67	69	70	**21**	**66**	**41**

indicates the accuracy of the adversarial sample after no defense method.

The overall performance of our proposed defense method is better as seen in Table 1. For the three attack algorithms BIM, Deep Fool, and C&W, our proposed method is slightly less accurate than Pixel Deflection and Wavelet Denoising + Image Super Resolution, but for the FGSM series of attacks with stronger attack performance, the three compared methods perform poorly, while our proposed method has better robustness.

In the FGSM attack algorithm, our method achieves 45% accuracy on the VGG19 network model, which is 14%, 2% and 17% better than the above three methods, respectively. For the DI^2FGSM algorithm with stronger attack performance, on the Xception classification model, our proposed algorithm can reach 34% accuracy, compared to 32%, 22% and 21% for the other three methods, respectively.

4.2 Effects of Different Super-resolution Reconstruction Methods

For the proposed defense method, USRNet and EDSR super-resolution reconstruction methods are chosen to compare the performance. The amplification factor s=2, the restoration parameter λ=0.0, the accuracy is top-1 metric, and the classification network model is Google Net.

As can be seen from Table 2, our proposed method is able to exhibit effective defense performance. These results also demonstrate that the super-resolution reconstruction method is indeed effective in adversarial sample defense. Comparison of network structures: USRNet is a super-resolution reconstruction method that combines both model-based and learning-based approaches, making this method simultaneously model-based flexible and learning-based efficient. However, it only reconstructs information in low-resolution images compared with CS-NL attention, and does not introduce information in cross-scale images in the reconstruction process; EDSR also uses residual networks and its adds multi-scale operations, but this method does not correlate the correlation between low-resolution images and high-resolution images,

Table 2: Performance comparison of different super-resolution methods

Classifier	No Defense	USRNet	EDSR	OURS
Clean Images	51	56	52	**57**
FGSM	35	43	38	**45**
BIM	50	54	48	**55**
DeepFool	46	55	47	**56**
C&W	54	55	53	**60**
MI-FGSM	44	31	27	**34**
DI^2FGSM	45	53	50	**57**
MDI^2FGSM	36	38	38	**41**

Table 3: Effect of different down sampling methods on the accuracy

	Inception Resnet-v2			Inception-v3			Xception		
	BI	BD	DN	BI	BD	DN	BI	BD	DN
clean	**73**	69	27	**62**	**62**	13	**65**	57	21
Bim	67	**68**	27	**61**	57	19	**67**	65	23
C&W	**68**	67	30	60	**61**	22	**64**	56	22
DI^2FGSM	**69**	65	28	52	**59**	19	**65**	64	22
MDI^2FGSM	52	**53**	28	6	11	**14**	40	**42**	23

and cannot deeply mine the information in low-resolution images.

4.3 Effect of Different Down Sampling

When constructing **L**ow-**R**esolution (LR) and **H**igh-**R**esolution (HR) datasets for image super-resolution reconstruction using the **B**icubic **I**nterpolation (BI) method, we compared the Gaussian **B**lur **D**own Dampling (BD) and Bicubic Interpolation **D**ownscale + Gaussian **N**oise (DN) methods. Experiments were conducted at a scale of magnification s=3.

As can be seen from Table 3, among the three classification models, the overall performance of the results generated using the BI down sampling method is better. Compared to BI, BD and DN use Gaussian blur and Gaussian noise, respectively, in the down sampling process, which is equivalent to adding a new perturbation to the image after restoration, so the accuracy will not be improved.

4.4 The Respective Effects of Image Restoration and Super-resolution Reconstruction

Table 4 shows the results of defending against the adversarial attack using two modules, image restoration technique and super-resolution reconstruction, respectively. Where the denoising parameter $\lambda = 0.1$ and the scale factor s = 2.

The data in the table show that image restoration can remove added perturbations, but it also loses the high-frequency information in the image, and the robustness of the network model is not effectively improved, but also reduced; using super-resolution reconstruction as a defense method, the performance is improved than image restoration. The performance is further improved by combining image restoration with super-resolution reconstruction because image restoration based on adaptive feature modification layer can remove small perturbations and super-resolution reconstruction based on cross-scale non-local attention can effectively recover high-frequency information in images. The proposed defense method is robust to both iterative and non-iterative attacks.

4.5 Effect of Different Magnifications

S denotes the scale factor of the reconstructed image, and this section will compare the effects of different reconstruction factors on the robustness of the model, and the scale factors S chosen are 2, 3 and 4, respectively.

As can be seen from Table 5, both in the ResNet152 model and in the Google Net model, the C&W, FGSM, and MIFGSM attack algorithms show good performance when S = 2, and for clean samples, their accuracy is also improved. The reason for this is that when the scale factor is large, it causes smoothing in the reconstructed images after reconstruction, which affects the classification performance.

4.6 Defense Results of Different Threshold λ Against FGSM Variants

In the experiments, it is found that for different FGSM variants attack algorithms, different λ values have a large impact on the robustness of the defense model. Here, we choose λ values of 0.0, 0.1, 0.3, 0.5, and 0.7, respectively.

As can be seen from Table 6, the robustness of the proposed method is better overall when λ is 0.1, 0.3 and 0.5, respectively. In particularly, for the MI-FGSM and MDI^2FGSM attack algorithms, when $\lambda = 0.5$, the accuracy is improved by 30% and 29%, respectively, on the Inception ResNet v2 model compared to Wavelet Denoising + Image Super Resolution. The reason is that when λ gradually increases, the image is also smoother, which can effectively reduce the effect of perturbation and reconstruct a high-quality image, but when λ exceeds a certain threshold, the restored image is too smooth and the key information has been lost along with the added perturbation, which affects the quality of the reconstructed image and causes a decrease in robustness. Figure 2 shows the different reconstructed images corresponding to different threshold λ values as an example of the MIFGSM attack algorithm.

5 Conclusion

Deep neural networks have penetrated into people's life, but with them come many security issues. Adversarial attack is a new security problem proposed by researchers in recent years, which makes artificial intelligence make wrong judgment by interfering with the input data. Therefore, it is especially important to design defense methods against attacks. In this paper, we proposed a defense method based on image restoration and super-resolution reconstruction, where image restoration is used to reduce the effect of noise and image super-resolution reconstruction is used to recover key information of the image and remap the adversarial samples back to the clean sample space. The proposed method does not require retraining or learning the network and is effective for both iterative and non-iterative attacks. Experimental results

Table 4: Effect of image restoration and super-resolution reconstruction on different network models

	VGG16				ResNet50				Xception			
Attack	No Defense	Denoising	SR	Denoising+SR	No Defense	Denoising	SR	Denoising+SR	No Defense	Denoising	SR	Denoising+SR
Clean	63	54	62	**66**	58	54	56	**63**	70	66	66	**75**
FGSM	40	38	40	**48**	37	42	**47**	**45**	53	42	50	**58**
BIM	59	45	56	**65**	57	52	51	**61**	63	57	61	**67**
DeepFool	53	47	50	**58**	43	47	46	**57**	63	59	56	**67**
C&W	47	55	**59**	55	57	51	57	**62**	56	56	53	**66**
MI-FGSM	17	20	27	**31**	17	27	21	**30**	14	17	18	**26**
DI^2FGSM	48	44	47	**52**	49	50	51	**58**	51	55	58	**63**
MDI^2FGSM	29	26	32	**35**	33	21	26	**35**	21	26	28	**34**

Table 5: Effect of different magnifications

	ResNet152				GoogleNet			
Attack	No Defense	S=2	S=3	S=4	No Defense	S=2	S=3	S=4
clean	70	**77**	71	68	51	**57**	55	55
C&W	70	**76**	69	64	54	**60**	51	48
FGSM	50	**56**	52	42	45	**45**	42	38
MIFGSM	25	**42**	26	31	44	**44**	34	39

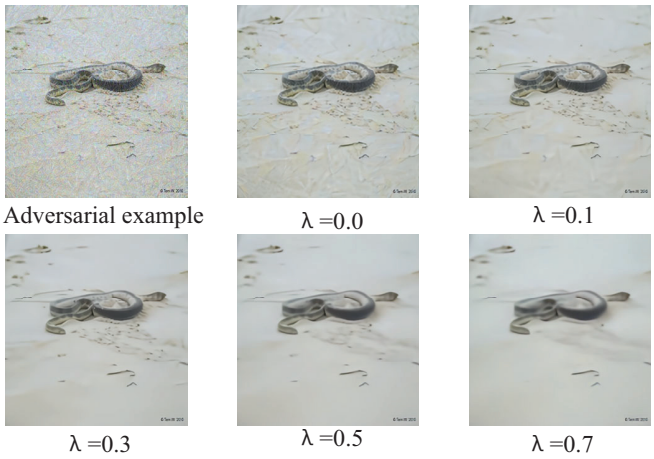


Figure 2: Different restoration parameters λ and their corresponding reconstructed images

Table 6: Experimental results of different λ values on FGSM variants

Classifier	FGSM	MI-FGSM	DI^2FGSM	MDI^2FGSM
Our work:λ=0.0				
VGG19	42	25	52	29
Xception	49	19	67	26
Inception-v3	32	9	36	4
Inception ResNet v2	59	12	66	31
Our work:λ=0.1				
VGG19	45	31	52	32
Xception	58	26	63	34
Inception-v3	30	14	47	4
Inception ResNet v2	60	21	66	41
Our work:λ=0.3				
VGG19	48	34	46	40
Xception	57	32	61	54
Inception-v3	33	22	53	15
Inception ResNet v2	63	33	63	51
Our work:λ=0.5				
VGG19	43	35	43	38
Xception	53	39	55	48
Inception-v3	38	27	47	26
Inception ResNet v2	59	36	57	54
Our work:λ=0.7				
VGG19	42	35	40	39
Xception	37	29	39	27
Inception-v3	37	29	39	27
Inception ResNet v2	56	39	54	52

show that our proposed method is more robust than the recently proposed defense methods compared to them.

Acknowledgments

This research is supported by the National Natural Science Foundations of China under Grants No. 51668043 and No. 61262016. The authors gratefully acknowledge the anonymous reviewers for their helpful comments and suggestions.

References

[1] N. Akhtar, J. Liu, and A. Mian, "Defense against universal adversarial perturbations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3389–3398.

[2] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," *arXiv preprint arXiv:1801.00554*, 2018.

[3] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee sympo-*

sium on security and privacy (sp). IEEE, 2017, pp. 39–57.

[4] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[5] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, "Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression," *arXiv preprint arXiv:1705.02900*, 2017.

[6] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9185–9193.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[9] J. He, C. Dong, and Y. Qiao, "Modulating image restoration with continual levels via adaptive feature modification layers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 056–11 064.

[10] X. Jia, X. Wei, X. Cao, and H. Foroosh, "Comdefend: An efficient image compression model to defend adversarial examples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6084–6092.

[11] A. Kurakin, I. Goodfellow, S. Bengio *et al.*, "Adversarial examples in the physical world," 2016.

[12] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin, "Non-locally enhanced encoder-decoder network for single image de-raining," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1056–1064.

[13] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

[14] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T. S. Huang, and H. Shi, "Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5690–5699.

[15] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 135–147.

[16] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[17] A. Mustafa, S. H. Khan, M. Hayat, J. Shen, and L. Shao, "Image super-resolution as a defense against adversarial attacks," *IEEE Transactions on Image Processing*, vol. 29, pp. 1711–1724, 2019.

[18] A. Nayebi and S. Ganguli, "Biologically inspired protection of deep networks from adversarial attacks," *arXiv preprint arXiv:1703.09202*, 2017.

[19] N. Papernot and P. McDaniel, "On the effectiveness of defensive distillation," *arXiv preprint arXiv:1607.05113*, 2016.

[20] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8571–8580.

[21] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *International conference on machine learning*. PMLR, 2019, pp. 5231–5240.

[22] S. Ren, Y. Deng, K. He, and W. Che, "Generating natural language adversarial examples through probability weighted word saliency," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 1085–1097.

[23] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[24] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.

[25] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 501–509.

[26] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2730–2739.

[27] K. Zhang, L. V. Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3217–3226.

[28] X. Zhao, "Attack-defense game model: Research on dynamic defense mechanism of network security," *International Journal of Network Security*, vol. 22, no. 6, pp. 1037–1042, 2020.

Biography

Hong Zhao was born in 1971. He is a professor and a supervisor of doctor student at Lanzhou University of Technology. His major research field is System modeling and simulation, deep learning, natural language processing. E-mail: zhaoh@lut.edu.cn.

You-kang Chang was born in 1994. He is a doctor student at Lanzhou University of Technology. His major research field is adversarial attacks and defense adversarial attacks. E-mail: 2507576651@qq.com.

Wei-jie Wang was born in 1994. She is a doctor student at Lanzhou University of Technology.Her major research field is speaker recognition. E-mail: 1132744259@qq.com

Network Traffic Feature Weight Map Based Approach for Intrusion Detection

Jianwu Zhang[1,2], Yu Zhang[1], Xingbing Fu[2,3], Yanjun An[1], Yuhang Yang[1], and Fagen Li[4]
(Corresponding author: Xingbing Fu)
School of Communication Engineering, Hangzhou Dianzi University[1]
Hangzhou 310018, P.R. China
Science and Technology on Communication Networks Laboratory, P.R. China[2]
Email: uestcfuxb@126.com
School of Cyberspace, Hangzhou Dianzi University[3]
Hangzhou 310018, P.R. China
School of Computer Science and Engineering, University of Electronic Science and Technology of China[4]
(Received Feb. 26, 2021; Revised and Accepted Dec. 5, 2021; First Online Feb. 26, 2022)

Abstract

The development of the Internet has increased the importance of network security. Hence, it is urgent to detect network intrusions effectively. Deep learning has provided a new direction for network intrusion detection in recent years. In this work, Subspace Weighting Co-Clustering (SWCC) algorithm and its improved algorithm, Feature Weight Matrix (FWM) algorithm, are applied to Convolutional Neural Networks (CNN) employed as intrusion detection engines. Two network traffic-based intrusion detection models, SWCC-CNN and FWM-CNN, are proposed. To measure the validity of the models, we used the NSL-KDD intrusion detection dataset for evaluation. Experimental results show that the two models proposed in this work have better classification performance than the traditional CNN model; furthermore, FWM-CNN is less complex and more accessible to implement than SWCC-CNN.

Keywords: Convolutional Neural Networks; Deep Learning; Feature Weight Map; Intrusion Detection

1 Introduction

With the development of computer technology, the Internet is changing people's daily life [14] , but it is followed by a variety of security threats [15]. Therefore, network security issues have attracted widespread attention. How to use intrusion detection systems (IDS) to protect various networks has become a hot topic. As a system monitoring and analyzing network events, IDS is mainly used to detect intrusion events that are occurring or have occurred. Intrusion detection is a classification problem that can be divided into binary classification and multi-classification. the binary classification is divided into normal samples and abnormal samples, and the multi-classification is divided into normal samples and various attack samples. At present, IDS is divided into host-based IDS and network-based IDS. Host-based IDS depends on the reliability of the host, cannot detect network attacks, and the type of attacks detected is more limited. With the development of network technology, network attack methods are becoming more diverse, and network-based IDS research is urgently in need. The main purpose of this work is to improve the performance of IDS.

Machine learning methods have been widely used to identify various types of attacks. At present, most conventional supervised learning algorithms have been applied to IDS. Rajesh Thomas *et al.* [17] used the NSL-KDD dataset to comprehensively review the related research of machine learning-based IDS, and proposed a general process based on abnormal IDS, and emphasized the importance of sample processing. Zhang *et al.* [25] proposed a network intrusion detection algorithm based on improved k-means and multi-layer support vector machine (SVM). The algorithm first improved k-means, and the improved k-means can be used to divide the dataset into different clusters, and label them as normal or abnormal. Finally, they use the multi-level SVM to classify the anomaly clusters in detail. Apichit Pattawaro *et al.* [12] proposed an exception network IDS based on feature selection, k-means clustering and XGBoost classification model. They filter the redundancy features, and use the k-means clustering method to perform hyperparameter tuning on each cluster of the corresponding classification model. Prashant Kushwaha *et al.* [8] proposed a feature selection algorithm based on mutual information. The algorithm selects the top k features as the input feature in the descending order of relevance, and compares a series of classifiers. It is concluded that SVM performs best in binary classification.

As the network traffic grows, the intrusion data become more complex. Compared with the conventional machine learning algorithm, the deep learning algorithm can optimize the internal parameters of the model, and deeper computing to explore more complex data structures in large datasets. Because of the problem of neural networks in intrusion detection, such as information redundancy, a large amount of data, long training time, and ease to fall into local optimum, Zhao *et al.* [26] proposed a deep belief network (DBN) and probabilistic neural network (PNN) intrusion detection method; this method uses DBN to convert raw data into low-dimensional data, and shortens the training and test time of PNN. The particle swarm optimization (PSO) algorithm is used to optimize the number of DBN hidden layer nodes to improve the performance of the network feature expression of DBN. Chowdhury *et al.* [6] implemented a small sample deep learning method for intrusion detection, trained a deep CNN as a general feature extractor, and then trained an SVM or 1-NN classifier to classify filtered datasets.

Wu *et al.* [21] used CNN to automatically select traffic characteristics from the original dataset, and set the weight coefficients of the cost function of each class according to the number of each class to solve the problem of dataset imbalance; the model can reduce the false positive rate and improve the accuracy of rare attack recognition. Xiao *et al.* [22] proposed a large-scale network intrusion detection method based on CNN; they use a dimensionality reduction method to generate a dataset and convert the dataset into a two-dimensional matrix for identification. Li *et al.* [9] proposed a deep learning intrusion detection method based on multi-convolutional neural network fusion according to the correlation, the feature data is divided into four parts, and then the one-dimensional feature data is transformed into a gray image. The method of applying long-short-time memory network (LSTM) to intrusion detection system is studied in detail in [2,3,23]. Yin *et al.* [24] proposed an intrusion detection model based on RNN, and studied the performance of the model for two-class and multi-classification. Vinayakumar *et al.* [19] proposed stacked recurrent neural network(S-RNN) which has potential to learn complex temporal behaviors quickly including sparse representations.

Muhuri *et al.* [10] combined LSTM with RNN, proposed the IDS based on LSTM-RNN, and proved the validity of the IDS. Diro *et al.* [1] employed a deep learning approach for distributed intrusion detetection in the Internet of Things. The performance of their deep learning model are better than those of the shallow counterpart, and their distributed detection system is superior to a centralized detection system. Unfortunately, they employ the public dataset, not the IoT dataset, which does not reflect the practical network traffic in the Internet of Things. Koroniotis *et al.* [11] employed machine learning and deep learning algorithms to detect intrusions in the Internet of Things. They employ a bot-IoT dataset that incorporates both normal IoT-related, other network traffic and various types of attack traffic used by botnets. The dataset is validated by four metrics in terms of Accuracy, Precision, Recall, and Fall-out. However,these works do not calculate the weights of different categories of features, so the importance of different features is not clear. When the features of different categories are uniformly filtered or all the features are simply input, the loss and redundancy of the features will be caused. Therefore, this work proposes a feature weight map intrusion detection system based on network traffic. We calculate the weights for different categories of features to obtain the weight reference map, which is then applied to the CNN for recognition.

The remainder of the paper is organized as follows: in Section 2, we introduce two benchmark mapping algorithms, as well as SWCC-CNN and FWM-CNN models. In Section 3, we performed experimental analysis and comparison to prove the validity of the proposed models. In Section 4, we conclude and specify the future work.

2 Proposed Method and Design

2.1 Dataset Processing

The intrusion detection dataset used in this work is NSL-KDD [18], which is a subset of the KDD99 dataset that does not contain redundant records [16]. We classify the dataset by labels and divide them into 39 subcategories. It can be seen that the NSL-KDD dataset is unevenly distributed, and it is divided into five categories, of which the normal class and the DoS class account for a large proportion. Sireesha Rodda *et al.* [13] evaluated the class imbalance performance of the NSL-KDD dataset using four commonly used classification models and found that highly unbalanced classes were not correctly classified. To balance the data of the training set, we can randomly partition the data into a training set or a test set according to a certain proportion, and randomly sample the categories in which the normal class and the DoS class occupy a large proportion in the training set to reduce the sample proportion; for a small number of categories, we make sure that all samples are present, repeated sampling is carried out until the sample number meets the small sample base set by the experiment.In this experiment, the number ratio of U2R, U2L and Probe was set as 0.5:0.8:1.Through the above method, we can get a training set with class balance.

The data sampling pseudo code is in Algorithm 1.

In the pseudo-code, 'M' is the cardinality of samples of normal class and DoS class; 'N' is the cardinality of other small categories.

We store the feature data and labels of the class equilibrium training set in order, and perform the one-hot encoding process on the data labels. Then we quantify the character features in the feature data of the samples, and finally normalize the feature data samples. For the feature data and labels of the test set, we performed the same processing.

Algorithm 1: The data sampling

```
if label == 'normal' or label == 'DoS' then
    if label == 'normal' then
        arr = random_array(0 : len(data))[0 : 1.5 * M];
        data[arr]addsavetofile.csv;
    else
        arr = random_array(0 : len(data))[0 : M];
        data[arr]addsavetofile.csv;
    end
else
    data[:]addsavetofile.csv;
    L = data.length;
    if label == 'u2r' or label == 'u2l' then
        if label == 'u2l' then
            mul = 0.8;
        else
            mul = 0.5;
        end
    else
        mul = 1;
    end
    while i < mul * N − L do
        i++;
        num = random(len(data));
        data[num]addsavetofile.csv;
    end
end
```

2.2 Feature Map Acquisition

2.2.1 SWCC Weight Matrix

Referring to Chen *et al.* [5], it is found that when we perform feature clustering to filter features to reduce redundancy, we can generate a feature weight matrix, the degree of importance of all features in each category. Inspired by the theory of Chen *et al.* [5], because the weights of different classification samples are different, for different classifications, we can obtain a reference map to represent the feature weights of all classifications to obtain better recognition performance. This work proposes a Convolutional Neural Networks (CNN) intrusion detection system based on feature weight matrix C. In the new model, the first step is to cluster the NSL-KDD training set to generate the feature weight matrix. In the second step, we multiply each feature value x_i in a single sample data by the weights $c_{:i}$ of the corresponding sequence number in C, and finally obtain the feature map of the same dimension as C. Finally, we introduce the feature map set and label it into the CNN for training.

We introduce the above balanced training set A into the weight matrix algorithm to obtain the weight matrix C. Chen *et al.* [5] defined a subspace weight matrix on the row cluster $C = [c_{g,j}]_{K\times M}$, where $c_{g,j}$ is the weight of the jth column in the gth row cluster. Chen *et al.* [5] expressed the problem as an iterative clustering process by minimizing Equation (1).

$$P(U,V,Z,C) = \frac{1}{MN}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M} u_{i,g}v_{j,h}c_{g,j}d(x_{i,j},z_{g,h}) + \frac{\eta}{M}\sum_{g=1}^{K}\sum_{j=1}^{M} c_{g,j}\log\ c_{g,j}, \tag{1}$$

Subject to

$$\begin{cases} \sum_{g=1}^{K} u_{i,g} = 1,\ u_{i,g} \in \{0,1\}, & 1 \le \mathrm{i} \le \mathrm{N} \\ \sum_{h=1}^{L} v_{j,h} = 1,\ v_{j,h} \in \{0,1\}, & 1 \le \mathrm{j} \le \mathrm{M} \\ \sum_{j=1}^{M} c_{g,j} = 1,\ 0 < c_{g,j} < 1, & 1 \le \mathrm{g} \le \mathrm{K}, \end{cases} \tag{2}$$

where $U = [u_{i,g}]_{N\times K}$ is a 0 or 1 matrix of row clustering, where $u_{i,g} = 1$ represents the ith sample classification in the gth row cluster. $V = [v_{j,h}]_{M\times L}$ is a 0 or 1 matrix of column clustering, where $v_{j,h} = 1$ represents the jth sample classification in the hth column cluster. $Z = [z_{g,h}]_{K\times L}$ is the center distance for $K\times L$. $C = [c_{g,j}]_{K\times M}$ is a weight matrix, where $c_{g,j}$ represents the weight of the jth feature on the gth cluster. η is a regularization parameter. The superparameter η can adjust the matrix C, and the larger η can make the matrix C more average.

The distance $d(x_{i,j}, z_{g,h})$ is defined as

$$d(x_{i,j}, z_{g,h}) = (x_{i,j} - z_{g,h})^2, \tag{3}$$

where $x_{i,j}$ is the jth feature value of the ith sample in the dataset.

By this method, Chen *et al.* [5] simultaneously minimize intra-cluster dispersion and maximize weight entropy to stimulate more columns to facilitate identification of co-clusters. At the same time, this method effectively avoids the problem of identifying the co-cluster by few columns in the sparse data.

In the scheme proposed by Chen *et al.* [5], the weight matrix is used for feature selection by weight ranking method. In their work, if the weight matrix algorithm is improved, the obtained weight matrix can also be used as the reference map of the dataset to deepen the feature contour to achieve better recognition performance.

The clustering method proposed by Chen *et al.* [5] is carried out without specific labels in the rows and columns. Because the NSL-KDD dataset already has a category label, that is, the row matrix U in its co-cluster is a known matrix. Therefore, in this work, when we process data, we first perform the one-hot encoding process on its label to form a row matrix $U = [u_{i,g}]_{N\times K}$.

This work assumes that some features may play a role in different categories when the feature columns are copolymerized. That is, one feature will be assigned to a different column cluster. In the method proposed by Chen *et al.* [5], each feature can only be assigned to one column co-cluster when it is assigned, where $\sum_{h=1}^{L} v_{j,h} = 1$.

Therefore, the constraints considered in this work are shown in Equation (4).

$$\begin{cases} \sum_{g=1}^{K} u_{i,g} = 1,\ u_{i,g} \in \{0,1\} \quad 1 \le \mathrm{i} \le \mathrm{N} \\ v_{j,h} \in \{0,1\}\ , 1 \le \mathrm{j} \le \mathrm{M} \\ \sum_{j=1}^{M} c_{g,j} = 1,\ 0\ < c_{g,j}\ < 1 \quad 1 \le \mathrm{g} \le \mathrm{K}. \end{cases} \tag{4}$$

According to the constraints in the above Equation (4), $U = [u_{i,g}]_{N\times K}$ and $X = [x_{i,j}]_{N\times M}$ are known, and the Equation (1) proposed by Chen *et al.* [4, 5] needs to be partially improved as Equation (5).

$$P(V,Z,C) = \frac{1}{MN}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M} u_{i,g}\frac{v_{j,h}}{\sum_{h=1}^{L} v_{j,h}} c_{g,j} d(x_{i,j}, z_{g,h}) + \frac{\eta}{M}\sum_{g=1}^{K}\sum_{j=1}^{M} c_{g,j}\log\ c_{g,j}. \tag{5}$$

We can minimize Equation (5) to obtain the weight matrix $C = [c_{g,j}]_{K\times M}$. To find the minimum of Equation (5), we can solve it by iteratively solving the following three minimization steps.

1) Given the variables $Z = [z_{g,h}]_{K\times L}$, $C = [c_{g,j}]_{K\times M}$, find the minimum value of $P(V, \hat{Z}, \hat{C})$.

2) Given the variables $V = [v_{j,h}]_{M\times L}$, $C = [c_{g,j}]_{K\times M}$, find the minimum value of $P(\hat{V}, Z, \hat{C})$.

3) Given the variables $V = [v_{j,h}]_{M\times L}$, $Z = [z_{g,h}]_{K\times L}$, find the minimum value of $P(\hat{V}, \hat{Z}, C)$.

The minimum value of $P(V, \hat{Z}, \hat{C})$ can be obtained by Equation (6).

$$\begin{cases} v_{j,t} = 1 \quad , if\ \min P_{([h])} \quad for\ 1 \le t \le L \\ P_{([h])} = \frac{1}{Len([h])}\sum_{g=1}^{K}\sum_{i=1}^{N} \hat{u}_{i,g}\hat{c}_{g,j} d(x_{i,j}, \hat{z}_{g,t}) \\ where\ t\ in\ [h],\ Len([h])\ is\ the\ length\ of\ [h] \\ v_{j,h} = 0,\ otherwise. \end{cases} \tag{6}$$

In order to find $V = [v_{j,h}]_{M\times L}$, this work introduces the concept of average minimum. In each row of $V = [v_{j,h}]_{M\times L}$, the average minimum set [h] is calculated separately. If t is in the set [h], then $v_{j,h}$=1; if t is not in the set [h], then $v_{j,h}$=0. The minimum value of $P(\hat{V}, Z, \hat{C})$ can be obtained by Equation (7).

$$z_{g,h} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} u_{i,g}\hat{v}_{j,h}\hat{c}_{g,j}x_{i,j}}{\sum_{i=1}^{N}\sum_{j=1}^{M} u_{i,g}\hat{v}_{j,h}\hat{c}_{g,j}}. \tag{7}$$

The minimum value of $P(\hat{V}, \hat{Z}, C)$ can be obtained by Equation (8).

$$c^{*}_{g,j} = \frac{\exp\{\frac{-E_{g,j}}{\eta}\}}{\sum_{j'=1}^{M}\exp\{\frac{-E_{g,j'}}{\eta}\}}, \tag{8}$$

where

$$E_{g,j} = \frac{1}{N}\sum_{h=1}^{L}\sum_{j=1}^{N} u_{i,g}\hat{v}_{j,h} d(x_{i,j}, \hat{z}_{g,h}). \tag{9}$$

By iterating the above three steps, until the Equation (5) reaches the local minimum, the weight matrix $C = [c_{g,j}]_{K\times M}$ is achieved.

The SWCC weight matrix process is in Algorithm 2.

Algorithm 2: SWCC

Require:
the intrusion dataset X, the labelset U, the number of feature clusters L and η;

Ensure:
the matrix of weight-value C.

Init
set $v_{j,h}$= 0, $c_{g,j}$= $\frac{1}{m}$, i=0 and $z_{g,h}$= random()
start co-clustering to follow these steps.

Repeat
Calculate V^{i+1} by (6).
Calculate Z^{i+1} by (7).
Calculate C^{i+1} by (8) and (9).
i =i+1.

Until (5) obtains its local minimum.

According to the SWCC, the minimum of the Equation (5) can be solved by iteratively solving the Equation (6), (7), (8) and (9), each of which is a convex optimization problem with regard to the variables. Therefore, the algorithm can converge to a local minimal solution.

2.2.2 FWM Weight Matrix

The computational complexity of the improved algorithm of the SWCC weight matrix is O(rNMKL), where r is the number of iterations of SWCC, M is the number of the features, N is the number of the data, K is the number of the categories, and L is the number of the clusters of features of SWCC. Due to the large number of samples in the experiment, the time it takes us to iterate each sample to obtain the weight matrix is long. In order to save time cost, this work proposes another weight matrix algorithm, FWM. It is well known that each useful feature value in a sample of the same classification should be relatively close, that is, the variance of each useful feature value in the sample of the same classification is small. Therefore, we compare with the different variances and means of a particular feature in each class. If the variance is small and the mean is large, we consider that the feature contributes significantly to the classification, and give it a

larger weight. If the variance and mean are small or the variance is large, we end up giving a smaller weight.

Based on the above considerations, we can list Equation (10).

$$P(C) = \frac{1}{M}\sum_{g=1}^{K}\sum_{j=1}^{M}\frac{E_{D_{g,j}}E_{E_{g,j}}}{D_{g,j}E_{g,j}}c_{g,j} + \frac{\eta}{M}\sum_{g=1}^{K}\sum_{j=1}^{M}c_{g,j}\log\ c_{g,j}, \tag{10}$$

subject to

$$\begin{cases} \sum_{g=1}^{K} u_{i,g} = 1,\ u_{i,g} \in \{0,1\}\ ,\ 1 \le \mathrm{i} \le \mathrm{N} \\ \sum_{j=1}^{M} c_{g,j} = 1,\ \ 0\ < c_{g,j}\ < 1\ \ ,\ 1 \le \mathrm{g} \le \mathrm{K}, \end{cases} \tag{11}$$

where

$$E_{g,j} = \frac{1}{\sum_{i=1}^{N} u_{i,g}}\sum_{i=1}^{N} u_{i,g}x_{i,j}, \tag{12}$$

$$D_{g,j} = \frac{1}{\sum_{i=1}^{N} u_{i,g}}\sum_{i=1}^{N}(u_{i,g}x_{i,j} - E_{g,j})^2, \tag{13}$$

$$E_{D_{g,j}} = \frac{1}{K}\sum_{g=1}^{K} D_{g,j}, \tag{14}$$

$$E_{E_{g,j}} = \frac{1}{K}\sum_{g=1}^{K} E_{g,j}. \tag{15}$$

$E_{g,j}$ is the mean of the jth feature in the g classification; $D_{g,j}$ is the variance of the jth feature in the g classification; $E_{E_{g,j}}$ is the mean of the mean $E_{g,j}$ of the jth feature in each category; $E_{D_{g,j}}$ is the mean of the variance $D_{g,j}$ of the jth feature in each category.

According to Equation (10), we can transform the problem of weight matrix into the minimum of P(C), and then have the weight matrix, where $c_{g,j}$ can be obtained from Equation (16).

$$c_{g,j}=\frac{e^{-\frac{E_{D_{g,j}}E_{E_{g,j}}}{\eta D_{g,j}E_{g,j}}}}{\sum_{j=1}^{M} e^{-\frac{E_{D_{g,j}}E_{E_{g,j}}}{\eta D_{g,j}E_{g,j}}}}. \tag{16}$$

Proof. According to the Lagrange multiplier method, if P(C) is to be minimized, this is equivalent to minimizing Equation (17).

$$\mathrm{L}(c,\gamma,\tau) =\frac{1}{M}\sum_{g=1}^{K}[\frac{1}{N}\sum_{j=1}^{M}\frac{E_{D_{g,j}}E_{E_{g,j}}}{D_{g,j}E_{g,j}}c_{g,j} +\eta\sum_{j=1}^{M}c_{g,j}\log c_{g,j} + \gamma_g(\sum_{j=1}^{M}c_{g,j} - 1) + \sum_{j=1}^{M}\tau_{g,j}(c_{g,j} - {\theta_{g,j}}^2)]. \tag{17}$$

In Equation (17), ${\theta_{g,j}}^2$ is a relaxation factor which ensures that $c_{g,j}$ is greater than 0. We give a partial derivative to the parameters, and we have Equation (18),

$$\begin{cases} \frac{\partial \mathrm{L}(c,\gamma,\tau,\theta)}{\partial c_{g,j}} = \frac{1}{M}(\frac{1}{N}\frac{E_{D_{g,j}}E_{E_{g,j}}}{D_{g,j}E_{g,j}} + \eta + \eta\log c_{g,j} + \gamma_g + \tau_{g,j}) = 0 \\ \frac{\partial \mathrm{L}(c,\gamma,\tau)}{\partial \gamma_g} = \frac{1}{M}(\sum_{j=1}^{M} c_{g,j} - 1) = 0 \\ \frac{\partial \mathrm{L}(c,\gamma,\tau)}{\partial \tau_{g,j}} = \frac{1}{M}(c_{g,j} - {\theta_{g,j}}^2) = 0 \\ \frac{\partial \mathrm{L}(c,\gamma,\tau)}{\partial \theta_{g,j}} = -\frac{2}{M}\theta_{g,j}\tau_{g,j} = 0, \end{cases} \tag{18}$$

where $\tau_{g,j}$ is a constant equal to 0, and

$$y_{g,j} = \frac{1}{N}\frac{E_{D_{g,j}}E_{E_{g,j}}}{D_{g,j}E_{g,j}}. \tag{19}$$

So from Equation (18) and (19), we have

$${c_{g,j}}^*=\mathrm{e}^{-\frac{y_{g,j}}{\eta}}\mathrm{e}^{-\frac{\eta+\gamma_g}{\eta}}. \tag{20}$$

Since $\sum_{j=1}^{M} c_{g,j}$=1 in Equation (18), we accumulate $c_{g,j}$ to have

$$\mathrm{e}^{-\frac{\eta+\gamma_g}{\eta}}=\frac{1}{\sum_{j=1}^{M}\mathrm{e}^{-\frac{y_{g,j}}{\eta}}}. \tag{21}$$

Finally, from Equation (20) and (21), we have

$${c_{g,j}}^*=\frac{\mathrm{e}^{-\frac{y_{g,j}}{\eta}}}{\sum_{j=1}^{M}\mathrm{e}^{-\frac{y_{g,j}}{\eta}}}. \tag{22}$$

□

We perform $\frac{\partial^2 \mathrm{L}(c,\gamma,\tau,\theta)}{\partial {c_{g,j}}^2}$ to obtain $\frac{\partial^2 \mathrm{L}(c,\gamma,\tau,\theta)}{\partial {c_{g,j}}^2}=\frac{\eta}{c_{g,j}} > 0$, so ${c_{g,j}}^*$ is the local minimal solution of $P(C)$. The FWM weight matrix algorithm is in Algorithm 3.

Algorithm 3: FWM

Require:
the intrusion dataset X, the labelset U and η;

Ensure:
the weight matrix C.

Init
let $c_{g,j}= \frac{1}{m}$

Calculate
$E_{g,j}$, $D_{g,j}$, $E_{D_{g,j}}$, $E_{E_{g,j}}$ by (12), (13), (14), (15).
${c_{g,j}}^*$ by (19), (22).

The computational complexity of this algorithm is O(MNK), where M is the number of the features, N is the number of the data, K is the number of the categories, L is the number of the clusters of features of SWCC, and

r is the number of iterations of SWCC. We can see that the algorithm complexity of FWM has significantly decreased, compared with the SWCC computational complexity O(rNMKL).

2.2.3 Model Designs of SWCC-CNN and FWM-CNN

Since the NSL-KDD dataset is divided into five categories and 41-dimensional features, the matrix size of the weight matrix $C = [c_{g,j}]_{K\times M}$ is 5*41. We use matrix $C = [c_{g,j}]_{K\times M}$ as the reference map. The reference map is divided into 5 lines, each of which is a sequence of weights for features in a category. We multiply the weights in each row of the reference map by the sequence of data feature values (Figure 1). In this way, the original one-dimensional vector can be expanded into a feature map of size 5*41.

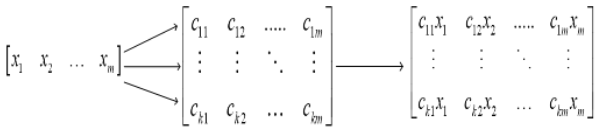


Figure 1: Schematic Diagram of Feature Map Acquisition.

After obtaining the data feature map, we can regard the feature map as a 2-dimensional image for identification with the original data label. Vinayakumar *et al.* [20] and Ding *et al.* [19] applied CNN in intrusion detection, they consider the feature selection and preprocessing of the original one-dimensional vector, and then feed the length 41 of data vector directly into the 1D CNN for training. As we say in the previous section, the set of important features included in each type of attacks will be different. If we simply extract the same feature set for each sample to identify different types of attacks, it may cause the loss of important features or the redundancy of features, which will have some impact on the recognition performance. Since the feature weights of each class in the reference map are respectively represented in different rows, and the feature map is used for 2-dimensional image recognition, the feature weight information of different categories can be preserved.

We use the reference map to multiply the data vector by the method of Figure 1 to obtain the feature map, which is similar to the image samples in image recognition. Based on the above statements, we use the obtained feature map to find the hidden feature relationship through CNN to identify. We set W to the minimal dimension value, and set the convolution kernel size to be $F \times F$, the convolution step size to S, the number of padding 0 to be P, and the maximal pooling layer size to be $D \times D$. The output minimal side length N can be calculated by the Equations (23) and (24).

$$M = [\frac{W - F + 2 \times P}{S}] + 1, \tag{23}$$

$$N = [\frac{M}{D}]. \tag{24}$$

We multiply the data vector by the reference map in the method of Figure 1 to generate a numerical matrix with a feature map size of 5*41, so the minimal dimension value is W=5. The sample data set can be divided into five categories, so the final output of the CNN is a length 5 of the one-dimensional vector. We use the CNN proposed by Ding *et al.* [19] and modify it. Then we introduce the feature maps obtained by the two reference maps in the previous section into the CNN for identification. Two IDS models, the SWCC-CNN model and the FWM-CNN model, can be obtained. In the following experimental research and comparison, the specific experimental comparison and recognition performance are given.

3 Experimental Evaluations

In this study, Intrusion Detection System (IDS) was evaluated using four main parameters. These are Accuracy, True Positive Rate (TPR), False Positive Rate (FPR) and Precision. TP is the number of positive samples that are correctly classified, FP is the number of positive samples that are misclassified, and FN is the number of negative samples that are misclassified.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{25}$$

$$TPR = \frac{TP}{TP + FN}. \tag{26}$$

$$FPR = \frac{FP}{FP + TN}. \tag{27}$$

$$Precision = \frac{TP}{TP + FP}. \tag{28}$$

In the SWCC-CNN method, we choose L=K=5 for feature clustering. When we regularize the parameter η selection in the experiment, a set $\eta = [0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10]$ is selected as the regular parameters. For each of the η, 20 initial co-cluster centers Z can be generated at random. So we end up with 100 co-cluster results for analysis.

We feed the 20 co-cluster results generated by the different parameters η into the network to find the final recognition rate, and average the 20 recognition rates generated by the different parameters η. The average of recognition rates is used to evaluate the performance of the weight matrix generated by the different parameters of η in the model. In this way, we can get the identification rate as Figure 2.

As illustrated in Figure 2, where the ordinate is the recognition rate, and the abscissa is the value of η, when η is 0.1, the average recognition rate of the SWCC-CNN method is the highest; therefore, 0.1 is used as the value of the parameter η.

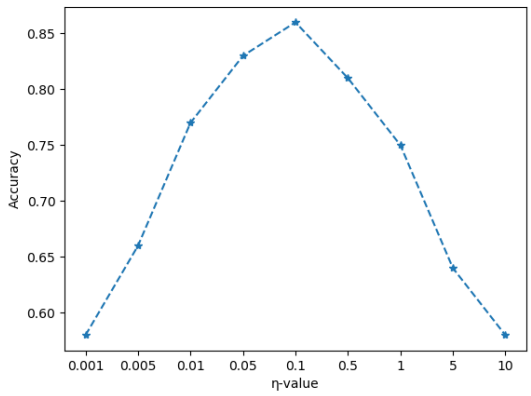


Figure 2: Recognition Rate of Different η Values in SWCC-CNN.

In the FWM-CNN method, the regularization parameter is selected as a given set η= [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1].

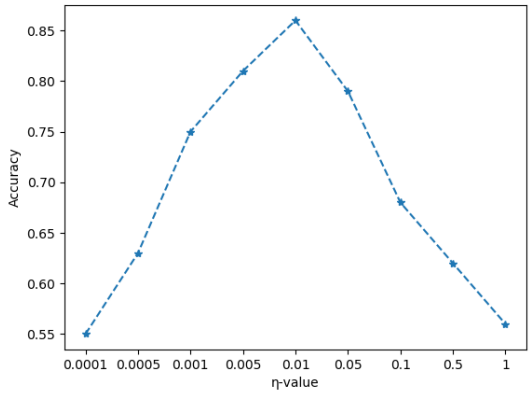


Figure 3: Recognition Rate of Different η Values in FWM-CNN.

As illustrated in Figure 3, where the ordinate is the recognition rate, and the abscissa is the value of η, when η is 0.01, the average recognition rate of the FWM-CNN method is the highest; therefore, 0.01 is used as the value of the parameter η.

When we handle the training set from NSL-KDD [18], we set M=31493 and N=20995. We get the training set and KDDTest+ as shown in Table 1.

Table 1: The Number of Each Class in Training Set and KDDTest+

Class	Normal	Dos	u2r	u2l	Probe	Total
Training Set	47239	31493	10497	16796	20995	127020
KDDTset+	9711	7458	200	2754	2421	22544

This work demonstrates the effectiveness of the SWCC-CNN model and the FWM-CNN model through three comparative experiments. In the first group, we use the method of Ding *et al.* [7] to compare and verify the validity of the weight matrices in the SWCC-CNN model and the FWM-CNN model, and directly import the one-dimensional data into the CNN of Ding *et al.* [7] for identification (hereafter referred to as CNN-1); in the second group, we use SWCC-CNN to set the parameter η to 0.1 for identification; in the third group, we use the FWM-CNN method to set the parameter η to 0.01 for identification.

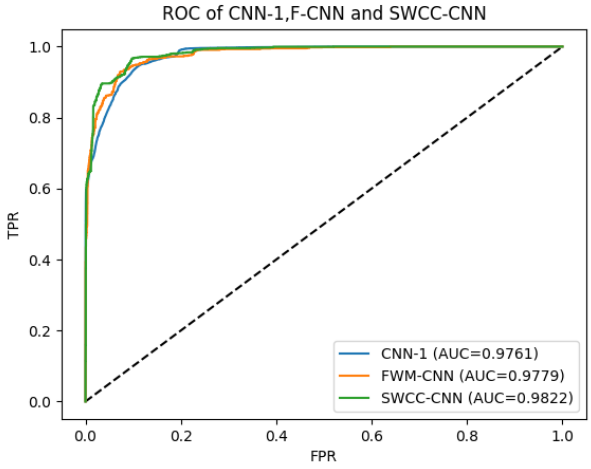


Figure 4: ROC of CNN-1, FWM-CNN, and SWCC-CNN

We perform the experiments to compare the ROC of three models, CNN-1, FWN-CNN, and SWCC-CNN, and the experimental results are shown in Figure 4. In each group, we use the sample set A as the training set, and calculate the true positive rate (TPR), false positive rate (FPR), and Precision of each category and the total recognition rate in the same test set. The experimental results are listed in Table 2. As shown in Table 2 that our proposed SWCC-CNN and FWM-CNN are better than CNN-1 [13] in terms of Accuracy, TPR ,FPR, and Precision.

As illustrated from Figure 4 that one of our models, SWCC-CNN has the biggest AUC value, which means SWCC-CNN has the best performance. Additionally, in the experimental results of Ding *et al.* [7], the TPR of u2r and u2l, which are the rare attacks, are very low, and the recognition rate of different attacks is not balanced. It can be seen from the Table 1 that after using the balanced dataset, the recognition rates of the five categories are more balanced in the CNN-1 model than Ding *et al.* [7], but the recognition rate of the other three categories has declined. In our two models, SWCC-CNN and FWM-CNN, it can be seen that the recognition rate of attack categories is significantly improved, while the recognition rate of normal class is not much decreased. Compared with SWCC-CNN, the normal class of FWM-CNN has

Table 2: Test Results of Each Algorithm in the NSL-KDD Dataset

Algorithms	indicators	Normal	Dos	u2r	u2l	Probe	Accuracy
CNN-1 [7]	TPR	93.21%	84.62%	28.5%	56.46%	52.87%	80.97%
	FPR	20.96%	4.65%	0.21%	6.41%	0.26%	
	Precision	77.09%	89.99%	55.34%	55.06%	96.10%	
SWCC-CNN	TPR	93.05%	92.07%	43.0%	60.31%	73.73%	86.21%
	FPR	18.67%	2.07%	0.10%	2.92%	1.34%	
	Precision	79.04%	95.68%	80.37%	74.18%	86.95%	
FWM-CNN	TPR	89.62%	89.11%	57.0%	63.87%	83.11%	85.32%
	FPR	13.5%	3.44%	0.88%	4.35%	1.34%	
	Precision	83.45%	92.76%	36.66%	95.34%	88.17%	

lower TPR and higher FPR of attack class, but the TPR of FWM-CNN is improved compared with the recognition of u2r, u2l, and Probe attack categories. When the true positive rate is increased, the false positive rate is acceptable.

4 Conclusions and Future Work

In this work, we propose two CNN network intrusion detection models based on feature maps, and compare the performance of these two models with that of the CNN-1 model. Experimental results show that our two models are better than the CNN-1 model; furthermore, among the two models, FWM-CNN is less complex and easier to implement than SWCC-CNN.

The first direction for future work is to continue to investigate how to improve the performance of the model in detecting real-time network traffic, and add more features to enrich its feature library. The second direction of future research is a reduction of computational cost in complex structures with the high number of parameters. The third direction of future research is that we will employ more deep learning algorithms to build intrusion detection models to detect network intrusions.

Acknowledgments

This document is the results of the research project funded by the National Science Foundation of China No. 61772162 and No. U1866209, Science and Technology on Communication Networks Laboratory under Grant 6142104180413. The authors thank the anonymous reviewers for helpful suggestions.

References

[1] N. C. Abebe Abeshu Diro, "Distributed attack detection scheme using deep learning approach for internet of things," in *Future Generation Computer Systems*, pp. 761–768, 2018.

[2] S. Althubiti, W. Nick, J. Mason, X. Yuan, and A. Esterline, "Applying long short-term memory recurrent neural network for intrusion detection," in *Southeast-Con 2018*, pp. 1–5, 2018.

[3] A. Boukhalfa, A. Abdellaoui, N. Hmina, and H. Chaoui, "Lstm deep learning method for network intrusion detection system." *International Journal of Electrical & Computer Engineering*, vol. 10, no. 3, 2020.

[4] R. Chen, N. Sun, X. Chen, M. Yang, and Q. Wu, "Supervised feature selection with a stratified feature weighting method," *IEEE Access*, vol. 6, pp. 15 087–15 098, 2018.

[5] X. Chen, J. Z. Huang, Q. Wu, and M. Yang, "Subspace weighting co-clustering of gene expression data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 2, pp. 352–364, 2019.

[6] M. M. U. Chowdhury, F. Hammond, G. Konowicz, C. Xin, H. Wu, and J. Li, "A few-shot deep learning approach for improved intrusion detection," in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON'17)*, pp. 456–462, 2017.

[7] Y. Ding and Y. Zhai, "Intrusion detection system for nsl-kdd dataset using convolutional neural networks," in *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, pp. 81–85, 2018.

[8] P. Kushwaha, H. Buckchash, and B. Raman, "Anomaly based intrusion detection using filter based feature selection on kdd-cup 99," in *TENCON*

2017 - 2017 IEEE Region 10 Conference, pp. 839–844, 2017.

[9] Y. Li, Y. Xu, Z. Liu, H. Hou, Y. Zheng, Y. Xin, Y. Zhao, and L. Cui, "Robust detection for network intrusion of industrial iot based on multi-cnn fusion," *Measurement*, vol. 154, p. 107450, 2020.

[10] P. S. Muhuri, P. Chatterjee, X. Yuan, K. Roy, and A. Esterline, "Using a long short-term memory recurrent neural network (lstm-rnn) to classify network attacks," *Information*, vol. 11, no. 5, p. 243, 2020.

[11] E. S. B. T. Nickolaos Koroniotis, Nour Moustafa, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," in *Future Generation Computer Systems*, pp. 779–796, 2019.

[12] A. Pattawaro and C. Polprasert, "Anomaly-based network intrusion detection system through feature selection and hybrid machine learning technique," in *2018 16th International Conference on ICT and Knowledge Engineering*, pp. 1–6, 2018.

[13] S. Rodda and U. S. R. Erothi, "Class imbalance problem in the network intrusion detection systems," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT'16)*, pp. 2685–2688, 2016.

[14] I. H. Sarker and A. Kayes, "ABC-ruleminer: User behavioral rule-based machine learning method for context-aware intelligent services," vol. 168, Elsevier, p. 102762, 2020.

[15] I. H. Sarker, A. S. M. Kayes, and S. Badsha, "Cybersecurity data science: an overview from machine learning perspective," in *Journal of Big Data*, pp. 1–29, 2020.

[16] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, 2009.

[17] R. Thomas and D. Pavithran, "A survey of intrusion detection models based on nsl-kdd data set," in *2018 Fifth HCT Information Technology Trends (ITT'18)*, pp. 286–291, 2018.

[18] UNB, *NSL-KDD Dataset*, Feb. 19, 2022. (https://www.unb.ca/cic/datasets/nsl.html)

[19] R. Vinayakumar, K. Soman, and P. Poornachandran, "Evaluation of recurrent neural network and its variants for intrusion detection system (ids)," *International Journal of Information System Modeling and Design*, vol. 8, no. 3, pp. 43–63, 2017.

[20] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *2017 International Conference on Advances in Computing, Communications and Informatics*, pp. 1222–1228, 2017.

[21] K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50 850–50 859, 2018.

[22] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42 210–42 219, 2019.

[23] Y. Yan, L. Qi, J. Wang, Y. Lin, and L. Chen, "A network intrusion detection method based on stacked autoencoder and lstm," in *IEEE International Conference on Communications*, pp. 1–6, 2020.

[24] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21 954–21 961, 2017.

[25] X. F. Zhang, X. H. Hao, "Research on intrusion detection based on improved combination of k-means and multi-level svm," in *2017 IEEE 17th International Conference on Communication Technology*, pp. 2042–2045, Oct 2017.

[26] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing*, vol. 1, 2017, pp. 639–642, 2017.

Biography

Jianwu Zhang received the B.S. and M.S. degrees in wireless communication engineering from the Nanjing Institute of Communication Engineering, in 1983 and 1988, respectively, and the Ph.D. degree in information and communication system from Zhejiang University, China, in 1999. He is currently a Professor of telecommunication engineering with Hangzhou Dianzi University, China. His research interests include next generation mobile telecommunication systems, network optimization, information security, and AI application.

Yu Zhang is currently pursuing the master degree in electronics and communication engineering, Hangzhou Dianzi University. His research interests are network traffic analysis and information security.

Xingbing Fu received the Ph.D. degree from University of Electronic Science and Technology of China(UESTC) in 2016. His research interests include machine learning, cloud computing and cryptography.

Yanjun An is currently pursuing the master degree in information and communication engineering, Hangzhou Dianzi University. His research interests are artificial intelligence and cybersecurity.

Yuhang Yang received the master degree in electronics and communication engineering, Hangzhou Dianzi University in 2020. His research interests are intrusion detection and information security.

Fagen Li is a professor. His research interests are information security, and cryptography

Integrity-Preserving and Efficient Policy Evaluation for XACML

Kai Zheng and Xiuxia Tian
(Corresponding author: Xiuxia Tian)
College of Computer Science and Technology, Shanghai University of Electric Power
No. 2588 Changyang Road, Shanghai 200090, China
Email: xxtian@shiep.edu.cn
(Received Mar. 2, 2021; Revised and Accepted Dec. 25, 2021; First Online Feb. 26, 2022)

Abstract

Extensible Access Control Markup Language (XACML) is one of the current standards for specifying access control policies. In a cloud or distributed environment, policies missing integrity assurance can mislead the decisions of the access control system. We propose an architecture that stores policymakers' policies and policy verification data based on Merkle B Tree to cope with this challenge. This ensures the policy integrity via the collision resistance of hash functions. Furthermore, the evaluation performance of PDP is one of the critical aspects of the access control decision system. In this paper, we design an evaluation engine based on Rule Trie, an architecture that can accelerate the process of decision-making through the common rule attribute prefix. By simulating the arrival of access requests, experimental results reveal that the evaluation performance is better than that of Sun PDP and SBA-XACML. The methods preserve the correctness of decisions and can improve the evaluation efficiency significantly.

Keywords: Evaluation Performance; Merkle B Tree; Policy Integrity; Rule Trie; XACML

1 Introduction

Due to the rapid development of cloud computing [23], ubiquitous computing and e-commerce, the application of access control continues to expand. Access control plays an essential role in resource sharing and is an essential mechanism in security services.

Among many access control models, the Attribute-Based Access Control mode (ABAC) is suitable for an open environment. It uses attributes to describe the access subject and object uniformly, to realize the fine-grained authorization. The policy decision point (PDP) is an important component in ABAC and is responsible for evaluating access requests. An eXtensible Access Control Markup Language (XACML) developed by the OASIS for the ABAC model uses XML to describe access requests, policies, and responses. XACML standard provides the decision result by checking whether the user's access request matches the policy. The access control policies are contained in PAP (Policy Administration Point) within the XACML architecture. In the data-flow model of XACML, PAP makes policies and policy sets available to the PDP, which works like a policy service provider. Like in the Database Outsourcing Model [27], data is outsourced to a third-party database service provider (DSP) by the data owner. DSP maintains data in the DBMS and answers queries from clients as an agent of the data owner, which incurs the data integrity problem.

In a cloud or distributed environment, PAP is not fully trusted by policymakers. Therefore, policymakers must protect policy integrity when outsourcing policies to PAP. Particularly, PDP should have the ability to verify the returned policies are not modified by PAP deliberately when PDP retrieves polices from PAP. There are three critical dimensions in policy query authentication: correctness, completeness, and freshness. Correctness denotes that the PDP is capable of validating that the returned policy exists in the owner's database and has not been altered in any way. Completeness means that no replies have been elided from the result. Finally, freshness implies that the result policies are based on the most current version of the policy database. Especially in dynamic scenarios, policy owners update the policies staying at the servers from time to time and PDP's stale policies may cause unexpected effects.

Distributed resource sharing, web service, and other scenarios require the formulation of many XACML policies for fine-grained access control of resources. However, with the increase of access requests and the expansion of policy rules, policy evaluation's performance becomes the bottleneck affecting the performance of the authorization system. After the OASIS specified the XACML standard, Sun introduced the prototype system for evaluating access control polices. In the PDP module, Sun PDP takes a brute-force approach to retrieve rules in the policy library that apply to access requests. In a scenario with

a small policy size, the method of violent item-by-item retrieval has little impact on the performance evaluation. However, when there are tens of thousands of policy rules in the system, the evaluation delay and system cost cannot be ignored.

Our work focuses on finding an approach to guarantee the policy integrity before evaluating requests and improving the evaluation performance of PDP. Our contributions are as follows.

- An authentication data structure based on Merkle Hash Tree is used to guarantee the integrity of the policy. Radix Path Identifiers are used to store and retrieve Merkle Hash Tree-based authentication data to and from policy service providers.
- Rule Trie based on policy is established for the efficient evaluation of PDP, which uses the common attribute prefix of the rule to reduce the matching time.
- A Policy Matching Optimization Engine with Numerization Machine and Rule Trie is designed to improve the evaluation performance of PDP, and experimental comparison is made with Sun PDP, XEngine and SBA-XACML.

The rest of the article is organized as follows: Section 2 describes some of the work on Merkle Hash Tree, XACML, and PDP. Section 3 provides an approach overview. Section 4 presents data structures and techniques for ensuring policy integrity. An efficient policy matching optimization engine is presented in Section 5. Section 6 focuses on experiments and offering analysis. Section 7 concludes.

2 Related Work

Merkle Hash Tree is widely used to ensure data integrity. In terms of improving PDP evaluation performance, related work can be divided into three categories: policy clustering and reordering, employment of distributed authorization model, and optimization of mass policy sets.

2.1 Merkle Hash Tree

As an effective tool to generate data signatures and ensure data integrity, Merkle tree is widely used in cloud storage [12], outsourcing database [21, 27], big data [26], edge computing [10], and smart grid [18,25]. Merkle Hash Tree is a binary tree, the leaf node stores the hash value of data, and each non-leaf node holds the hash of the concatenated hashes of its children. When one of these data changes, the root hash changes. The root hash is usually public. After the data requester gets the data and verification data, he calculates the new root hash, the calculation overhead being usually small. If the new root hash is the same as the public root hash, the data requested is correct.

In the database outsourcing model [27], the Merkel B Tree (MBT), an authenticated data structure, is established for the data of the data owner. The data and authentication data are outsourced to the Database Service Provider and the client verifies the integrity of the requested data through the verification object. Mao *et al.* [12] proposed the Position-Aware Merkle Tree, a structure aiming to protect the integrity of cloud data and support dynamic maintenance. A triple is used to define the nodes of the Merkle tree. The user verifies the consistency of the challenge-response block by calculating the root node directly, rather than retrieving the entire Merkle tree. Sharma *et al.* [24] used the electrical medical records of the day as the leaf node, and then constructed the Merkle tree upward. The root of the Merkle tree is stored in the block, and the block is immutable. If the electrical medical records in the block are tampered with, the recalculated root hash will be inconsistent with that in the block, which can be detected by periodic tests. Nguyen *et al.* [20] used blockchain and Merkel tree to safely store user data in the e-education system. The Compact Spare Merkle Tree (CSMT) is used to store user profiles based on their constantly updated and inflated features. Among them, the huge user data is decomposed, and then the traditional Merkel tree is used for storage, and the root of the Merkel tree is used as a component of CSMT. Li *et al.* [10] used the improved Merkle Tree VMHT to generate proof of the integrity of the backup data to accelerate the audit of massive edge backup data in the edge computing environment. The application vendor establishes VMHT on the original data, generates the audit request, and sends it to the edge server. The edge server responds to the integrity proof. The application vendor checks the integrity proof against its VMHT and realizes the data integrity audit.

2.2 Policy Clustering and Reordering

In the field of policy clustering and reordering research, S. Marouf *et al.* [13] proposed a PDP framework based on adaptive reordering and clustering. First, rules are classified according to subject attributes, and then the K-means algorithm is used to cluster rules for subject attributes. The rules are reordered based on their success hit ratio based on their evaluation history. Clustering can reduce the number of rules compared to access requests, but the K-means algorithm needs to specify the number of clusters in advance, and the number of clusters is unknown in advance. Deng *et al.* [1] conducted two-stage clustering for policies. First, rules are clustered according to subject attributes, and then in each rule clusters, the spectral clustering algorithm is used for further rule clustering. According to the similarity between the access request and each rule cluster, the request search for the adaptation rule in the rule cluster with the highest similarity. If no adaptation rule is found, the next rule cluster is used. Zhang *et al.* [29] used the ACO algorithm to classify the policies, claiming that the ACO classification

effect was better than K-means.

2.3 Employment of Distributed Authorization Model

Focusing on employing an efficient distributed authorization model, Donia *et al.* [6] defined seven decomposition patterns based on attribute sets to decompose policies into multiple policies with fewer rules. A multi-PDP architecture is adopted, with each PDP processing a small number of rules. Diaz-Lopez *et al.* [5] proposed an efficient distributed policy management solution. The approach redefines the XACML architecture, using master/slave PAPs for communication. Policy privilege is defined using meta-policies, and the SAML is adopted to protect policy. Deng *et al.* [1] proposed a distributed PDP evaluation engine architecture, including a similarity calculator, access request repeater, and rule matcher. The access requests are forwarded to the appropriate rule matcher, which loads a small set of policies and evaluates the access requests received.

2.4 Optimization of Mass Policy Sets

Directing at the optimization of mass policy sets, Mourad *et al.* [17] proposed a new XACML framework based on set algebra: SBA-XACML. This framework converts XACML policies/requests into SBA-XACML policies/requests, and the evaluation module evaluates the transformed requests, which is better than XEngine [11] in performance. Deng *et al.* [2] established the attribute bitmap and determined the rules corresponding to the access request through the AND operation of the attribute bitmap. Based on the automata theory, a policy decision engine, including a decision machine and a evaluation machine, was proposed. Han Daojun *et al.* [8] proposed an XACML policy query method based on attribute and/or matrix and type analysis to calculate each rule attribute's discrimination degree. Besides, they used the discrimination degree and attribute and/or matrix to screen out rules unrelated to the current access request. The XACML evaluation engine proposed by Ngo *et al.* [19] is based on data interval segmentation and aggregation and a new decision graph combination. However, its experimental policy set is small, including 360 rules at most, which cannot fully reflect the performance improvement effect under each rule number. The XEngine system [11] adopts a slightly different optimization approach than other evaluation engines. In this scheme, XACML policy rules' string attributes are mapped to numerical attributes to avoid inefficient string matching. Policies with hierarchical structures are converted to flat structures. Combining algorithms are converted to First-Applicable combining algorithms to avoid traversal matching of policies and rules. Deng *et al.* [3] proposed a rule dictionary, a data structure similar to an array. A rule's index can be determined quickly by using the rule dictionary. In addition, the bitmap is used to store the effect of rules in the policy set to save storage space. However, if the rule property values change, the rule dictionary's content structure and text structure needs to be updated. The bitmap stores only the effects of the rules, but the rules are still stored in a rule dictionary similar to a four-dimensional array. In the case of many rules, the memory overhead is large. Niu *et al.* [22] used the multi-level caching mechanism based on statistical analysis to store the most frequently called information in the cache.

In a summary, the above schemes to improve the performance of PDP evaluation only focus on the efficiency improvement, without considering the possible integrity problems of the policy itself. Therefore, this paper designs a scheme to improve the evaluation performance of PDP and guarantee the security of the decision system on the premise of ensuring the integrity of the policy.

3 Approach Overview

Our study includes policy authentication and policy matching optimization, as shown in Figure 1 and Figure 2. Policy integrity assurance depends on the verification data structure based on Merkle B Tree. The policy matching optimization engine improves PDP performance by optimizing matching processes.

We choose to adopt Merkle hash tree based Authenticated Data Structure (ADS), in particular, Merkle B Tree. MBT based ADS can ensure correctness, completeness, and afford efficient freshness assurance because only one root hash needs to be maintained correctly. The policies and authentication data are stored at PAP. The policy and verification object (VO) are returned when PDP queries for them so that PDP can verify the resulting policy's integrity.

When the PDP verifies that the policy is integrated, a Rule Trie based on the retrieved policy is built, which is the core of the Policy Matching Optimization Engine (PMOE). Before creating the Rule Trie, XACML policy normalization and numeralization are needed. Rule Trie based on policy is established for the efficient evaluation of PDP, which uses the common attribute prefix of the rule to reduce the matching time. When access requests arrive, they enter the numeric machine and are converted into numeric access requests. The access request then matches the rule in the Rule Trie, and the output result is the decision result of the PMOE.

4 Policy Authentication

4.1 Authenticated Data Structure

Plenty of methods have been proposed that utilize ADS for examining data integrity. The scheme based on signature aggregation requires modifying signatures of all the records, making it ineffective considering the number of signatures. Di Battista *et al.* [4] proposed the authenticated skip lists based approach. A modified Merkle Hash

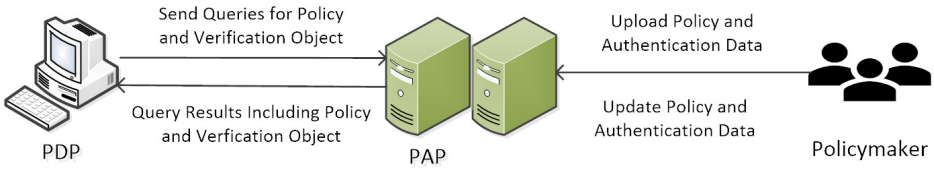


Figure 1: Policy outsource model

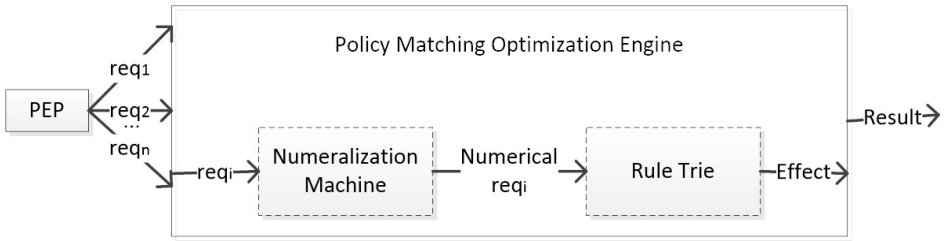


Figure 2: Policy matching optimization engine

Tree based scheme has been submitted by Goodrich *et al.* [7], which is named Merkle B Tree scheme. In contrast to Merkle Hash Tree, each non-leaf node of MBT can contain more than two child nodes.

Figure 3 displays a Merkle B Tree built based on the policy table. The values in the *policyId* column are adopted as keys in the MBT. A hash *hi* is correlated with a pointer in an internal node or a record in a leaf node. For the sake of simplification, the hashes associated with pointers and records in the MBT nodes are not displayed in the figure. The hash of a record in a leaf node is the hash value of the data record in the data table. The hash associated with a pointer in an internal node is the hash of concatenating all hashes in the nodes pointed by the pointer.

4.2 Identify Authentication Data

Radix Path Identifiers are adopted to identify pointers to internal nodes in MBT and records in leaf nodes. Its advantages include convenient storage of parent-child and sibling relationships and maintenance of the order of pointers or records in a node of MBT. Utilizing numbers based on a specified radix to identify each pointer or record in an MBT is the basic idea of Radix Path Identifiers. Figure 4 shows the Merkle B Tree with Radix Path Identifiers.

l denotes the level of the MBT. The level of the root node is 0, and the level of leaf nodes is the maximum. Let rb be the radix base. f represents the fanout of the MBT. rb is a random number equal to or greater than f of the MBT. i indicates the index of a pointer or a key in a node, which ranges from 0 to f. Radix Path Identifier RPI can be computed using the following equation:

$$RPI = \begin{cases} l & l = 0, \\ RPI_{parnet} * r_b + i & l > 0. \end{cases} \quad (1)$$

For instance, to determine the RPI of the key 6 in the leaf node, the level of the key is 2. i in the case is 1 for 6 is the second key in the node. RPI of the parent is 12 and rb is 3. So we can calculate the RPI of key 6 is 121.

The Radix Path Identifier scheme has several significant characteristics:

- From an RPI in a node, the RPI of its parent pointer can be calculated by $RPI_{parent} = \lfloor RPI/r_b \rfloor$
- From an RPI in a node, the min, and max RPIs are determined, which are $\lfloor RPI/r_b \rfloor * r_b$ and $\lfloor RPI/r_b \rfloor * r_b + (r_b - 1)$
- From an RPI in a node, the index i of the pointer or key in the node equals RPI mod rb.

4.3 Store and Extract Authentication Data

Radix Path Identifiers Based MBT are saved in a database in two ways. One way is to save all the data in a single validation table, named Single Authentication Table (SAT). Another approach is to keep each layer of MBT in a table called Level-Based Authentication Table (LBAT).

It is quicker to update authentication data in LBATs than SAT due to the existence of the table level locks during updates and insertions. In the LBAT, authentication data is saved along with the data record, making it straightforward to retrieve authentication data for the leaf level and the required table data.

The LBAT storage form of the Radix Path Identifiers based MBT is shown in Table 1. In the leaf node of the MBT, the ID column stores values from the policy table's policyId column. As the number of pointers in the internal nodes is more than the number of keys, -1 is used as the far left pointers in the internal nodes. RPI stores identifiers for pointers or records in the MBT. *hash*

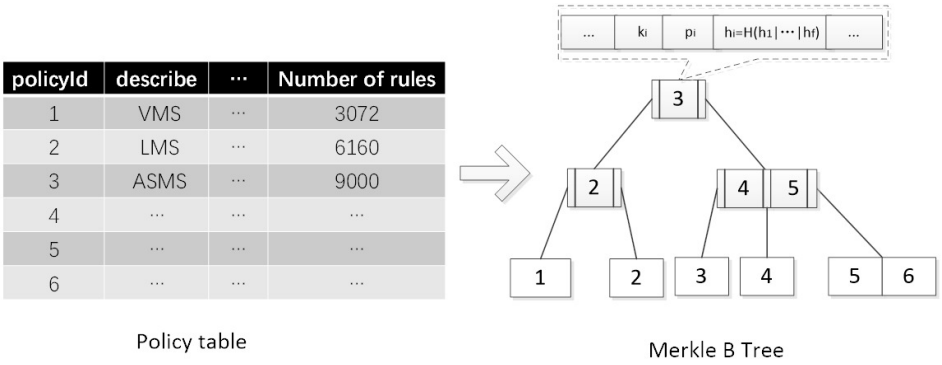


Figure 3: Policy table to Merkle B tree

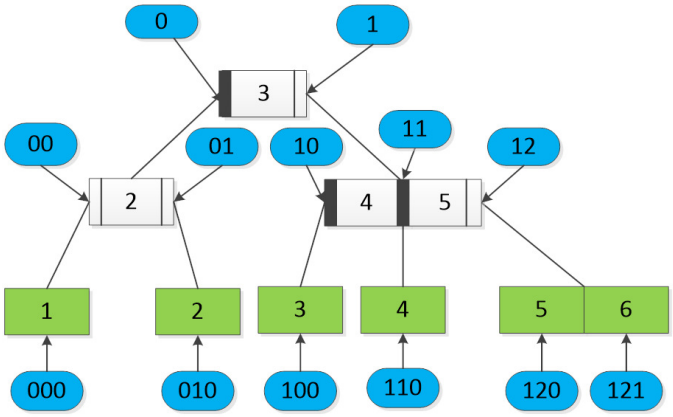


Figure 4: Merkle B tree with radix path identifiers

column reserves the hash values of pointers or records in the MBT, critical to integrity verification.

We extract authentication data from LBAT to compute the root hash of the MBT. If we want to verify the authenticity of the policy with id 6, we need to extract all the data on the authentication path from the LBATs. Pointers on the authentication path have been shown in black in Figure 4. For data extraction from the LBAT table, four different methods have been given: Multi-Join, Single-Join, Zero-Join, and Range-Condition in [28].In these methods, we use Zero-Join to demonstrate extracting data from LBAT to verify data records' authenticity with ID 6. As the name implies, querying authentication data does not require joining any tables. Each table is queried independently. First, we extract the *RPI* of the record to be queried and store it in a variable. This variable is then employed to extract authentication data from the LBATs.

```
declare @var as int;
select @var = RPI from Policy where ID = 6;
select RPI, hash from Policy where RPI/3=@var/3;
select RPI, hash from Policy1 where RPI/3=@var/(3*3);
select RPI, hash from Policy2 where RPI/3=@var/(3*3*3);
```

Table 1: Level based authentication tables

Policy2 (root)		
ID	RPI	hash
-1	0	hash
3	1	hash

Policy1		
ID	RPI	hash
-1	0	hash
2	1	hash
-1	3	hash
4	4	hash
5	5	hash

Policy (Leaf Nodes)				
ID	describe	numberOfRules	RPI	hash
1	VMS	3072	0	hash
2	LMS	6160	3	hash
3	ASMS	9000	9	hash
4	continue-a	298	12	hash
5	continue-b	306	15	hash
6	pluto	21	16	hash

5 Policy Preprocessing and Matching Optimization

When the PMOE receives the policy, it verifies whether the policy is integrated through the verification object. If it is integrated, the policy is preprocessed, including two stages of policy normalization and numericalization. Rule Trie for all rules is created after preprocessing.

When an access request arrives at PMOE, it is first processed by a numerical machine and converted into a numeric access request. Then the numerical access request matches the applicable rule in Rule Trie to get the decision result. The processing time of the access request contains the time to digitize the access request and match the applicable rules in Rule Trie.

5.1 Policy Normalization and Numericalization

The process of XACML policy normalization is to turn an XACML policy with a hierarchical structure into an equivalent policy with a flat structure. Besides, convert an XACML policy with four rule combining algorithms into an equivalent policy with only one rule combining algorithm, which is First-Applicable. Furthermore, We extract effect, subject attribute, resource attribute, action attribute, and condition attribute from the rules to simplify the rules into five tuples. The simplified XACML rule is shown in Figure 5. The converted rule table is shown in Table 2.

```
<Rule RuleId="1" Effect="Deny">
    <Target>
        <Subjects><Subject>subAttr1</Subject></Subjects>
        <Resources><Resource>resAttr1</Resource></Resources>
        <Actions><Action>acAttr1</Action></Actions>
    </Target>
    <Condition>conAttr1</Condition>
</Rule>
```

Figure 5: Simplified XACML rule

Table 2: Converted rule table

Permit	subject1	resource3	action5	condition1
Deny	subject0	resource1	action0	condition2
Deny	subject1	resource1	action1	condition2

Due to the normalization technique, our XACML policy matching optimization engine avoids comparing the request against all the rules in an XACML policy when processing an XACML request.

We convert the string values in an XACML policy into integer values during XACML policy numericalization. HashMap is adopted to store the string value and integer value. When processing XACML requests, our XACML policy evaluation engine uses the efficient integer comparison without the inefficient string matching due to this numericalization technique.

5.2 Rule Trie

After normalizing and digitizing the policies, the rule table is obtained. However, searching for applicable rules in a large rule table causes a high time cost and reduces the evaluation performance of the PDP. Considering that many rules have the same attribute prefix, we adopt the idea of Trie and establish Rule Trie to reduce the query time of matching rules. Rule Trie has the following nodes:

- Internal node: It is a node containing the pointers pointing to the next node. The pointers are associated with the attribute values.
- End node: It represents the end of the rule and stores the effect of the rule.

Algorithm 1 InsertRule

input: R: the rule to be inserted
root: the root node of Rule Trie

```
1:  TrieNode node = root; // the working node
2:  int [] nums = new int[4];
3:  nums[0] = subjectMap.get(R.subject);
4:  nums[1] = resourceMap.get(R.resource);
5:  nums[2] = actionMap.get(R.action);
6:  nums[3] = conditionMap.get(R.condition);
7:  for num ∈ nums do
8:    if ! node.containsKey(num) then
9:      node.put(num , new TrieNode());
10:   end if
11:   node = node.get(num);
12: end for
13: node.setEnd();
14: node.setEffect(R.effect);
```

Algorithm 2 EstablishRuleTrie

input: ruleList: all the rules in a policy
output: root: the root node of the established Rule Trie

```
1: TrieNode root = new TrieNode();
2: for rule ∈ ruleList do
3:   InsertRule(rule);
4: end for
5: return root;
```

Take the rightmost leaf node in Figure 6 as an example, which is an end node representing the rule (subject=1, resource=1, action=0, condition=2, effect=Deny). It has the same attribute prefix as the sibling node on the left. The same attribute prefix is stored only once and does not need to be restored. The height of Rule Trie is 4, which means that querying a rule in Rule Trie can be completed

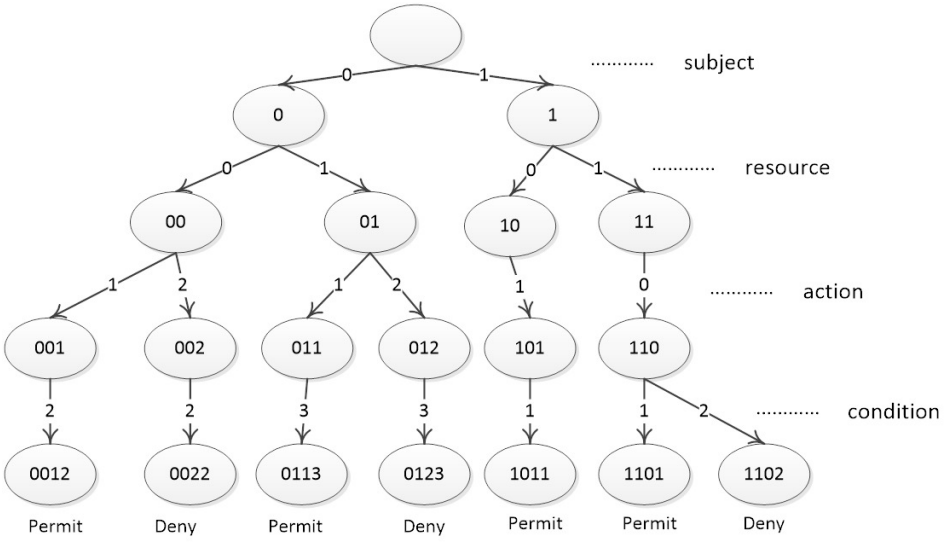


Figure 6: Rule trie

Algorithm 3 SearchRule

input: req: a numerical request
output: res: a bool variable indicating whether the matching rule exists

```
// If the matching rule exists, the function returns true
TrieNode node = root; // the working node
int [] nums = new int[4];
nums[0] = req.subject;
nums[1] = req.resource;
nums[2] = req.action;
nums[3] = req.condition;
for num ∈ nums do
    if node.containsKey(num) then
        node = node.get(num);
    else
        node = null; break;
    end if
end for
res = node != null && node.isEnd();
return res;
```

in constant time, i.e., the query time complexity is O(1). Here we present some concepts and functions needed to better understand the establishment of Rule Trie algorithm as shown in Algorithms 1, 2 and searching for the matching rule algorithm as shown in Algorithm 3.

- TrieNode: It is a node of the rule Trie. It contains member variables, including links, isEnd, and effect.
- links: It is a pointer array, which contains the pointers pointing to the current node's children nodes.
- isEnd: It is a bool member variable, which marks whether the current node is an end node.
- effect: It is a string variable that represents the rule effect when the current node's isEnd is true.
- containsKey(int x): It is a function to judge whether the current node contains the pointer associated with the attribute value x. If the current node contains, it returns true.
- put(int x, TrieNode newNode): It is a function to make the pointer associated with the attribute value x pointer to the new node.
- get(int x): It is a function to get the pointer associated with x in the current node.

6 Experiments Results and Analysis

The experiments are divided into two parts. The first part is to explore the time cost of retrieving verification objects in the PAP's database, and the second part is to verify that the proposed Rule Trie-based policy matching optimization engine can improve the performance of PDP evaluation. In the second part, we introduce the

policies adopted in the experiment, the method of generating access requests, the result of preprocessing, and the comparison of the evaluation performance of PMOE with other evaluation engines.

The experiments are carried out on a laptop computer running Windows 10, with an Intel Core i7-6700H 2.6 GHz processor and 8GB of RAM. Our PMOE is implemented in Java. The Java Runtime Environment version is 1.8.0 and the MySQL version is 8.0.18.

6.1 Retrieve the Verification Object

In the policy outsourcing model, when the PDP needs to request the policy and the verification object, the PAP needs to retrieve the verification object from the policy database and return it to the PDP together with the policy. If the verification object retrieval time is long, the evaluation performance of PDP will be affected. Therefore, the experiment in this section explores the time cost of retrieving verification objects in the PAP's database.

In the experiment, we convert 1000, 4000, 7000, and 10000 policies into Merkle B Tree with a fanout of 3 respectively. Select Radix as 3 and establish Radix Path Identifier for the pointer of each internal node and record of the leaf node. We use the LBAT method to store Merkle B Tree, the MBT being serialized into the database via JDBC technology. Then we adopt the Zero-Join method to extract validation data. As can be seen from Figure 7, the retrieval time of VO increases with the increase of policies. When the number of policies reaches 10000, the VO retrieval time is only about 22 ms. It can be seen that the verification data structure based on Merkle B Tree does not bring a large retrieval time delay.

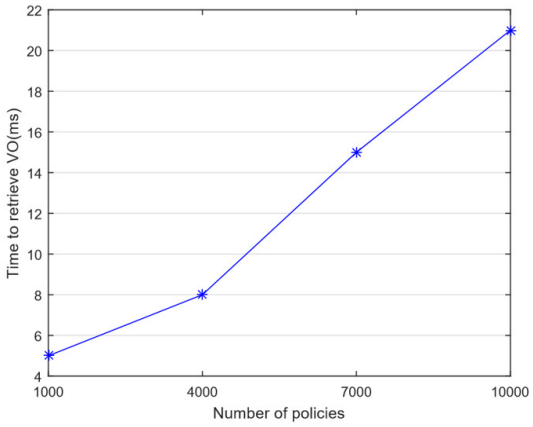


Figure 7: Retrieve VO

6.2 Test Policies

We choose the following three XACML access control policies from practical systems:

- Library management system [9] (LMS): The LMS offers access control policies by which a public library can adopt the web to manage books.
- Virtual meeting system [15] (VMS): The VMS provides access control policies by which the web conference services can be under management.
- Auction sale management system [16] (ASMS): The ASMS offers access control policies by which items can be bought or sold online.

The LMS policy contains 720 rules, the VMS policy includes 945 rules, and the ASMS contains 1760 rules.

For the sake of practical purposes, the three policies need to be extended to contain more rules. Therefore, we extract the subject attribute set, resource attribute set, action attribute set, and condition set from the original policy. Then we take the Cartesian product of these four sets, construct new rules, and add them into the original policy. After expansion, the three policies include 6160, 3072, 9000 rules respectively.

6.3 Generation of Test Requests

A target-driven approach to generating access requests is proposed by Martin and Xie [14], which considers each rule independently. Collect attribute ID-value pairs of subject, resource, and action in target and form predicates. A request set is generated that satisfies all possible combinations of truth values for each independent clause. We adopt the Simple Combinatorial scheme and the construction of access requests is based on properties in policy set rules. The policy set is parsed to obtain the subject attribute set, resource attribute set, action attribute set, and condition attribute set. The access request is obtained by taking the Cartesian product of the four sets.

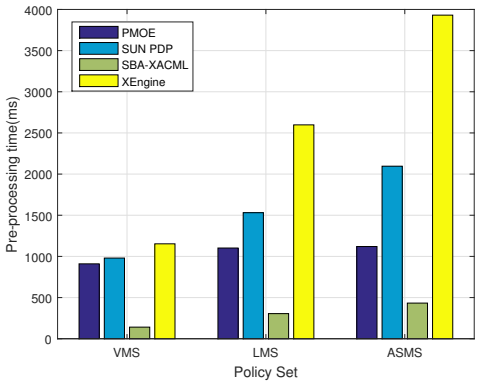
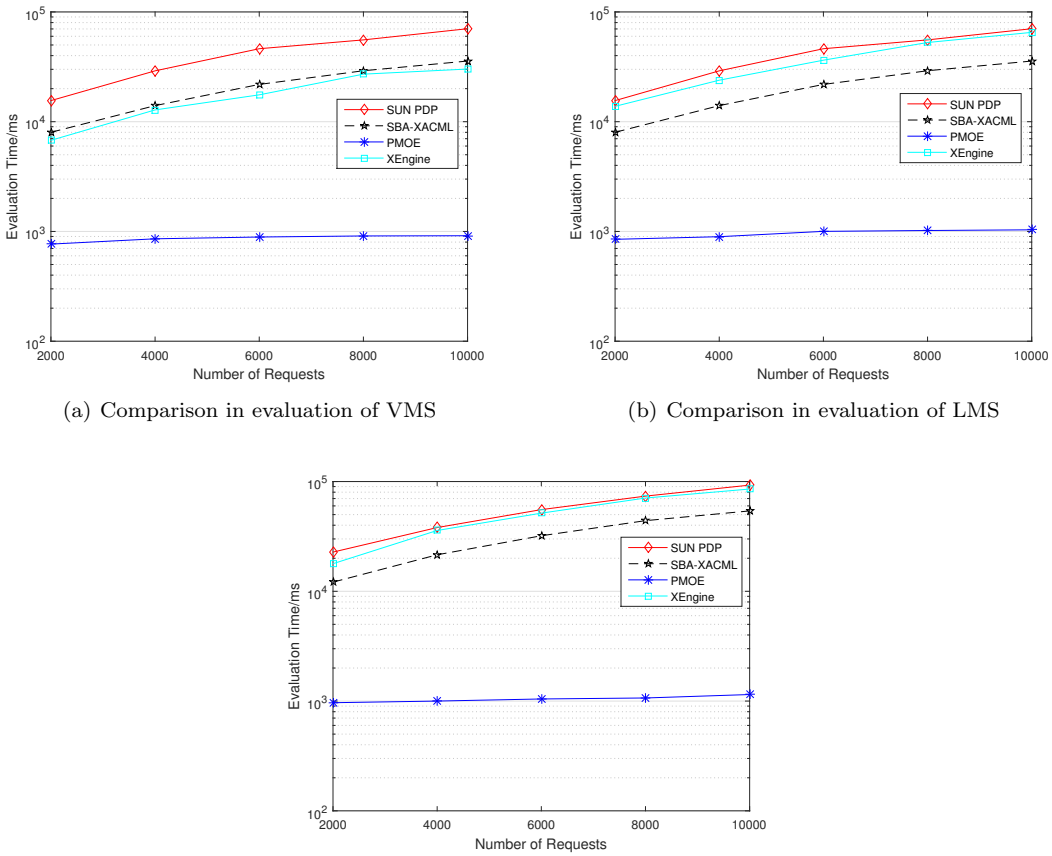


Figure 8: Average preprocessing time

6.4 Preprocessing Result

For each evaluation engine, we perform preprocessing experiments 50 times on VMS, LMS, and ASMS respectively. We record the average preprocessing time for each evaluation engine in the same policy set, as shown in Figure 8.



(a) Comparison in evaluation of VMS

(b) Comparison in evaluation of LMS

(c) Comparison in evaluation of ASMS

Figure 9: Comparison of evaluation performance

As can be seen from Figure 8, XEngine's preprocessing takes the longest time, which is mainly spent on policy numericalization, normalization, and converting the policy into a Policy Decision Diagram. The policy preprocessing stage of SBA-XACML is to convert the XACML form of the policy into SBA-XACML, and the time measurement includes the time of converting and the time of loading the policy. The preprocessing time of PMOE includes policy loading, numericalization, normalization, and establishment of Rule Trie. Although the preprocessing time of PMOE is slightly longer than that of SBA-XACML, the policy pre-processing is performed only once, and PMOE is 1-2 orders of magnitude faster in evaluating performance than SBA-XACML.

6.5 Performace Tests and Comparisons

To evaluate the performance of PMOE, we compare it with the standard Sun PDP implementation, XEngine, and SBA-XACML. We choose them in our experiments because they are all open source. Sun PDP is the first and most widely used XACML evaluation engine and is an industry standard. XEngine transfers the hierarchical structure of an XACML policy to a flat structure and adopts a tree structure to hold the corresponding numerical values of attributes in rules, which is aimed at improving the efficiency of evaluation. SBA-XACML is a novel Set-Based Algebra (i.e. SBA) scheme that provides an efficient evaluation of XACML policies. It has its SBA-XACML grammar, and the scheme converts the XACML policy/request to SBA-XACML policy/request. SBA-XACML contains formal semantics and algorithms that take advantage of the mathematical operations to provide efficient policy evaluation. The evaluation time of different evaluation engines in VMS, LMS, and ASMS policies is measured in the experiment. We generate 2000, 4000, ..., 10000 access requests randomly to calculate the evaluation performance of the PDP. The experimental results are shown in Figure 9. Because Sun PDP uses the brute-force method to find the applicable rules in the policy library, the evaluation time is long and inefficient. Because of this, SBA-XACML, XEngine and PMOE are designed to improve the evaluation performance of Sun PDP. SBA-XACML aims to provide a new syntax for describing XACML based on collection logic and defines a new evaluation module. XEngine is devoted to numerical and normalized policy and realizes efficient processing of requests via a tree data structure. PMOE is working on a new evaluation engine that includes a numerical machine and Rule Trie. In Figure 9, we observe that

- The evaluation time of Sun PDP, SBA-XACML, XEngine and PMOE rises when the number of access requests grows.
- The growth rate of the evaluation time of PMOE is less than that of Sun PDP, XEngine and SBA-XACML. Sun PDP takes more time than others as it needs to compare one request with every rule in a policy set when evaluating access requests.
- For policy VMS, LMS and ASMS, when access requests reach 10000, Sun PDP's evaluation time is almost 44 times, 70 times, and 80 times that of PMOE.
- The evaluation time of PMOE is much lower than that of Sun PDP. The average evaluation time of SBA-XACML is shorter than that of Sun PDP but still longer than that of PMOE.

7 Conclusions and Future Work

In this paper, we propose an integrity-preserving scheme and an efficient policy matching engine for XACML, aiming at secure and efficient evaluation. We use Merkle B Tree as the authorization data structure for the policy and Radix Path Identifiers for efficient storage and retrieval of verification data in the database. Moreover, we design a policy matching optimization engine, whose core is Rule Trie. Since lots of rules have the same attribute prefix, building the rules into a Trie can accelerate the matching process of requests. Experimental results verifies that our PMOE has better performance than Sun PDP and SBA-XACML. Since our PMOE only supports rules of single-valued attributes, in the future we will focus on evaluating multi-valued rules.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61772327), State Grid Gansu Electric Power Company (No. H2019-275), and Shanghai Engineering Research Center on Big Data Management System (No.H2020-216)

References

[1] F. Deng, J. Lu, S. Y. Wang, J. Pan, and L. Y. Zhang, "A distributed pdp model based on spectral clustering for improving evaluation performance," *World Wide Web*, vol. 22, no. 4, pp. 1555–1576, 2019.

[2] F. Deng, S. Y. Wang, L. Y. Zhang, X. Q. Wei, and J. P. Yu, "Establishment of attribute bitmaps for efficient xacml policy evaluation," *Knowledge-Based Systems*, vol. 143, pp. 93–101, 2018.

[3] F. Deng, L. Y. Zhang, C. Y. Zhang, H. Ban, C. Wan, M. H. Shi, C. Chen, and E. T. Zhang, "Establishment of rule dictionary for efficient xacml policy management," *Knowledge-Based Systems*, vol. 175, pp. 26–35, 2019.

[4] G. Di Battista and B. Palazzi, "Authenticated relational tables and authenticated skip lists," in *IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 31–46. Springer, 2007.

[5] D. Diaz-Lopez, G. Dolera-Tormo, F. Gomez-Marmol, and G. Martinez-Perez, "Managing xacml systems in distributed environments through meta-policies," *Computers & Security*, vol. 48, pp. 92–115, 2015.

[6] D. El Kateb, T. Mouelhi, Y. Le Traon, J. H. Hwang, and T. Xie, "Refactoring access control policies for performance improvement," in *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*, pp. 323–334, 2012.

[7] M. T Goodrich, R. Tamassia, and N. Triandopoulos, "Super-efficient verification of dynamic outsourced databases," in *Cryptographers' Track at the RSA Conference*, pp. 407–424. Springer, 2008.

[8] D. J. et al. Han, "Xacml polcicy query method based on attribute and/or matrix and type analysis," *COMPUTER SCIENCE(in Chinese)*, vol. 45, no. 9, pp. 224–229, 2018.

[9] Y. Le Traon, T. Mouelhi, A. Pretschner, and B. Baudry, "Test-driven assessment of access control in legacy applications," in *2008 1st International Conference on Software Testing, Verification, and Validation*, pp. 238–247. IEEE, 2008.

[10] B. Li, Q. He, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Auditing cache data integrity in the edge computing environment," *IEEE Transactions on Parallel and Distributed Systems*, 2020.

[11] A. X. Liu, F. Chen, J. H. Hwang, and T. Xie, "Xengine: a fast and scalable xacml policy evaluation engine," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 1, pp. 265–276, 2008.

[12] J. Mao, Y. Zhang, P. Li, T. Li, Q. H. Wu, and J. W. Liu, "A position-aware merkle tree for dynamic cloud data integrity verification," *Soft Computing*, vol. 21, no. 8, pp. 2151–2164, 2017.

[13] S. Marouf, M. Shehab, A. Squicciarini, and S. Sundareswaran, "Adaptive reordering and clustering-based framework for efficient xacml policy evaluation," *IEEE Transactions on Services Computing*, vol. 4, no. 4, pp. 300–313, 2010.

[14] E. Martin, "Automated test generation for access control policies," in *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, pp. 752–753, 2006.

[15] T. Mouelhi, F. Fleurey, B. Baudry, and Y. Le Traon, "A model-based framework for security policy specification, deployment and testing," in *International Conference on Model Driven Engineering Languages and Systems*, pp. 537–552. Springer, 2008.

[16] T. Mouelhi, Y. Le Traon, and B. Baudry, "Transforming and selecting functional test cases for secu-

rity policy testing," in *2009 International Conference on Software Testing Verification and Validation*, pp. 171–180. IEEE, 2009.

[17] A. Mourad and H. Jebbaoui, "Sba-xacml: set-based approach providing efficient policy decision process for accessing web services," *Expert systems with applications*, vol. 42, no. 1, pp. 165–178, 2015.

[18] M. C. Muñoz, M. Moh, and T. S. Moh, "Improving smart grid security using merkle trees," in *2014 IEEE Conference on Communications and Network Security*, pp. 522–523. IEEE, 2014.

[19] C. Ngo, Y. Demchenko, and C. De Laat, "Decision diagrams for xacml policy evaluation and management," *Computers & Security*, vol. 49, pp. 1–16, 2015.

[20] D. M. Nguyen, Q. H. Luu, N. Huynh-Tuong, and H. A. Pham, "Mb-pba: Leveraging merkle tree and blockchain to enhance user profile-based authentication in e-learning systems," in *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 392–397. IEEE, 2019.

[21] M. S. Niaz and G. Saake, "Merkle hash tree based techniques for data integrity of outsourced data.," in *GvD*, pp. 66–71, 2015.

[22] D. H. Niu, "Hpengine:high performance xacml policy evaluation engine based on statistical analsysis," *Journal on Communications(in Chinese)*, vol. 35, no. 8, pp. 206–215, 2014.

[23] X. Pei, H. Q. Yu, and G. S. Fan, "Achieving efficient access control via xacml policy in cloud computing.," in *SEKE*, pp. 110–115, 2015.

[24] B. Sharma, C. N Sekharan, and F. Y. Zuo, "Merkle-tree based approach for ensuring integrity of electronic medical records," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 983–987. IEEE, 2018.

[25] H. Tohidi and V. T. Vakili, "Lightweight authentication scheme for smart grid using merkle hash tree and lossless compression hybrid method," *IET Communications*, vol. 12, no. 19, pp. 2478–2484, 2018.

[26] Y. Z. Wang, Y. L. Shen, H. Wang, J. L. Cao, and X. H. Jiang, "Mtmr: Ensuring mapreduce computation integrity with merkle tree-based verifications," *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 418–431, 2016.

[27] W. Wei and T. Yu, "Integrity assurance for outsourced databases without dbms modification," in *IFIP Annual Conference on Data and Applications Security and Privacy*, 2014.

[28] W. Wei and T. Yu, "Integrity assurance for outsourced databases without dbms modification," in *IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 1–16. Springer, 2014.

[29] Y. P. Zhang and B. B. Zhang, "An aco-based algorithm for efficient xacml policy evaluation," in *2017 2nd International Conference on Control, Automation and Artificial Intelligence (CAAI 2017)*. Atlantis Press, 2017.

Biography

Kai Zheng Graduate. College of Computer Science and Technology in Shanghai University of Electric Power. His research interests mainly focus on the XACML.

Xiuxia Tian received the MS degree in applied cryptography-based information security from Shanghai Jiaotong University in 2005, and the PhD degree in database security and privacy preserving in cloud computing from Fudan University in 2011. She is currently a professor in the College of Computer Science and Technology, Shanghai University of Electric Power. She is a visiting scholar of two years at UC Berkeley working with groups of SCRUB and SecML. She has published more than 40 papers and some papers are published in international conferences and journals such as DASFAA, ICWS, CLOUD, and SCN. Her main research interests include database security, privacy preserving (large data and cloud computing), applied cryptography, and secure machine learning.

Towards Forward Secure Conjunctive Searchable Symmetric Encryption with Result Pattern Hidden

Yunling Wang[1], Yichao Zhu[2], and Jianfeng Wang[2]
(Corresponding author: Jianfeng Wang)
School of Cyberspace Security, Xi'an University of Posts and Telecommunications[1]
Xi'an 710121, China
State Key Laboratory of Integrated Service Networks, Xidian University[2]
Xi'an 710071, China
Email: jfwang@xidian.edu.cn
(Received Apr. 17, 2021; Revised and Accepted Oct. 26, 2021; First Online Feb. 26, 2022)

Abstract

Searchable Symmetric Encryption (SSE) enables a client to outsource encrypted data securely while supporting efficient keyword search. The secure forward SSE (FSSE) has recently attracted widespread attention due to its security properties for data updates. For example, it can prevent the server from knowing whether the newly updated document contains the previously queried keywords. However, most of the existing FSSE schemes are designed for single-keyword search. As a result, the existing FSSE schemes leak significant information or require inefficient primitives regarding the conjunctive keyword search. To this end, we propose forward secure conjunctive keyword search schemes without result pattern leakage. In particular, we propose a baseline conjunctive keyword FSSE scheme by combining the resulting pattern hiding conjunctive keyword search scheme (HXT) with a single keyword FSSE. However, it can only support document updates. Therefore, it only allows the data owner to search after all the keyword-documents pairs in the document have been updated. Then we proposed a new conjunctive keyword FSSE scheme with a result pattern hidden to support flexible keyword-document pair update, in which the data owner can perform a search at any time. Finally, we implemented our scheme, and the results show that our solution is efficient for real-world applications.

Keywords: Conjunctive Keyword Search; Forward Security; Searchable Symmetric Encryption

1 Introduction

With the development of cloud computing, more and more data is outsourced to the cloud server. However, the cloud server may not always be trustworthy, and the security issues of the outsourced data have become the key restriction for the development of cloud computing. To protect data security, users encrypt the data before outsourcing and then store the data in the cloud server in the form of ciphertext [5]. Although traditional encryption technology can guarantee the confidentiality of outsourced data, it lost the function of searching over the encrypted data. A simple way to perform search is to download the ciphertexts locally and then decrypt them for searching. However, it costs dramatically communication and computation overheads. To this end, searchable encryption scheme (SSE) is proposed as a promising solution to perform search over encrypted data.

Song *et al.* [16] first proposed the concept of SSE in 2000. In their scheme, the user encrypts each keyword in the document to build an index. When the data owner is interested in some keyword, s/he encrypts the keyword into a trapdoor and sends to the cloud server. The cloud server searches over the ciphertext according to the search trapdoor, and then returns the matched documents to the client. This scheme solves the problem of ciphertext retrieval in the cloud server, but the search complexity is linear in the number of keywords in all the documents. Subsequently, the research of the SSE scheme mainly focused on efficiency, security, and expressiveness, and achieved a series of results [4, 6, 7, 12, 13]. Recently, researchers have paid great attention to the security of SSE schemes. For example, several attacks [3,11,23] have demonstrated the use of common vulnerabilities in the SSE scheme to compromise query privacy. In particular, the powerful file injection attack was proposed by Zhang *et al.* [23], which can completely recover the queried keyword by injecting a few carefully selected files.

The file injection attack emphasizes the importance of forward security, which requires that a newly added document cannot be linked with the previous search queries.

The notion of forward secure SSE was first introduced by Stefanov *et al.* [18], the FSSE scheme they proposed uses an ORAM-like structure, resulting in huge communication overhead. In 2016, Bost *et al.* [2] constructed the FSSE scheme $\Sigma_{o\varphi o\varsigma}$ using trapdoor permutation, which improves the search and update efficiency. Subsequently, Song *et al.* [17] used symmetric encryption to replace the public key primitive in the trapdoor permutation of $\Sigma_{o\varphi o\varsigma}$, which further improved search and update efficiency. However, the above schemes are mainly designed for single keyword search.

In practice, the conjunctive keyword query is more practical. A simple idea is to search for each keyword separately and return the intersection of the single keyword search results. This approach is not only inefficient, but also leaks too much information. In 2013, Cash *et al.* [4] proposed a well-designed conjunctive keyword search scheme OXT. When performing search, the client selects the keyword with the least frequency to perform single keyword search, that is, the least frequency keyword has the least number of documents among all the queried keywords. Finally, the server judges whether the search result for the least frequency keyword contains other queried keywords. This scheme has sub-linear search complexity so that it can be well applied to large databases. In 2018, Lai *et al.* [15] proposed the hidden cross tags protocol (HXT) to eliminate the leakage of keyword pair result pattern (KPRP) in OXT. This can be achieved by using the primitive of Hidden Vector Encryption which can determine whether a document contains some keywords at the same time.

Similar to the single keyword search scheme, the conjunctive keyword search also faces security challenges. Therefore, many researchers have focused on the forward security of conjunctive keyword search schemes. Wu and Li [22] designed an FSSE scheme that supports conjunctive keywords search based on a tree structure. However, this scheme leaks the results of each search keyword. Hu *et al.* [10] used inner product encryption to construct an FSSE scheme that supports conjunctive keywords search to avoid leaking the results of each search keyword. However, the introduction of inner product encryption technology makes the scheme inefficient. Recently, Wang *et al.* [21] adopted the Bost single keyword FSSE scheme and the idea of OXT to construct an efficient FSSE scheme that supports conjunctive keywords search, which solved the above problems. Nevertheless, this scheme still has the KPRP leakage for conjunctive keyword search.

1.1 Our Contributions

In this paper, we constructed two conjunctive keyword FSSE schemes FHXT-B and FHXT-E without KPRP leakage. The main contributions can be summarized as follows:

1) We first construct a basic conjunctive keyword FSSE scheme named FHXT-B. The main idea is that we combine the conjunctive keyword search scheme HXT with keyword pair result pattern hidden and single keyword FSSE scheme. Therefore, our FHXT-B scheme achieves forward secure conjunctive keyword search without keyword pair result pattern leakage. However, FHXT-B is limited to supporting document updates, that is, users are not allowed to perform search operation until all the keyword-document pairs in a document are updated.

2) We then propose an enhanced scheme FHXT-E to solve the above limitation. In FHXT-E, we temporarily store the updated ciphertext in a forward security structure, and update the ciphertext when performing search. This enhanced scheme can support flexible keyword-document pair updates, that is, users can perform search operation at any time without waiting for the whole document to be updated.

3) We provide formal security analysis and thorough implementation. The results show that our schemes can achieve the desired security goal with good efficiency.

1.2 Related Work

Searchable symmetric encryption was first proposed by Song *et al.* [16] in 2000. This scheme encrypts each keyword in the document, and then uses keywords to match the encrypted document when searching. The search complexity is linear to the number of keywords in the whole documents. In order to improve search efficiency, Goh *et al.* [8] used Bloom Filter (BF) to construct a forward index, and then hashed the keywords in the document and stored it in the index. The Bloom filter is used to determine whether the searched keywords are included in the encrypted document. The search efficiency was improved, which is linear in the number of documents. To further improve the search efficiency, Curtmola *et al.* [6] constructed an SSE scheme based on the inverted index. This scheme achieves sub-linear search complexity and gives a formal security definition for SSE. After that, the researchers conducted an in-depth study on the safety, performance, and expressiveness of SSE [7, 12, 19, 20, 24].

To rich the expressiveness, Golle *et al.* [9] proposed the first conjunctive keyword search scheme. However, the search complexity is linear to the number of documents, and there are a large number of modular exponential operations and two-line pair operations. In 2013, Cash *et al.* [4] proposed the first sub-linear conjunctive keyword search scheme OXT. After that, Lai *et al.* [15] further compressed the leakage, and proposed a new conjunctive keyword SSE protocol based on the primitive of hidden vector encryption, called HXT. This scheme removes the keyword pair result pattern leakage, which can determine whether some keywords in a document at a time.

In dynamic setting, SSE enables the client to add and delete data. In 2012, Kamara *et al.* [14] proposed the first dynamic searchable symmetric encryption (DSSE) scheme. Yet this scheme has a high complexity for the add and delete operations. Stefanov *et al.* [18] introduced the

forward and backward security for addition and deletion, they proposed the first forward secure SSE (FSSE) scheme based on hierarchical ORAM. Subsequently, Zhang *et al.* [23] proposed a powerful file injection attack that can use the information leaked during the add operation to recover query keywords. The attack highlights the importance of forward security. Bost *et al.* [2] used trapdoor permutation to construct the FSSE scheme $\Sigma_{o\varphi o\varsigma}$, which has the best search and update complexity. After that, Song *et al.* [17] used symmetric encryption to replace the public key primitive in the trapdoor permutation of $\Sigma_{o\varphi o\varsigma}$, greatly improved search and update efficiency. Recently, there have been some researches on conjunctive keyword FSSE. Hu *et al.* [10] used inner product encryption to construct a conjunctive keywords FSSE scheme. Wu and Li [22] designed a conjunctive keywords FSSE scheme based on a tree structure. However, this scheme will leak the results of each search keyword. Wang *et al.* [21] combined the scheme of Bost [2] with the OXT to construct a conjunctive keywords FSSE scheme. However, all of the above schemes have KPRP leakage for conjunctive keyword search.

1.3 Organization

The rest of this paper is organized as follows. In Section 2, some preliminaries will be given. In Section 3, we propose our conjunctive keyword FSSE schemes. In Section 4, we provide the security and comparison analysis. In Section 5, we give the performance evaluation. Finally, we make the conclusion in Section 6.

2 Preliminaries

In this section, we first give a list of notations (as shown in Table 1) and then present some basic primitives which are used in our work.

Table 1: Notations in this work.

Notation	Meaning
λ	a security parameter
id_i	the document identifter of the i-th document
w	a keyword
$\mathrm{W}(id)$	the keyword set of id
w_i	the i-th keyword in W
$\{H_i\}_{1\le i\le k}$	k independent hash functions of BF
F	a pseudorandom function
$s \xleftarrow{\$} S$	uniformly sample a random element s from S

2.1 Bloom Filters

Bloom filters (BF) can be used to detect whether an element is in a set. Its advantage is that the space efficiency and query time far exceed the general algorithm. BF is composed of a binary vector $\mathbf{b}$ with m bits and a series of hash functions $\{H_i\}_{1\le i\le k}$. Put a set $S=\{s_1,s_2,\ldots,s_N\}$ of N elements into the BF, for each element $s\in S$, map the element to k points in the binary vector $\mathbf{b}$ by calculating $\{H_i(s)\}_{1\le i\le k}$, and set them to 1.When retrieving element q, we only need to see if these points are all 1 to know if q in the set: if any of these points has a 0, q must not be there; if they are all 1, q is very likely in. This is the basic idea of BF. If q is not in set S, the retrieved result is 1, we call it a "false positive" event. The false-positive probability is:

$$P_e \le (1-e^{-k\cdot N/m})^k. \tag{1}$$

2.2 Searchable Symmetric Encryption

A symmetric searchable encryption scheme consists of two entities: client and server. The user is responsible for encrypting files, generating encrypted indexes, generating search tokens, and decrypting ciphertexts. The server is responsible for searching the encrypted index according to the search token. In dynamic searchable symmetric encryption (DSSE) scheme, the user can update the data outsourced to the server. A DSSE scheme consists of the following algorithms:

Setup(1^λ, DB). This algorithm runs by the client, takes a security parameter λ and a database DB as inputs, outputs (K, σ, EDB), where K is a secret key, σ is the state of client, and EDB is the encrypted database.

Search(K, $\overline{w},\sigma$; EDB). This is a search protocol, runs between client and server. The client takes as input the secret key K, the state σ and the search keywords $\overline{w}$, outputs the search token to the server. The server's input is the encrypted database EDB. When the server receives the search token, it performs search over the EDB, and then returns the search results to the client.

Update(K, σ, op, in; EDB). a update protocol, runs between client and server. The client's input is (K, σ, op, in), where op is the update operation and in is an input (id, W(id)) or (w, id).

2.3 Symmetric-key Hidden Vector Encryption

Hidden vector encryption (HVE) is a predicate encryption scheme that provides fine-grained access control and supports conjunctive, comparison, and subset queries on encrypted database. HVE was formally defined in the public-key setting in [1]. In [15], Lai *et al.* introduced a symmetric-key hidden vector encryption (SHVE) scheme. The details of the construction are as follows: Σ is a

finite field $\mathbb{Z}_p$, where p is a prime. * is a wildcard symbol not in Σ , $\Sigma_* = \Sigma \cup \{*\}$. $\mathcal{P}^{SHVE} : \Sigma^m \to \{0,1\}$ is a family of predicates. For each $\mathbf{v} = (v_1, \ldots, v_m) \in \Sigma_*^m$, there exists a predicate $P_v^{SHVE} \in \mathcal{P}^{SHVE}$, for $\mathbf{x} = (x_1, \ldots, x_m) \in \Sigma^m$, have:

$$P_v^{SHVE}(\mathbf{x}) = \begin{cases} 1 & \forall\ 1 \le i \le m(v_i = x_i \text{ or } v_i = *), \\ 0 & \text{otherwise.} \end{cases}$$

The parameter m is called the width of SHVE.

SHVE.Setup(1^λ). It takes a security parameter λ as input, outputs $(msk, \mathcal{M})$, where $msk \xleftarrow{\$} \{0,1\}$ and payload message space $\mathcal{M} = \{\text{'True'}\}$.

SHVE.KeyGen(msk, $\mathbf{v} \in \Sigma_*^m$). It takes a predicate vector $\mathbf{v}= (v_1, \ldots, v_m)$ and the master secret key msk as inputs, $S = \{l_j \in [m] | v_{l_j} \neq *\}$ is the set of all locations in $\mathbf{v}$ that not contain wildcard characters *. Let these locations be $l_1 < l_2 < \cdots < l_{|S|}$, $K \xleftarrow{\$} \{0,1\}^{\lambda+\log\lambda}$ and then sets the following:

$$\begin{aligned} d_0 &= \oplus_{j\in[|S|]}\Big(F_0(msk, v_{l_j}||l_j)\Big) \oplus K, \\ d_1 &= \text{Sym.Enc}\Big(K, 0^{\lambda+\log\lambda}\Big). \end{aligned}$$

Finally, outputs the decryption key:

$$\mathbf{s} = (d_0, d_1, S).$$

SHVE.Enc($msk, \mu \in \mathcal{M}$, $\mathbf{x} \in \Sigma^m$). It takes a message μ, the master secret key msk and an vector $\mathbf{x} = (x_1, \ldots, x_m)$, outputs the ciphertext $\mathbf{c}$:

$$\mathbf{c} = (\{c_l\}_{l\in[m]}).$$

where $c_l = F_0(msk, x_l||l)$, for each $l \in [m]$.

SHVE.Query(s,c). It takes a ciphertext $\mathbf{c} = \Big(\{c_l\}_{l\in[m]}\Big)$ and a decryption key $\mathbf{s} = (d_0, d_1, S)$, and then computes the following:

$$K' = \Big(\oplus_{j\in[|S|]}\, c_{l_j}\Big) \oplus d_0.$$

$$\mu' = \text{Sym.Dec}(K', d_1).$$

If $\mu' = 0^{\lambda+\log\lambda}$, this algorithm outputs 'True', else outputs $\perp$.

3 The Conjunctive Keyword FSSE Scheme

In this section, we construct two FSSE Schemes that support conjunctive keyword query: the basic scheme FHXT-B and the enhanced scheme FHXT-E. The main idea is that we extend the HXT protocol to support forward secure conjunctive keyword search. Specifically, our basic scheme is just a simple combination, limited to supporting the update of the entire document (id, W(id)). That is, only after all keyword document pairs (w, id) in this document have been updated, the user is allowed to perform the search. In order to support more flexible updates, we apply the forward structure of the FASTIO protocol to the ciphertext database $\mathbf{c}$ of the HXT protocol to support the update of the keyword document pair (w, id), allowing users to perform search operations at any time.

3.1 Dynamic Update of HXT

HXT protocol was proposed by Lai *et al.* [15] to hide the keyword pair result pattern in conjunctive keyword search scheme. For simplicity, the HXT protocol puts the XSet structure in the OXT protocol into a BF, and then encrypts the vector corresponding to the BF using SHVE.Enc to obtain the ciphertext database $\mathbf{c}$. When searching for conjunctive keywords, a search token is generated, for each document id corresponding to the keyword w_1, and then the server runs SHVE.Query(token$_c$, $\mathbf{c}$) for each token. If the return value is the preset message 'True', it means that all other keywords searched are also present in the document corresponding to the token. If it returns the failure flag $\perp$, it means that all other keywords searched are not in the document corresponding to the token, but the server cannot understand which specific keyword is not matched.

If we want the HXT protocol to support dynamic updates, must enable the ciphertext database $\mathbf{c}$ to support updates. The XSet structure in the OXT protocol is a simple set, it can be added arbitrarily. However, in the HXT protocol, $\mathbf{c}$ is generated after encrypting the BF corresponding to the XSet structure. A simple idea is to re-encrypt each time the XSet structure is updated to generate a new ciphertext database $\mathbf{c}$. In this way, each update operation needs to generate a new $\mathbf{c}$ for the entire database on the server, which undoubtedly brings huge resources and computational overhead. Our idea is to use the characteristics of the Bloom filter and the encryption algorithm SHVE.Enc of the ciphertext database $\mathbf{c}$ to update.When we add new elements to the XSet structure, we add new elements to BF, $\{H_i\}_{1\le i\le k}$ is the k independent hash functions of BF. If we want to add element e to BF, we need to set the bit of position $\{H_i(e)\}_{1\le i\le k}$ in the corresponding vector of BF to 1. Therefore, according to the SHVE.Enc algorithm, when we add element e to the ciphertext database, we only need to calculate:

$$\begin{aligned} &e_i = F_0(msk, H_i(e)||1), \quad (1 \le i \le k) \\ &\mathbf{c}[H_i(e)] = e_i, \qquad (1 \le i \le k). \end{aligned} \tag{2}$$

In this way, the dynamic update of the ciphertext database $\mathbf{c}$ is realized, that is, the dynamic update of the HXT protocol is realized.

3.2 Basic Construction FHXT-B

In this section, we will give the specific structure of our basic scheme FHXT-B. Our main idea is to combine the FSSE scheme FASTIO that supports single-keyword

search with our modified HXT protocol that supports dynamic updates. We use the forward security structure $\mathbf{T}_e$ in the FASTIO scheme to perform a single keyword search, and then use the SHVE in the HXT protocol to implement membership checks to complete the connected keyword search. Next, we will describe the details of the update and search operations of FHXT-B. When the client wants to update the document id_c, the client will update each keyword document pair (w, id_c) in the id_c in turn. First, the client uses two hash functions to hash the state of the keyword w, the calculator, and the document index to make them unlinkable. Then calculate xtag, and use BF's k hash functions $\{H_i\}_{1\leq i\leq k}$ and SHVE.enc to encrypt and calculate the data that needs to be added to the ciphertext database $\mathbf{c}$, and then send it to the server. The server updates the databases $\mathbf{T}_e$ and $\mathbf{c}$. When the client wants to search for the conjunctive keyword $(w_1, \ldots, w_n)$, the server first uses a single keyword search to find the document id corresponding to the least frequent keyword w_1. Then according to the search token xtoken, the client calculates the $xtag$ corresponding to other keywords and the document id_c, and generates the vector $\mathbf{v}_c$, and finally sends $token_c$ according to $\mathbf{v}_c$ to the server. The server uses SHVE.Query for membership check, and judge whether the document contains all other keywords. If it does, return the id_c. The details of our basic scheme are given in Algorithm 1, Algorithm 2 and Algorithm 3.

Setup(1^λ). This algorithm is to generate parameters, keys, and data structures necessary for the scheme. λ is security parameter, F_1 and F_p are two pseudo-random functions, Σ, T_e and T_c are three empty maps. Σ is kept in the client, which stores the status st and counters c_1, c_n of each keyword w. T_e and T_c are kept in the server, which is used to store the encrypted index and the last search result of the keyword w separately. Besides, the SHVE.Setup(1^λ) algorithm is run to generate the SHVE master key msk, and initialize a well-designed bloom filter BF. Then SHVE.Enc is run to encrypt BF to obtain the ciphertext database $\mathbf{c}$, which can be regarded as an empty database at this time. Note that the master key msk is stored on the client side, while $\mathbf{c}$ is stored on the server side.

Update(add, ind, σ; EDB). To insert a new document ind, we need to update all keyword document pairs (w, ind) in ind, where $w \leftarrow \mathrm{W}(ind)$. Specifically, for each keyword w, the client reads its latest state st and counters c_1 and c_n from the map Σ. If not, it randomly generates a state (lines 7-11 in Algorithm 1 Update). The counter c_1 represents the number of updates of w after the last search for the keyword w, and the counter c_n represents the total number of documents containing the keyword w. Then use the hash function h_1 to generate the position u for the state st and the counter c_1, and use the hash function h_2 and XOR operation to encrypt the document index ind. The client saves the latest state of the keyword w into Σ, and then calculates the $xtag$ related to the keyword document pair (w, ind).Then, the client uses the k hash functions of the BF and the pseudo-random function F_0 in the SHVE.Enc algorithm to calculate $xtag$ and save it in the map ABF (Lines 17-22 in Algorithm 1 Update). Finally, the client sends the set AddSet and ABF of all $(u, (e, y))$ to the server. The server updates $\mathbf{T}_e$ and $\mathbf{c}$.

Search($(w_1, \cdots, w_n), \sigma$; EDB). During the search protocol, there are two rounds of interaction between client and server. The first round aims to perform search on the least frequency keyword. If the

Algorithm 1 FHXT-B

Setup(1^λ)

1: Select keys K_S for PRF F_1
2: Select keys K_X, K_I, K_Z for PRF F_p
3: Σ ,$\mathbf{T}_e$, $\mathbf{T}_c \leftarrow$ empty map
4: Run SHVE.Setup(1^λ) to get msk
5: Select hash functions $\{H_j\}_{1\leq j\leq k}$ for BF
6: Initialize BF $\leftarrow 0^m$
7: Compute $\mathbf{c} \leftarrow$ SHVE.Enc($msk, \mu =$"True", BF)

Update(add, ind, σ; EDB)

1: *Client* :
2: WSet $\leftarrow$ W(ind)
3: AddSet $\leftarrow \{\}$
4: ABF $\leftarrow$ empty map
5: **while** |WSet| $\neq \phi$ **do**
6: $w \xleftarrow{\$}$WSet; WSet $\leftarrow$ WSet $\setminus\{w\}$
7: $(st, c_1, c_n) \leftarrow \Sigma[w]$
8: **if** $(st, c_1, c_n) = \perp$ **then**
9: $st \xleftarrow{\$} \{0,1\}^\lambda$
10: $c \leftarrow 0$, $c_n \leftarrow 0$
11: **end if**
12: $u \leftarrow h_1(st||(c_1+1))$
13: $e \leftarrow (ind||(c_n+1)) \oplus h_2(st||(c_1+1))$
14: $\Sigma[w] \leftarrow (st, c_1+1, c_n+1)$
15: $z \leftarrow F_p(K_Z, w||c_n+1); y \leftarrow xind \cdot z^{-1}$
16: $xtag \leftarrow g^{F_p(K_X,w)\cdot xind}$
17: **for** $j = 1$ to k **do**
18: Let $l = H_j(xtag)$
19: Set $c_l = F_0(msk, 1||l)$
20: ABF$[l] = c_l$
21: **end for**
22: AddSet.$add((u, (e, y)))$
23: **end while**
24: Send (AddSet, ABF) to th server
25: *Server* :
26: **for** each $(u, (e, y)) \in AddSet$ **do**
27: $\mathbf{T}_e[u] \leftarrow (e, y)$
28: **end for**
29: **for** l, c_l **in** ABF **do**
30: $\mathbf{c}[l] = c_l$
31: **end for**

Algorithm 2 FHXT-B

Search$((w_1, \cdots, w_n), \sigma;$ EDB)

Client :
$(st, c_1, c_n) \leftarrow \Sigma[w_1]$
if $(st, c_1, c_n) = \perp$ **then**
  return ϕ
end if
$t_w \leftarrow F_1(K_S, w_1)$
if $c_1 \neq 0$ **then**
  $k_w \leftarrow st$, $st \xleftarrow{\$} \{0,1\}^\lambda$
  $\Sigma[w_1] \leftarrow (st, 0, c_n)$
else
  $k_w = \perp$
end if
for $1 \leq i \leq c_n$ **do**
  for $2 \leq j \leq n$ **do**
    $\text{xtoken}[i,j] \leftarrow g^{F_p(K_Z, w_1||i) \cdot F_p(K_X, w_j)}$
  end for
  Set $\text{xtoken}[i] = \text{xtoken}[i,2], \cdots, \text{xtoken}[i,n]$
end for
Send $(t_w, k_w, c_1, \text{xtoken}[1], \cdots, \text{xtoken}[c_n])$ to server
$R \leftarrow \{\}$
ID $\leftarrow \{\}$
ID.add($\mathbf{T}_c[t_w]$)
if $k_w \neq \perp$ **then**
  for $i = c_1$ to 1 **do**
    $u_i \leftarrow h_1(k_w||i)$
    $(e, y) \leftarrow \mathbf{T}_e[u_i]$
    $(ind, c_i) \leftarrow e \oplus h_2(k_w||i)$
    ID.add((ind, c_i, y))
    delete $\mathbf{T}_e[u_i]$
  end for
  $\mathbf{T}_c[t_w] \leftarrow$ **ID**
end if
for $(ind, c, y) \in$ **ID do**
  $\text{U}_c \leftarrow \{\}$
  for $l = 2$ to n **do**
    $xtag = \text{xtoken}[c, l]^y$
    for $j = 1$ to k **do**
      $\text{U}_c.add(H_j(xtag))$
    end for
  end for
end for
Send U_c to client

Algorithm 3 FHXT-B

Client :
Initialises $\mathbf{v}_c \leftarrow *^m$
for $u_j \in \text{U}_c$ **do**
  Sets $\mathbf{v}_c[u_j] \leftarrow 1$
end for
$\text{token}_c \leftarrow \text{SHVE.KeyGen}(msk, \mathbf{v}_c)$
Sends token_c to server
Server :
$\text{res}_c \leftarrow \text{SHVE.Query}(\text{token}_c, \mathbf{c})$
if $\text{res}_c =$'True' **then**
  Adds e_c to R
end if
Sends R to client

client wants to perform conjunctive keyword search $w_1 \wedge \cdots \wedge w_n$, s/he finds the keyword w_1 with the smallest c_n in the local storage, and then gets the state s_t, counters c_1 and c_n of w_1. If there is a document containing the keyword w_1, the client generates t_w, which is to enable the server to find the last search query result of w_1 in $\mathbf{T}_c$. The client next judges whether the counter c_1 is 0. If it is not 0, it means that new documents about the keyword w_1 have been added after the last search for the keyword w_1. The client will send st to the server, and a new random state st must be regenerated, and the counter c_1 is reset (Lines 6-10 in Algorithm 2). If c_1 is 0, it means that after the last search for keyword w_1, there is no new documents related to w_1 have been added. In this case, the server can directly use t_w to find all relevant documents in $\mathbf{T}_c$. The client then generates xtoken and sends it to the server. On the server side, the server first retrieves $\mathbf{T}_c$ using t_w to get the result of the keyword w_1. Subsequently, the server calculates the $xtag$ between all ind corresponding to w_1 and all other keywords. The second round is to find out the final search results. After receiving all the positions sent by the server, the client initializes a vector $\mathbf{v}_c$, and sets some positions to 1 according to U_c (Line 2-5 in Algorithm 3). Subsequently, the client uses $\text{SHVE.KeyGen}(msk, \mathbf{v}_c)$ to calculate the corresponding search token token_c and sends it to the server. The server executes $\text{SHVE.Query}(\text{token}_c, \mathbf{c})$ to determine whether the document ind_c corresponding to the token_c contains other queried keywords. If the returned result is "True", it will be added to the return set R. Finally, the result R is returned to the client.

Note that our basic scheme only supports updating a document, that is, users can not perform search operations until all keyword document pairs in a document are updated. The reason is similar to the FOXT-B scheme of Wang *et al.* [21]. We use the following example to illustrate the problem. First, the user outsources the documents id_1 and id_2 to server, where id_1 contains keywords (w_1, w_2) and id_2 contains w_2. Then, the user wants to add a new document id_3, which contains keywords (w_1, w_2). In the following, we will show how the forward privacy is broken if the user performs a search operation before the server has updated all the keyword document pairs. Specifically, after the server has updated the keyword document pair (w_1, id_3), client performs the conjunctive keyword search $w_1 \wedge w_2$, and send $(t_{w_1}, k_{w_1}, c_1, xtoken[1,2], xtoken[2,2])$ to the server. The server uses t_{w_1} and k_{w_1} to find the (e_1, y_1) and (e_3, y_3) corresponding to the keyword w_1, then server and client

interact to generate the token$_1$ and token$_3$ corresponding to xtoken[1, 2]^{y_1}, and xtoken[2, 2]^{y_3}. Server executes HVE.Query(token, **c**), the result of token$_1$ is "True" and token3 is returned as $\perp$, so the search result is e_1 , which is document id_1. After client has updated the keyword document pair (w_2, id_3), server curiously uses token$_3$ to execute SHVE.Query(token$_3$, **c**), and the result at this time is "True". Therefore, server knows that the newly inserted document id_3 matches the previous search request $w_1 \wedge w_2$, so that the forward privacy are destroyed.

3.3 Enhanced Construction FHXT-E

Our enhanced scheme FHXT-E can support more flexible keyword document pair (w, id) update. In other words, the user can perform search operation at any time. The reason why the FHXT-B cannot support flexible update is that the ciphertext database **c** is just a simple database, not a forward-secure one. To this end, we will use the state chain-based forward security structure $\mathbf{T}_e$ in the FASTIO scheme. Specifically, we construct a similar structure **C** to store data that be added to the ciphertext database **c** during the update. The added data will not be updated to **c** until the keyword corresponding to the data is searched. The details are as follows.

Setup(1^λ)**.** Two new mapping structures **C** and Γ have been added. **C** is used to temporarily store the data that needs to be added to the ciphertext database **c** during the update. Γ is similar to the structure Σ, which corresponds to the $\mathbf{T}_c$ structure on the server side. It is used to store the state and counter of the keyword w and corresponds to the **C**.

Update(add, w, id, F_0, σ**; EDB).** This algorithm updates the counters of the keyword w in Σ and Γ on the client side, and then updates $\mathbf{T}_e$ on the server side. The main difference is that ABF is added to **C**.

Search($(w_1, \cdots, w_n, \sigma$**; EDB).** The single-keyword search is consistent with the search protocol of the FHXT-B scheme. The main differences are as follows. For the FHXT-E scheme, the client needs to update the status and counter of each queried keyword searched except the least frequent keyword in Γ (Lines 14-22 in Algorithm 5), and then sends the relevant status and counter in Γ to the server as part of the search token. On the server side, the first round of interaction is consistent with the FHXT-B scheme. In the second round of interaction, the server needs to update the data related to the search keywords in **C** to the ciphertext database **c**, and then deletes the corresponding data in **C** (Lines 7-15 in Algorithm 6).

Algorithm 4 FHXT-E

Setup(1^λ)

1: Select keys K_S for PRF F_1
2: Select keys K_X, K_I, K_Z for PRF F_p
3: Σ, Γ, T_e, T_c, **C** $\leftarrow$ empty map
4: Run SHVE.Setup(1^λ) to get msk
5: Select hash functions $\{H_j\}_{1\le j\le k}$ for BF
6: Initialize BF $\leftarrow 0^m$
7: Compute **c** $\leftarrow$ SHVE.Enc$(msk, \mu =$"True", BF)

Update(add, w, id, F_0, σ; EDB)

1: *Client* :
2: ABF $\leftarrow \{\}$
3: $(st, c_1) \leftarrow \Sigma[w]$
4: $(sc, c_2, c_n) \leftarrow \Gamma[w]$
5: **if** $(st, c_1) = \perp$ **then**
6: $st \xleftarrow{\$} \{0,1\}^\lambda$, $c_1 \leftarrow 0$
7: $sc \xleftarrow{\$} \{0,1\}^\lambda$, $c_2 \leftarrow 0$, $c_n \leftarrow 0$
8: **end if**
9: $u \leftarrow h_1(st||(c_1+1))$
10: $e \leftarrow (ind||(c_n+1)) \oplus h_2(st||(c_1+1))$
11: $\Sigma[w] \leftarrow (st, c_1+1)$
12: $\Gamma[w] \leftarrow (sc, c_2+1, c_n+1)$
13: $xind \leftarrow F_p(K_I, id)$
14: $z \leftarrow F_p(K_Z, w||(c_n+1)); y \leftarrow xind \cdot z^{-1}$
15: $xtag \leftarrow g^{F_p(K_X,w)\cdot xind}$
16: $u_c \leftarrow h_1(sc||(c_2+1))$
17: **for** $j = 1$ to k **do**
18: $l = H_j(xtag) \oplus h_2(sc||(c_2+1))$
19: $ce_l = F_0(msk, 1||H_j(xtag))$
20: ABF.add$((l, ce_l))$
21: **end for**
22: AddSet $\leftarrow \{u, (e, y), u_c,$ ABF$\}$
23: Send AddSet to server
24: *Server* :
25: $\mathbf{T}_e[u] = (e, y)$
26: $\mathbf{C}[u_c] =$ ABF

4 Analysis of Our Proposed Scheme

4.1 Security Analysis

In this section, we give the security analysis of our enhanced scheme FHXT-E. During the setup, the client generates its own private key and an encrypted database c, so no information is leaked to the server at this stage. In the update protocol, the client sends the server fu, (e; y), uc, and ABFg. Since these data are random to the server every time, the server does not obtain any information. In the following, we present the leakage function $\mathcal{L}^{Srch}$ during search. We represent a sequence of T conjunctive queries $\mathbf{q} = (\mathbf{s}, \mathbf{x})$ where $\mathbf{s}_t$ denotes the sterm (least frequent keyword) and $\mathbf{x}_t$ denotes the xterms (other keywords) in the t-th query, $t \in [T]$. We give the leakage function $\mathcal{L}^{Srch}$ of FHXT-E as follow:

Algorithm 5 FHXT-E

Search$((w_1, \cdots, w_n, \sigma;$ EDB$)$

Client :
$(st, c) \leftarrow \Sigma[w_1]$
$(sc_1, c_{2_1}, c_n) \leftarrow \Gamma[w_1]$
if $(st, c) = \perp$ **then**
  return ϕ
end if
$t_w \leftarrow F_1(K_S, w_1)$
if $c \neq 0$ **then**
  $k_w \leftarrow st$, $st \xleftarrow{\$} \{0,1\}^\lambda$
  $\Sigma[w_1] \leftarrow (st, 0)$
else
  $k_w = \perp$
end if
for $2 \leq i \leq n$ **do**
  $(sc_i, c_{2_i}, c_{n_i}) \leftarrow \Gamma[w_i]$
  if $c_{2_i} \neq 0$ **then**
    $cw_i \leftarrow sc_i$, $sc_i \xleftarrow{\$} \{0,1\}^\lambda$
    $\Gamma[w_i] \leftarrow (sc_i, 0, c_{n_i})$
  else
    $cw_i = \perp$
  end if
end for
$cw = \{(cw_2, c_{2_2}), \cdots, (cw_n, c_{2_n})\}$
for $1 \leq i \leq c_{n_1}$ **do**
  for $2 \leq j \leq n$ **do**
    xtoken$[i, j] \leftarrow g^{F_p(K_Z, w_1||i) \cdot F_p(K_X, w_j)}$
  end for
  Set xtoken$[i]$ = xtoken$[i, 2], \cdots,$ xtoken$[i, n]$
end for
Send $(t_w, k_w, cw, c,$ xtoken$[1], \cdots,$ xtoken$[c_n])$ to server
Server :
$R \leftarrow \{\}$
ID $\leftarrow \{\}$
ID.add$(\mathbf{T}_c[t_w])$
if $k_w \neq \perp$ **then**
  for $i = c$ to 1 **do**
    $u_i \leftarrow h_1(k_w||i)$
    $(e, y) \leftarrow \mathbf{T}_e[u_i]$
    $(ind, c_i) \leftarrow e \oplus h_2(k_w||i)$
    ID.add$((ind, c_i, y))$
    delete $\mathbf{T}_e[u_i]$
  end for
  $\mathbf{T}_c[t_w] \leftarrow$ **ID**
end if
for $(ind, c, y) \in$ **ID do**
  for $l = 2$ to n **do**
    $xtag =$ xtoken$[c, l]^y$
    for $j = 1$ to k **do**
      $u_j \leftarrow H_j(xtag)$
    end for
  end for
end for
Sends u_j to client

Algorithm 6 FHXT-E

Client :
Initialises $\mathbf{v}_c \leftarrow *^m$
Sets $\mathbf{v}_c[u_j] \leftarrow 1$
token$_c \leftarrow$ SHVE.KeyGen(msk,$\mathbf{v}_c$)
Sends token$_c$ to server
Server :
for $2 \leq i \leq n$ **do**
  for $j = c_{2_i}$ to 1 **do**
    $u_{c_j} \leftarrow h_1(cw_i||j)$
    for $(l, ce) \in \mathbf{C}[u_{c_j}]$ **do**
      $\mathbf{c}[l \oplus h_2(cw_i||j)] \leftarrow ce$
      delete $\mathbf{C}[uc_j]$
    end for
  end for
end for
res$_c \leftarrow$ SHVE.Query(token$_c$, $\mathbf{c}$)
if res$_c$ ='True' **then**
  Adds ind_c to R
end if
Sends R to client

- SRP$[t]$ is the access pattern for the sterm of the t-th query, SRP$[t]$ = R(s_t).
- SP$[t]$ is the number of the search results for the sterm of the t-th query, SP$[t]$ = $|$R$(s_t)|$.
- XT$[t]$ is the number of xterm's of the t-th query, XT$[t]$ = $|\mathbf{x}_t, .|$.
- UXP$[t, i]$ is the number of updated items after last search for xterm $\mathbf{x}_i$ of the t-th.
- WRP[t] is the result pattern of t-th query, WRP$[t]$ = R$[s_t] \cap \bigcap_{i=1}^{XT[t]}R[\mathbf{x}_i[t]]$.
- IP$[t_1, t_2, \alpha, \beta]$ is the condition insertion pattern. If $s_{t_1} \neq s_{t_2}$ and $x_{t_1,\alpha} = x_{t_2,\beta}$, IP$[t_1, t_2, \alpha, \beta]$ = R$(s_{t_1}) \cap$ R(s_{t_2}). Otherwise, IP$[t_1, t_2, \alpha, \beta] = \phi$.

In the following, we give formal security analysis.

Theorem 1. *Our scheme FHXT-E is $\mathcal{L}$-semantically secure against adaptive attacks, assuming that the DDH assumption holds in $\mathbb{G}$, that HVE is a selectively simulation-secure scheme, that FASTIO is a $\mathcal{L}$-adaptively-secure dynamic SSE with forward privacy.*

Proof. The proof is played through a series of indistinguishable games. The first game is a real-world game, and the last is an ideal-world game simulated by an efficient simulator. Here, we mainly introduce how the simulator works. The details of the simulator $\mathcal{S}$ is shown in Algorithm 7. In the search protocol, similar to [17], we maintain a mapping **Token** to store (w, t_w) pairs.In the search protocol, we use $\underline{s_w} \leftarrow min \quad \mathbf{sp}(s_t)$ to denote the very first index that w appeared in search pattern and $\overline{s_w} \leftarrow max \quad \mathbf{sp}(s_t)$ to denote the last index that w been searched. Each w uniquely identifies an unknown keyword

Algorithm 7 Simulator $\mathcal{S}$

Search($\mathcal{L}^{Srch}$)

$\underline{s_w} \leftarrow min \quad \mathbf{sp}(s_t)$
$\overline{s_w} \leftarrow max \quad \mathbf{sp}(s_t)$
if $\mathbf{Token}[\underline{s_w}] = \perp$ **then**
  $\mathbf{Token}[\underline{s_w}] \xleftarrow{\$} \{0,1\}^\lambda$
end if
$t_w \leftarrow \mathbf{Token}[\underline{s_w}]$
if $\mathbf{uh}^{>\overline{w}}(s_t) = \perp$ **then**
  $k_w \leftarrow \perp$
else
  $k_w \xleftarrow{\$} \{0,1\}^\lambda$
end if
$c \leftarrow |\mathbf{uh}^{>\overline{w}}(s_t)|$
for $1 \le i \le \mathrm{XT[t]}$ **do**
  $cw_i \leftarrow \{0,1\}^\lambda$
end for
$cw = \{(cw_1, \mathrm{UXP[t,1]}), \ldots, (\mathrm{cw_{XT[t]}}, \mathrm{UXP[t, XT[t]]})$
for $w \in \hat{\mathbf{x}}$ and $id \in \bigcup_{t=1}(\mathrm{WRP}[t] \cup \bigcup_{j\neq t,\alpha,\beta} \mathrm{IP}[t,j,\alpha,\beta])$ **do**
  $A[id, w] \xleftarrow{\$} \mathbb{G}$
end for
for $1 \le c \le \mathrm{SP[t]}$ **do**
  $\mathbf{v}_c \leftarrow *^m$
  for $1 \le l \le \mathrm{XT[t]}$ **do**
    $(e_c, y_c) \leftarrow \mathrm{SRP[t]}$
    if $\overline{id}_{\sigma(c)} \neq \perp \wedge \overline{id}_{\sigma(c)} \in WRP[t]$ **then**
      $\mathrm{xtoken}[c,l] \leftarrow A[\overline{id}_{\sigma(c)}, \hat{\mathbf{x}}[c,l]]^{1/y_c}$
    else if $\overline{id}_{\sigma(c)} \neq \perp \wedge \overline{id}_{\sigma(c)} \in \bigcup_{j\neq t,v} \mathrm{IP}[t,j,l,v]$ **then**
      $\mathrm{xtoken}[c,l] \leftarrow A[\overline{id}_{\sigma(c)}, \hat{\mathbf{x}}[c,l]]^{1/y_c}$
    else
      $\mathrm{xtoken}[c,l] \xleftarrow{\$} \mathbb{G}$
    end if
    for $j = 1 : k$ **do**
      $\mathbf{v}_c[H_j(\mathrm{xtoken}[c,l]^{y_c})] = 1$
    end for
  end for
  $\mathrm{token}_c[t] \leftarrow \mathcal{S}_{HVE}(\alpha(\mathbf{v}_c), \beta(\mathbf{v}_c, \mathrm{BF}))$
  $\mathrm{res}_c[t] = \mathrm{HVE.Query}(\mathrm{token}_c[t], e_c)$
  if $\mathrm{res}_c[t] = \mathrm{True}$ **then**
    $\varepsilon[t] \leftarrow \varepsilon[t] \cup \{e_c\}$
  end if
  $Res \leftarrow \varepsilon[t]$;ResInds $\leftarrow \mathrm{R}(s_t) \cap \bigcap_{l=1}^{\mathrm{XT[t]}} \mathrm{R}(x_l[t])$
end for
return (Res, ResInds, t_w, k_w, c_w, c, xtoken[t], token[t])

w and the simulator can just use w without knowing w. The token t_w is associated with w. The simulator can use the update history to decide whether there have been new updates since the last search query. We use $\mathbf{uh}^{>k}(s_t)$ to denote the partial update history after the k-th query. For updating the $\mathbf{c}$, we send the UXP[t, i] in cw to server. Then, we need to compute xtoken, similar to [15], We first define a restricted equality pattern of $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, which is denoted as $\hat{\mathbf{x}}$ and is not exactly the equality pattern of x but the known equality pattern derived from the leakage IP, which is similar to that of [15]. In the simulation, array A is only filled out for positions $w \in \hat{\mathbf{x}}$ and $id \in \bigcup_{t=1}(\mathrm{WRP}[t] \cup \bigcup_{j\neq t,\alpha,\beta} \mathrm{IP}[t,j,\alpha,\beta])$, which is used to keep the reuse pattern of A during the generation of xtoken. Then, the simulator uses SP, XT, SRP, WRP, IP to simulate xtoken. Finally, the simulator uses $\mathcal{S}_{HVE}$ to get the Res.

Then we construct successive indistinguishable games. The first one is designed to have the same distribution as the $\mathbf{Real}^{\Pi}_{\mathcal{A}}(\mathcal{K})$, and the last one has the same distribution as the $\mathbf{Ideal}^{\Pi}_{\mathcal{A}}(\mathcal{K})$ which is generated by the simulator. Since the leakage of $\mathcal{S}_{HVE}$ and $\mathcal{S}_{FASTIO}$ have been given in [15] and [17], we omit the details of games here. □

4.2 Comparison

In this section, we compare our scheme FHXT-E with FOXT-E [21] in terms of computational costs (of the setup, update and search phases), storage size, and communication overheads for conjunctive query ($w_1 \wedge w_2 \wedge \cdots \wedge w_n$) with s-term = w_1. The details are summarized in Table 3, and its symbols are given in Table 2.

Table 2: Notations for Comparison Analysis

Notation	Meaning
p	number of pairings
G	size of an element from DH group
m	length of the BF
m'	number of non-wildcard elements in a BF
n'	number of updated datas after last search for w_1
α	number of identifiters matching w_1
$\lvert\xi\rvert$	number of identifiters matching a query
T_{hash}	time taken to compute a hash
T_{PRF}	time taken to compute a PRF
T_{Enc}	time taken to compute a sym.ciphertext
T_{Dec}	time taken to decrypt a sym.ciphertext
T_{xtag}	time taken to compute a xtag
T_{XOR}	time taken to perform an exclusive-or operation over λ
T_{exp}	time taken to compute an exponentiation

Setup Computational Costs. In the setup phase, the main time consumption of the FHXT-E scheme is the calculation of the ciphertext database $\mathbf{c}$, and the time cost is $(m)T_{PRF}$. As shown in Figure 1 and Figure 2, the time costs and the size of $\mathbf{c}$ increase with the number of keyword-document pairs. However, the consumption is only in the setup phase which is acceptable.

Table 3: Computational costs, storage sizes, and communication overhead between the client and the server for FOXT-E and FHXT-E.

Conjunctive query q $= (w_1 \wedge w_2 \wedge \cdots \wedge w_n)$.			
		FOXT-E cost	FHXT-E cost
Comp.	**setup**	-	$(m)T_{PRF}$
	update	$4T_{hash} + T_{PRF} + T_{Enc} + T_{xtag}$	$(2k+3)T_{hash} + (k+1)T_{XOR} + T_{xtag} + kT_{PRF}$
	search(client)	$(n-1)(\alpha T_{exp} + T_{PRF})$	$(n-1)\alpha T_{exp} + \alpha((m')T_{PRF} + (m')T_{XOR} + T_{Enc})$
	search(server)	$(n-1)\alpha(3T_{hash} + T_{exp} + T_{XOR}) + \alpha(T_{hash} + T_{exp} + T_{PRF})$	$(n')(2T_{hash} + T_{XOR}) + (n-1)\alpha(T_{exp} + (k+2)T_{hash} + T_{XOR}) + \alpha((m') + T_{XOR} + T_{Dec})$
Stor.	**storage size (server)**	$3p\lambda$	$(m)\lambda + p\lambda$
Comm.	**rounds**	1	2
	bandwidth	$(n-1)\alpha G + (n+1)\lambda + \lvert\xi\rvert O(\lambda)$	$(n-1)\alpha G + (n+1)\lambda + \alpha(O(m') + 2\lambda) + \lvert\xi\rvert O(\lambda)$

Comp.:Computation; Stor.:Storage; Comm.:Communication

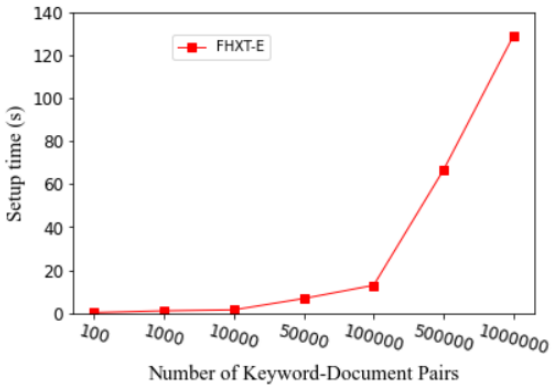


Figure 1: The setup efficiency

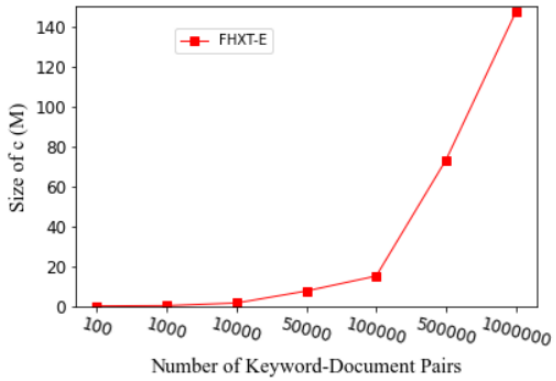


Figure 2: The size of **c**

Update Computation Costs. During the update phase, the common computational costs between FHXT-E and FOXT-E include the costs of generating xtag. Our scheme needs to hide the *xtag* using the k hash functions of the BF and the PRF F_0 in the SHVE.Enc algorithm, which costs $kT_{PRF} + (2k+1)T_{hash} + kT_{OXR}$. FHXT-E also uses hash and XOR operations to encrypt the *ind* and produce the state chain, so the time consumption is $2T_{hash} + T_{OXR}$. Therefore, the overall calculation cost of the client in FHXT-E is $(2k+3)T_{hash} + (k+1)T_{XOR} + T_{xtag} + kT_{PRF}$. In FOXT-E, due to the introduction of additional map and document encryption, it also has some hashing and encryption operations, which costs $4T_{hash} + T_{PRF} + T_{Enc} + T_{xtag}$.

Search Computation Costs. During the search for $w_1 \wedge w_2 \wedge \cdots \wedge w_n$, both FHXT-E and FOXT-E need to calculate *xtoken* in client, which costs $(n-1)\alpha T_{exp}$. In order to complete the membership check, FOXT-E change the structure of XSet from a set to a map and design a shell for each element in XSet, it calculate *wr* to find a location of XSet and search the *xtag*, which costs $(n-1)T_{PRF}$. Our scheme uses the SHVE.KeyGen to generate token$_c$, which costs $\alpha((m')T_{PRF} + (m')T_{XOR} + T_{Enc})$. In the server side, FOXT-E uses the *wr* to obtains all the $xtag'$ and saves them to a set XSubSet, then uses *xtoken* to complete the membership check, which costs $(n-1)\alpha(3T_{hash} + T_{exp} + T_{XOR}) + \alpha(T_{hash} + T_{exp} + T_{PRF})$. Our scheme calculate all the *xtag*, then uses SHVE.Query to complete the membership check, which costs $(n')(2T_{hash} + T_{XOR}) + (n-1)\alpha(T_{exp} + (k+2)T_{hash} + T_{XOR})\alpha((m') + T_{XOR} + T_{Dec})$

Storage Size. The EDB of FOXT-E consists of TSet, XSet and CR, which are all proportional to the num-

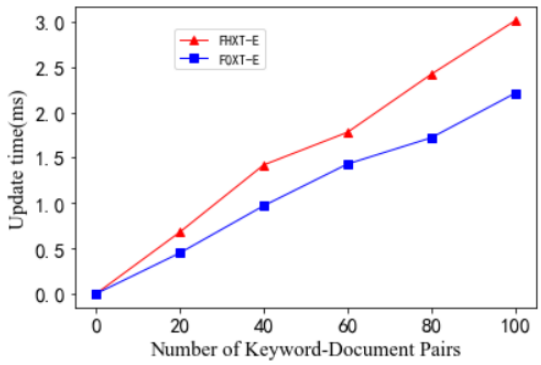


Figure 3: The update efficiency

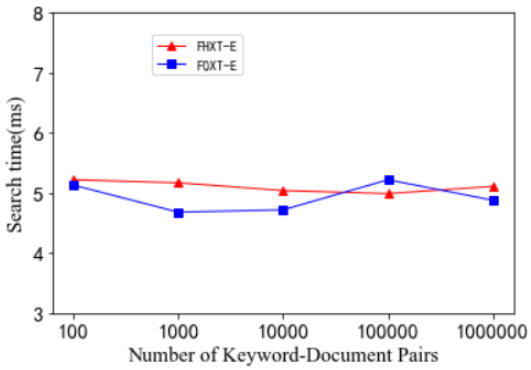


Figure 4: The search efficiency

ber of keyword-document pairs $3p\lambda$. The server of our scheme mainly stores the structures $\mathbf{T}_c$, $\mathbf{T}_e$, $\mathbf{C}$ and $\mathbf{c}$ generated by the BF encryption, where the size of $\mathbf{T}_c + \mathbf{T}_e$ is $p\lambda$, and the size of $\mathbf{c}$ is $(m)\lambda$.

5 Evaluations

We implemented the FHXT-E solution using Python3. For the cryptographic primitives and algorithms in this scheme, we use the hashlib library that comes with Python and the third-party encryption library *cryptography* to implement. The security parameter λ is set to 128 bits. We assess our scheme on three datasets from Wikimedia Download. The statistical features of the three datasets are given in Table 4.

Table 4: Statistics of the datasets used in the evaluation

Documents	Distinct keywords	Distinct(w, id)pairs
$3.2 * 10^2$	$1.7 * 10^3$	$7.4 * 10^4$
$1.8 * 10^3$	$2.6 * 10^4$	$5.3 * 10^5$
$2.7 * 10^4$	$1.4 * 10^5$	$8.3 * 10^6$

First, we focus on the size of search index EDB which is stored in the server side and used to perform search. As shown in Table 5, we give the size of the server-side EDB in the experiments of these three datasets. In HXT protocol, in order to satisfy the conjunctive keyword search, the FHXT-E scheme stores two ciphertext databases.

Table 5: Storage size of EDB

Distinct(w, id)pairs	$\mathbf{T}_c + \mathbf{T}_e$	$\mathbf{c} + \mathbf{C}$
$7.4 * 10^4$	13.3MB	23.55MB
$5.3 * 10^5$	80.89MB	172.03MB
$8.3 * 10^6$	1259.52MB	2426.88MB

Subsequently, we tested the update and search performance of the scheme FHXT-E and the scheme FOXT-E. First of all, we focus on the update efficiency. As shown in Figure 3, since our scheme needs to use SHVE.Enc to complete the dynamic update extension of the HXT protocol when updating, we are slightly inferior in terms of update efficiency. However, the FOXT-E scheme is based on the conjunctive keywords search implemented by the OXT protocol, and HXT protocol used in the our scheme has less information leakage to the server.

Finally, we focus on the comparison of the search efficiency of the two schemes. As shown in Figure 4, we perform conjunctive search of any two keywords on databases of different sizes to evaluate the search efficiency. It can be seen that the search time is not linearly related to the size of the database. As described above, our FHXT-E solution requires two rounds of interaction during search, while the FOXT-E solution only requires one round. Note that our scheme also has sub-linear search efficiency.

6 Conclusions

In this paper, we propose two effective forward-secure SSE schemes that support conjunctive keyword search, named FHXT-B and FHXT-E. Our schemes can hide the result pattern leakage when performing conjunctive keyword search. Specifically, FHXT-B is a simple combination of the forward security search solution FASTIO and the conjunctive keyword search protocol HXT. This simple combination leads the scheme cannot support flexible keyword-document pair updates. In other words, only all keyword-document pairs in the document are updated, the user can perform the search operation. We then construct an enhanced scheme to achieve more flexible keyword-document update, such that users can perform search at any time. The security analysis and experimental results show that our schemes gain better security at the cost of slightly increased communication and computing overheads.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62102313), Preferential Funding for Scientific and Technological Activities of Overseas Students in Shaanxi Province (No. 2019-25), Fundamental Research Funds for the Central Universities (No. JB211503), and Scientific Research Program Funded by Shaanxi Provincial Education Department (No. 21JK0903).

References

[1] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proceeding of Theory of Cryptography Conference (TCC'07)*, pp. 535–554, 2007.

[2] R. Bost, "$\sum o\varphi o\varsigma$: Forward secure searchable encryption," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*, pp. 1143–1154, 2016.

[3] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakage-abuse attacks against searchable encryption," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15)*, pp. 668–679, 2015.

[4] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Roşu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," in *Proceeding of Annual Cryptology Conference (CRYPTO'13)*, pp. 353–373, 2013.

[5] M.-Y. Chen, C.-W. Liu, and M.-S. Hwang, "Securedropbox: a file encryption system suitable for cloud storage services," in *Proceedings of ACM Cloud and Autonomic Computing Conference, (CAC'13)*, pp. 21:1–21:2. ACM, 2013.

[6] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.

[7] S. Faber, S. Jarecki, H. Krawczyk, Q. Nguyen, M. Rosu, and M. Steiner, "Rich queries on encrypted data: Beyond exact matches," in *Proceeding of European Symposium on Research in Computer Security (ESORICS'15)*, pp. 123–145, 2015.

[8] E.-J. Goh, "Secure indexes.," *IACR Cryptol. ePrint Arch.*, vol. 2003, p. 216, 2003.

[9] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proceeding of International Conference on Applied Cryptography and Network Security (ACNS'04)*, pp. 31–45, 2004.

[10] C. Hu, X. Song, P. Liu, Y. Xin, Y. Xu, Y. Duan, and R. Hao, "Forward secure conjunctive-keyword searchable encryption," *IEEE Access*, vol. 7, pp. 35035–35048, 2019.

[11] M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: ramification, attack and mitigation.," in *Proceeding of 19th Annual Network and Distributed System Security Symposium (NDSS'12)*, vol. 20, p. 12, 2012.

[12] S. Kamara and T. Moataz, "Boolean searchable symmetric encryption with worst-case sub-linear complexity," in *Proceeding of Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'17)*, pp. 94–124, 2017.

[13] S. Kamara, T. Moataz, and O. Ohrimenko, "Structured encryption and leakage suppression," in *Proceeding of 38th Annual International Cryptology Conference (CRYPTO'18)*, pp. 339–370, 2018.

[14] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS'12)*, pp. 965–976, 2012.

[15] S. Lai, S. Patranabis, A. Sakzad, J. K. Liu, D. Mukhopadhyay, R. Steinfeld, S.-F. Sun, D. Liu, and C. Zuo, "Result pattern hiding searchable encryption for conjunctive queries," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS'18)*, pp. 745–762, 2018.

[16] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding of 2000 IEEE Symposium on Security and Privacy (S&P'00)*, pp. 44–55, 2000.

[17] X. Song, C. Dong, D. Yuan, Q. Xu, and M. Zhao, "Forward private searchable symmetric encryption with optimized i/o efficiency," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 912–927, 2018.

[18] E. Stefanov, C. Papamanthou, and E. Shi, "Practical dynamic searchable encryption with small leakage.," in *Proceeding of 21st Annual Network and Distributed System Security Symposium (NDSS'14)*, vol. 71, pp. 72–75, 2014.

[19] J. Wang, X. Chen, S.-F. Sun, J. K. Liu, M. H. Au, and Z.-H. Zhan, "Towards efficient verifiable conjunctive keyword search for large encrypted database," in *Proceedings of the 23rd European Symposium on Research in Computer Security, (ESORICS'18)*, vol. 11099, pp. 83–100, 2018.

[20] Y. Wang, S.-F. Sun, J. Wang, J. K. Liu, and X. Chen, "Achieving searchable encryption scheme with search pattern hidden," *IEEE Transactions on Services Computing*, 2020.

[21] Y. Wang, J. Wang, S.-F. Sun, M. Miao, and X. Chen, "Toward forward secure sse supporting conjunctive keyword search," *IEEE Access*, vol. 7, pp. 142762–142772, 2019.

[22] Z. Wu and K. Li, "Vbtree: forward secure conjunctive queries over encrypted data for cloud computing," *The VLDB Journal*, vol. 28, no. 1, pp. 25–46, 2019.

[23] Y. Zhang, J. Katz, and C. Papamanthou, "All your queries are belong to us: The power of file-injection attacks on searchable encryption," in *Proceeding of 25th USENIX Security Symposium (USENIX Security'16)*, pp. 707–720, 2016.

[24] Z. Zhang, J. Wang, Y. Wang, Y. Su, and X. Chen, "Towards efficient verifiable forward secure searchable symmetric encryption," in *Proceedings of the 24th European Symposium on Research in Computer Security, (ESORICS'19)*, vol. 11736, pp. 304–321, 2019.

Biography

Yunling Wang received her Ph.D degree in Cryptography from Xidian University in 2015. During her Ph.D study, she worked as a visiting scholar at Monash University, Australia, for one year. Currently, she works at Xi'an University of Posts & Telecommunications. Her research interests include searchable encryption and cloud computing security.

Yichao Zhu received the B.S. degree from Xidian University, in 2017. He is currently pursuing the M.S. degree in applied mathematics with Xidian University, China. His research interests include searchable encryption and cloud security.

Jianfeng Wang is currently an associate professor in School of Cyber Engineering, Xidian University, China. He received his PhD degree on cryptography from Xidian University in 2016. His research interests include applied cryptography, searchable encryption, and database outsourcing. He has published more than 50 research papers in refereed international journals and conferences, such as IEEE Transactions on Computers, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Services Computing, ESORICS, etc. He has served as program committee member /reviewer in over 20 international conferences. He received Chinese Association for Cryptologic Research(CACR) Excellent Doctoral Dissertation Award.

Intrusion Detection Based on Feature Selection and Temporal Convolutional Network in Mobile Edge Computing Environment

Xubin Jiao, Jinguo Li, and Mi Wen

(Corresponding author: Jinguo Li)

Shanghai University of Electric Power, Shanghai 200090, China

Email: j18521732461@163.com

(Received May 9, 2021; Revised and Accepted Jan. 28, 2022; First Online Feb. 26, 2022)

Abstract

As an inevitable trend of future 5G networks, Mobile Edge Computing (MEC) has many advantages in providing content awareness and cross-layer optimization services. But it also faces malicious traffic attacks from emerging services and technologies. Intrusion Detection Systems (IDS) is an effective defense mechanism to monitor and detect abnormal data on the host side or network side. However, as the focus of research in network security, IDS can usually be deployed separately without collaboration. The previous work on IDS is mainly based on Recurrent Neural Network (RNN) design. However, there are characteristics of limited edge node resources in the MEC scenario, making it challenging to deploy IDS for MEC. Secondly, these RNN-based methods are not effective in terms of detection accuracy. We proposed the model with Extreme Gradient Boosting Decision Tree and Temporal Convolutional Network (XGBoost-TCN) to solve these problems. In short, we used the XGBoost algorithm to reduce high-dimensional traffic to low-dimensional traffic. After that, we used a TCN model to detect abnormal traffic. The effectiveness and adaptability of the model have been verified on the public dataset. In addition, the results of performance evaluation show that the model has higher detection accuracy than previous related work for highly imbalanced abnormal traffic datasets.

Keywords: Extreme Gradient Boosting Decision Tree; Intrusion Detection; Temporal Convolutional Network; XGBoost

1 Introduction

With the rapid development of smart devices, we are ushering in an era of the Internet of Things. IoT applications require mobile support, geographic distribution, location awareness and low latency. MEC was proposed to overcome the above-mentioned challenging problems [19]. The infrastructure of MEC is composed of three layers: cloud server, edge device layer and user equipment layer. The edge device layer is a unique computing node of MEC, which is closer to users. It can perform computing tasks and provide low-latency services.

However, the design of decentralizing the computing center to the edge in MEC will inevitably lead to user privacy and security issues. Attackers can attack different infrastructures in the MEC network architecture, such as user equipments, server nodes, network resources or virtual machines, and even the entire edge data center. This situation is similar to the attack scenario in the Internet, where the attacker can be "inside" or "outside" attacks on the network infrastructure.

IDS is an important tool for detecting computer network attacks. Early research on intrusion detection technology is mostly based on shallow machine learning methods. Dong *et al.* [12] proposed a WSN intrusion detection model based on information gain ratio and bagging algorithm. This model had the high detection accuracy for Blackhole, Grayhole, Flooding, Scheduling and Normal, but it did not experiment with other types of attacks in the WSN dataset. The deep learning algorithms that have emerged in recent years have promoted the development of intrusion detection technology. Wang *et al.* [21] extracts features through hidden layers to make the training data of the later levels more representative, which can effectively solve the detection problem of complex high-dimensional data, the rate of accuracy is 95.25%, but the ability to express features is general. In comparison, Rui-Hong Dong *et al.* [18] proposed the IG-PCA-RF to deeply express the feature information in the mining data. The classification accuracy of the method in this paper is 99.84% in NSL-KDD dataset, and the classification accuracy rate of CICIDS2017 dataset is 99.80%. However, the computing resources of mobile edge devices are very limited. Although the proposed method can achieve relatively high detection accuracy, it does take up a large amount of computing resources, which cannot meet the needs of deploying IDS in mobile edge devices of MEC.

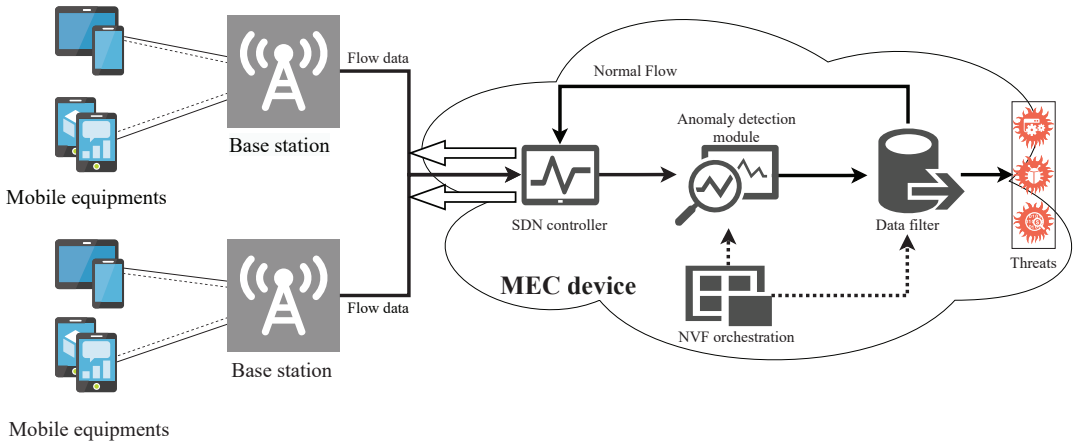


Figure 1: The MEC device handles abnormal traffic in the mobile equipments: first collect the traffic data of the mobile equipments through the base station, and then use the SDN controller in the MEC device to capture all the requested information of traffic statistics table to extract the traffic characteristics. Then the abnormal traffic detection module virtualized by NVF orchestration is used to detect the traffic; the virtual data filter is used for filtering, and then the filtered normal traffic SDN controller is returned to the mobile equipments.

To solve the above problems, we propose an intrusion detection model, XGBoost-TCN, which uses the XGBoost algorithm to perform feature selection on traffic to reduce the dimensionality of the traffic. Secondly, the selected representative features are sent to the TCN network, and the causal convolution and expansion convolution of the TCN network are used to refine the deep timing information in the features. Finally, we use the softmax classifier in the TCN network to classify the attack samples. The experimental results show that the model reduces the storage and improves the detection speed while ensuring a high detection rate. In conclusion, the proposed model is suitable for deployment in mobile edge devices. The main contributions of this paper are summarized as follows:

1) For the purpose of solving the problem of limited edge node resources in MEC, network traffic characteristics are screened according to the feature importance scoring mechanism of the XGBoost algorithm. While screening important features, the feature dimensions and classification model parameters are reduced, thereby reducing intrusion and the resource occupancy rate of the detection system.

2) In order to increase the flexibility of the system, this paper designs an intrusion detection architecture that combines Software Defined Network (SDN) and Network Functions Virtualization (NVF). The architecture diagram is shown in Figure 1. This architecture effectively deploys the IDS and adapts to the current 5G's network slicing requirements and developable.

3) We apply the method to the public dataset for the sake of proving the effectiveness of the proposed XGBoost-TCN model. The results of experiment show that this model can obtain higher detection accuracy than previous related work.

The rest of the paper is organized as follows. The second section reviews the works most relevant to intrusion detection. The third section describes the details of XGBoost-TCN model. In Section 4, the model is evaluated experimentally. Finally, the conclusion of the paper is given in Section 5.

2 Related Works

This section discusses the work of IDS based on machine learning methods. Generally speaking, it is mainly divided into two types, one is the traditional machine learning model; the other is the deep learning model. Both extract features from attack samples to detect abnormal traffic.

2.1 Traditional Machine Learning

Previous studies are mostly based on traditional machine learning methods, such as Support Vector Machine, Decision Tree and Naive Bayes. Support Vector Machine algorithm is an important algorithm in the field of machine learning and data mining. It is widely used in the tasks of classification, such as text classification and medical diagnosis. Cui *et al.* [9] combined PCA and OCSVM algorithm to build a multi-level intrusion detection model, using attack feature analysis method to preprocess data, while data cleaning and data feature selection of training set. The proposed method was difficult to solve the problem of high dimensionality of traffic. Radoglou *et al.* [17] combined three different classifiers of REP Tree, JRip and Forest PA. The highest accuracy rate on the CICIDS2017 dataset was 96.665% after using Min-Max Normalization

processing, and the lowest false alarm rate (FAR) was 1.145%. The accuracy rate was higher than that of Naive Bayes (74.528%) and other algorithms, which can effectively identify normal traffic and abnormal traffic. Due to the continuous changes of complex network traffic, the adaptive performance of this method is poor, and it cannot detect abnormal traffic of unknown networks effectively.

2.2 Deep Learning

In recent years, as a branch of machine learning, deep learning has become more and more popular. Research shows that deep learning completely surpasses traditional methods in performance with applying it to intrusion detection.

Zhang *et al.* [23] were dedicated to deep learning classifiers for SDN. They used WEKA and MATLAB to experiment on many dataset. A good classification accuracy has been achieved. Long Short-Term Memory (LSTM) is a special deep learning model of Recurrent Neural Network (RNN). It can remember the input and predict output of any period. Besides, it also solves the problem of gradient disappearance and explosion in RNN. Althubiti *et al.* [1] proposed a scalable hybrid IDS based on machine learning and Convolutional-LSTM, which can detect global and local potential threat attacks. Convolutional Neural Networks (CNN) is a machine learning algorithm with a multilayer network structure. It can learn hierarchical features from a large amount of data and has broad application prospects in the field of abnormal traffic detection. Naseer *et al.* [16] used Word Vectors and Text Convolutional Neural Networks to extract effective information from the payload, and then implemented the Random Forest algorithm on the combination of statistical features and payload features for classification. This document is experimented on the ISCX2012 dataset, and the accuracy of the proposed model is 99.13%. The FAR is 1.18%, which is better than traditional methods such as SVM (86.16%) and Random Forest algorithm (97.21%). It can effectively detect Distributed Denial of Service attacks and Penetration Testing attacks.

Based on the above work, traditional machine learning methods used for abnormal traffic detection often fail to detect many unknown and new security threats. This is because the constant changes in attack technology and the amount of data in the context of the big data era [20]. Traditional machine learning technologies such as Bayesian Network and Decision Tree have huge advantages in detection speed, but they are far less accurate than deep learning algorithms. In deep learning algorithms, the representative Deep Belief Network (DBN) adopts pre-training and fine-tuning methods solving the detection problem of complex high-dimensional traffic. In the public dataset NSL-KDD, the accuracy of DBN can get 95.25%, which is better than the SVM algorithm (91.36%) and the BP algorithm (89.07%) in machine learning. However, the detection speed is not as good as traditional machine learning algorithms. Many subsequent studies are based on the improvement of traditional machine learning and traditional deep learning. One of the goals is to make breakthroughs in speed and accuracy. This is also the focus of research on IDS in MEC.

3 Details of Proposed XGBoost-TCN Model

In this section, we describe the XGBoost-TCN model in the system architecture diagram in detail, and mainly introduce the core technologies of the model. The diagram of XGBoost-TCN model is shown in Figure 2. It is divided into two parts in total. The first part uses min-max standardization technology to preprocess the data. In order to overcome the problem of large dimension, it is based on the feature importance scoring mechanism of XGBoost algorithm to eliminate unnecessary features. The second part is building and training the TCN classifier, and then uses the TCN classifier to detect network intrusion. The following subsections will detail each step in the proposed method.

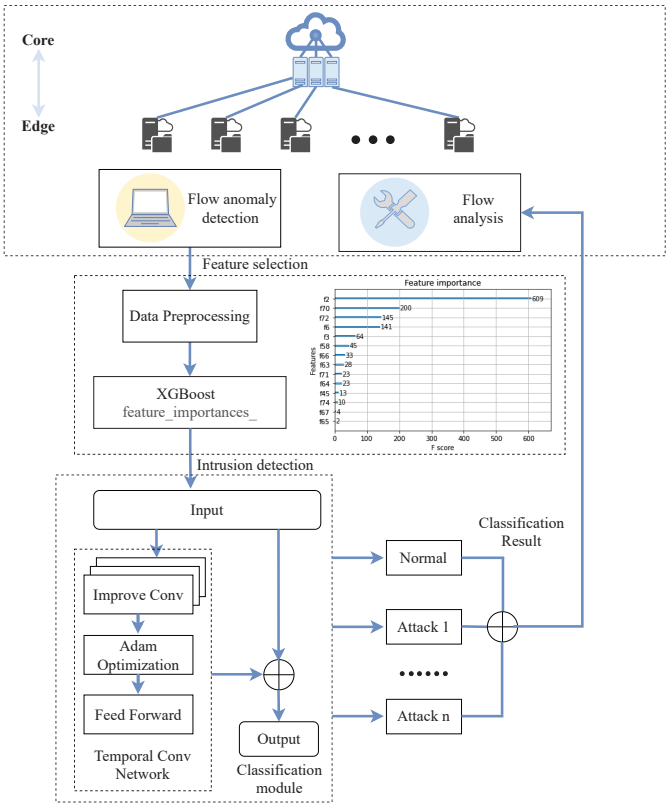


Figure 2: The XGBoost-TCN model diagram

3.1 Data Preprocessing

Data preprocessing is a process of normalizing the original data. Because many machine learning algorithms only

accept special forms during the training process, the input data needs to be normalized in advance. The Aegean WiFi Intrusion Dataset (AWID) dataset [14] used in this paper has 155 attributes, including 154 inputs and 1 class label, which indicates whether the trace shows legal behavior or intrusive behavior. Because many attributes of the AWID dataset are missing, it is necessary to clean the original dataset and fill some missing values. The AWID dataset has the characteristics of discrete and continuous values. When the discrete and continuous values of the features are combined, the range of the feature value will be different. In order to solve this problem, we apply the min-max standardization [6] to fit our calculations, which makes the entire feature in the same range, and provides greater flexibility for designing neural networks. The advantage of this normalization technique is that it accurately retains all the relationships in the data, so it does not cause any biases. The formula of the min-max normalization is as follows:

$$D' = \frac{D - D_{min}}{D_{max} - D_{min}} \tag{1}$$

where D' is the value of D after data normalization.

3.2 XGBoost for Feature Selection

Feature selection has the advantages of reducing computational complexity, eliminating redundant information, improving data generalization and understanding capabilities. The performance of the XGBoost model has been widely recognized in some data mining and machine learning challenges. Therefore, we adopted XGBoost technology to apply feature importance scoring mechanism to feature selection. XGBoost is composed of multiple Classification and Regression Trees (CARTs) [7]. Algorithm 1 introduces the process of building CART in detail. The basic decision tree of CART can be built on the concept of entropy. The object of CART is the Gini coefficient:

Algorithm 1 Greed algorithm for split finding

Input: I,instance set of current node
d,feature dimension
Output: Split with max score
1: $gain \Longleftarrow 0$
2: $G \Longleftarrow \sum_{i \in I} g_i$
3: **for** $k = 1$ to m **do**
4: $G_L \Longleftarrow 0$
5: Update $V_{o^+}^p$ based on Equation
6: **end for**

$$\begin{aligned} &\text{Obj} \ : \min \text{Gini}_{\text{index}}\left(D, a\right) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} \text{Gini}\left(D^v\right), \\ &\text{Gini}(D) = \sum_{k=1}^{K} \sum_{k' \neq k} p_k p'_k = 1 - \sum_{k=1}^{K} p_k^2 \end{aligned} \tag{2}$$

where a is an attribute of our choice, V is a value which is the scale of a, K is the scale of p_k. Intuitively, the Gini coefficient reflects the probability that the labels in the two samples in the dataset are different. In addition, this is the principle of constructing a tree. The goal of XGBoost is to fit the residuals. The residual is the difference between the actual value and the predicted value. XGBoost is defined as an additive model:

$$F(X, w) = \sum_{k=0}^{K} \alpha_k h_k\left(X, w_k\right) = \sum_{k=0}^{K} f_k\left(X, w_k\right) \tag{3}$$

where X is the input data, $F(X, w)$ is the model we finally get, h_k refers to a tree, w is the parameter of the tree, and α_k is the weight of the tree. By minimizing the loss function, we can get the optimal model $F(X, w)$. The loss function is defined as:

$$\begin{aligned} F^* &= \arg\min_{F} \sum_{k=0}^{K} L\left(Y_i, F\left(X_i, w_k\right)\right) \\ \text{loss} &= \sum_i l\left(\hat{Y}_i, Y_i\right) + \sum_k \Omega\left(f_k\right) \\ \Omega\left(f_k\right) &= \gamma N_{\text{leaf}} + \frac{1}{2}\lambda \left\|w_k\right\|^2 \\ l\left(\hat{Y}_i, Y_i\right) &= \left(Y \wedge_i - Y_i\right)^2 \end{aligned}$$

Among them, N_{leaf} represents the number of leaf nodes in the decision tree. This is a publicly released metric used to suppress the complexity of our model. Y_i is the real value and $\hat{Y}_i$ is the predicted value. γ and λ are parameters for calculation.

3.3 Classifier Based on TCN

Time series models generally use the traditional RNN model [2], RNN can effectively mine the timing information and semantic information in the data, and it has many improvements based on the traditional RNN model. For example, the common LSTM and GRU (Gate Recurrent Unit) can be very good to solve the problem of gradient disappearance and gradient explosion in the long-term memory and back propagation of the RNN model. However, with the development of CNN [8], there have been some studies trying to use CNN to solve sequence problems. The one-dimensional convolution operator is developed on the basis of the original two-dimensional image recognition operator, and some special variant structures are introduced from various CNN structures. As a unique member of CNN, TCN has beaten RNN in many major application fields [4].

The TCN model proposed in this paper is based on a time-domain convolutional neural network, and its core components are causal convolution and expansion convolution. The network structure of the model is shown in Figure 3. It contains following three parts.

3.3.1 Phase I: Causal Convolution

Causal convolution [13] means that the output at time t is only related to the input data before it. The input time of the convolution operator is earlier than time t. One of the

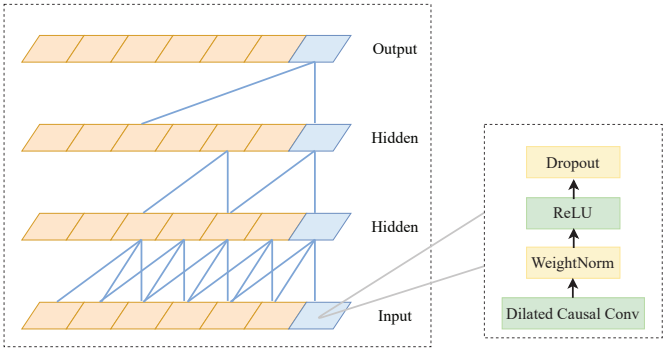


Figure 3: TCN model diagram

main advantages of using causal convolution is that the output will not be affected by future sequence values, and some bidirectional structures do not have this significance (such as bidirectional RNN). In our intrusion detection practice, data is obtained offline and causal convolution is not so important. However, it is useful when it comes to online scenarios.

3.3.2 Phase II: Dilated Convolution

Dilated convolution [15] increases the convolution field of view by changing the value interval of the convolution kernel. For example, for a one-dimensional input sequence $x \in \mathbb{R}$, the convolution kernel is $f : \{0, 1, 2, ..., k-1\} \rightarrow \mathbb{R}$, the dilated convolution operation F on the elements in the sequence is defined as follows:

$$F(s) = (\mathbf{x} *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d \cdot i} \tag{4}$$

where d is the expansion factor, k is the size of the convolution kernel, and $s - d \cdot i$ is the past direction. Where $d = 1$ means that dilated convolution is equivalent to regular convolution. The size of the receptive field of the dilated convolution can be calculated as $(k-1)d$. Obviously, we can increase the receptive field and choose a larger convolution kernel size k or increase the expansion factor d. In order to build a more structured and concise model, we add a deep network of d index ($d = o(2^i)$, where i represents the layer of network). The combination of causal convolution and dilated convolution enables the TCN model to extract features from network traffic.

3.3.3 Phase III: Residual Connection

Residual connection [10] performs the transformation $\mathcal{F}$, and its output is added to the input x of the block, enabling the layer to learn the modification of the identity mapping instead of the entire transformation.

$$o = Activation(\mathbf{x} + \mathcal{F}(\mathbf{x})) \tag{5}$$

The biggest difference between TCN and ordinary 1D convolution is the use of dilated convolutions. The higher the layer, the larger the convolution window, and the more 'empty holes' in the convolution window. The distinguishing features of TCN are as follows: (1) Convolution in the architecture is causal, which means that no information has been leaked in the past. (2) The architecture can take a sequence of any length and map it to an output sequence of the same length as the same as RNN model.

In addition, we emphasize the combination of deep networks (plus remaining layers) and dilated convolution to build a long effective historical scale.

4 Experimental Evaluation

For evaluating our proposed approach XGBoost-TCN, we have used a 64-bit Windows 10 operating system with 16GB of RAM and an Intel core i7-9750Hz 2.60GHz CPU. The data we use comes from online public datasets. We conducted a model comparison experiment and verified that the XGBoost-TCN model has higher accuracy and faster detection speed than a single TCN model. The experimental results show that, our model has better performance compared with traditional machine learning methods and some deep learning methods.

4.1 Experimental Setup

4.1.1 Experimental Data

The AWID dataset contains a rich mix of normal and attack traffic against IEEE 802.1 1 networks. AWID is larger and newer than NSL-KDD. AWID provides two different sets of data. We chose AWID-CLS-R, which includes an independent training and testing dataset. This dataset provides a class label containing 4 values: normal, flooding, injection and impersonation. The first value corresponds to non-attack situations, and the rest correspond to different types of attacks. The dataset contains 154 features and the training and testing datasets have 1795574 and 575642 samples respectively. After discarding features with null values, constant values and network addresses, we reduce the number of them to 76. Continuous features are scaled in the range from 0 to 1, while all discrete features are one-hot encoded. Besides, this is also a very unbalanced dataset, 91% of normal samples and 9% of abnormal samples, including 2.7% flood, 2.7% simulation and 3.6% injections. Table 1 clearly shows the distribution of the AWID dataset.

Table 1: Distribution of attack class in AWID dataset

Class	Training	Test	Percentage
Impersonation	48,522	20,079	3.6%
Flooding	48,484	8,097	2.7%
Injection	65,379	16,682	2.7%
Total	162,385	44,858	9.0%

Table 2: Settings of parameter

XGBoost		TCN	
Parameters	Value	Parameters	Value
N_estimators	140	SpatialDropout_rate	0.05
Learning_rate	0.2	Hidden_layers	4
Reg_lambda	2	Epoch	20
Booster	gbtree	Batch_size	256
Max_depth	4	Learning_rate	0.000009
Objective	Binary:logistic	N/A	N/A

4.1.2 Model Parameters

After setting up the experimental environment, we also set the basic model parameters of XGBoost and TCN. As shown in Table 2, n_estimators represents the maximum number of weak learner. Generally speaking, if n_estimators is too small, it is easy to underfit, while if n_estimators is too large, it is easy to overfit. Learning_rate makes the model robust by reducing the weight of each step, we set it to 0.2 in XGBoost, but set it to 0.000009 in TCN. Reg_lambda is expressed as the weight coefficient of the L2 Regularization. Booster is used to decide which calculational model to use, we choose 'gbtree' as the boost calculation. Max_depth is the maximum tree depth of the XBoost classifier. Objective specifies the learning task, we use 'Binary:logistic' as our detection target. In the TCN model, there are several unique parameters that affect the performance of the TCN model. SpatialDropout is a dropout method proposed by Tompson *et al.* [20] in the image field. Ordinary dropout will randomly set certain elements to zero, while SpatialDropout will set some areas to zero in a random manner. The effectiveness of this dropout method has been fully proven in the field of image recognition. The number of hidden layers for the model is set to 4, and the number of iterations of the model is set to 20 rounds. Batch_size controls how often the weight of the network is updated, we choose 256 as the parameter of batch_size.

4.1.3 Performance Indicators

Due to the imbalance of the dataset, the learning process of the model is more challenging, because the model will more easily classify all traffic as normal traffic, so as to achieve an accuracy of close to 100%. In order to effectively distinguish the adaptability of our proposed XGBoost-TCN to imbalanced dataset, we not only consider accuracy as an indicator, but also integrate precision, recall and F1_score to evaluate the model. The following are evaluation indicators:

Accuracy, which evaluates the overall success rate of the model in detecting normal and abnormal traffic.

$$Accuracy = \frac{TN + TP}{TP + FP + TN + TP} \tag{6}$$

Precision, which indicates what percent of positive predictions were correct.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

Recall, which defines what percent of positive cases do a classifier catch.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

F1_score, which shows the trade-off between the precision and recall regarding the positive class.

$$F1\_score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

4.2 Performance Evaluation

4.2.1 Parameters Comparison

Here we compare a single TCN model and a TCN model using XGBoost feature selection algorithm. Comparison of model parameters in this paper is shown in Table 3. The shape of the XGBoost-TCN model we proposed in the input layer is (10,1), which is different from (76,1) of the single TCN model. It is crucial to reduce the dimensions of feature for intrusion detection models. The shape of the data through conv1 is (76,128), the parameter size is 384, the shape of the data through dropout1 has not changed, but some convolution kernels are discarded to prevent overfitting. Then the shape of the data through conv_2 is (76,128), and the model parameter size is reached 32,896. In the merge layer, the data of the two channels are merged into one, and the data shape has changed. The shape of a single TCN model is (9728), while the data shape in our model is (1280). After this layer, the parameter size of the model has also changed. A single TCN model has 29,187 parameters. The model parameter of our proposed XGBoost-TCN is 3,843. In terms of the total parameters of the model, the total parameters of our model are close to half of the single TCN model. After the model is compiled, we evaluate the performance of both to detect attack samples.

Table 3: Comparison of parameters

TCN			XGBoost-TCN		
Layer	Shape	Param	Layer	Shape	Param
Input_1	(76,1)	0	Input_1	(10,1)	0
Conv1	(76,128)	384	Conv1	(10,128)	384
Dropout1	(76,128)	0	Dropout1	(10,128)	0
Conv2	(76,128)	32896	Conv2	(10,128)	32896
Dropout2	(76,128)	0	Dropout2	(10,128)	0
Flatten	(9728)	0	Flatten	(1280)	0
Dense	(3)	29187	Dense	(3)	3843
Total Patams	62,467		Total Patams	37,123	
Non-trainable params	0		Non-trainable params	0	

4.2.2 Methods Comparison

Here we compare some typical machine learning methods and deep learning methods on the AWID dataset. The details are as follows.

1) Naive Bayes [3]: Naive Bayes is a supervised learning classifier based on the theorem of Bayes. Using Bayesian rules, the previously calculated probability is combined with the current probability to classify the problem and get the next probability.

2) DT [17]: Decision Tree is an entropy-based classification algorithm that uses pruning technology to remove redundant features. Its computational complexity is low, and the construction rules are easy to understand.

3) DNN [11]: DNN refers to deep neural network. The difference from RNN and CNN is that DNN specifically refers to a fully connected neuron structure, and does not include convolutional units or temporal associations.

4) LSTM [22]: An improved model of intrusion detection based on RNN, using relu activation function, adam optimizer, 100 iterations and double-layer LSTM.

5) Our proposed method: Our method uses the XGBoost feature selection algorithm to perform feature screening on the original dataset, and then uses TCN which is an improved version of the one-dimensional convolution model for classification.

It is worth noting that we select a subset for experiments based on a certain proportion of the training dataset. The proportion of the training dataset is defined as the proportion of training samples. In each experimental dataset, we evaluated the performance of five methods on four indicators. The experimental results in Table 4 show that our proposed XGBoost-TCN is compared with other traditional machine learning methods and deep learning methods. Compared with other methods, the accuracy of our proposed XGBoost-TCN is 0.9396, precision is 0.6436, recall is 0.6622 and f1_scoer is 0.6527. This means that our proposed method is better than other traditional methods in detecting abnormal traffic. The effectiveness of our model on unbalanced samples is because the XGBoost feature selection algorithm effectively removes redundant features, and our TCN model has better feature learning capabilities.

Table 4: Comparison of performance of different methods

Model	Accuracy	Precision	Recall	F1
Naive Bayes	0.8873	0.2957	0.3333	0.3134
DT	0.8678	0.3006	0.3279	0.3127
DNN	0.9183	0.4651	0.6581	0.5180
LSTM	0.8976	0.4228	0.5276	0.4574
Proposed	**0.9396**	**0.6436**	**0.6622**	**0.6527**

4.2.3 Calculation Comparison

We also compare the time performance of our proposed XGBoost-TCN model and a single TCN model. In addition to testing on the AWID dataset, experiments are also conducted on the public dataset NSL-KDD. In Table 5, the time required for each epoch is recorded, and bold font indicates the best result. Experimental results show that XGBoost-TCN proved to be quite competitive.

4.2.4 Space Complexity Performance

The model size is an evaluation indicator of the memory space occupied by the model. Because the model needs to be lightweight on the platform of MEC, the model size should be as small as possible. As shown in Figure 4, our model is not as good as the DNN model, but it is better than the LSTM and TCN models.

Table 5: Comparison of time complexity using NSL-KDD and AWID

Methods	NSL-KDD Dataset		AWID Dataset	
	Training time(s)	Testing time(s)	Training time(s)	Tesing time(s)
TCN	656	4	2382	37
XGBoost-TCN	**224**	**2**	**895**	**18**

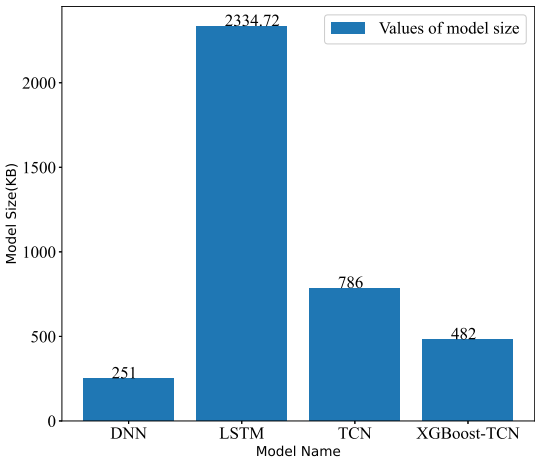


Figure 4: Space complexity performance of different methods

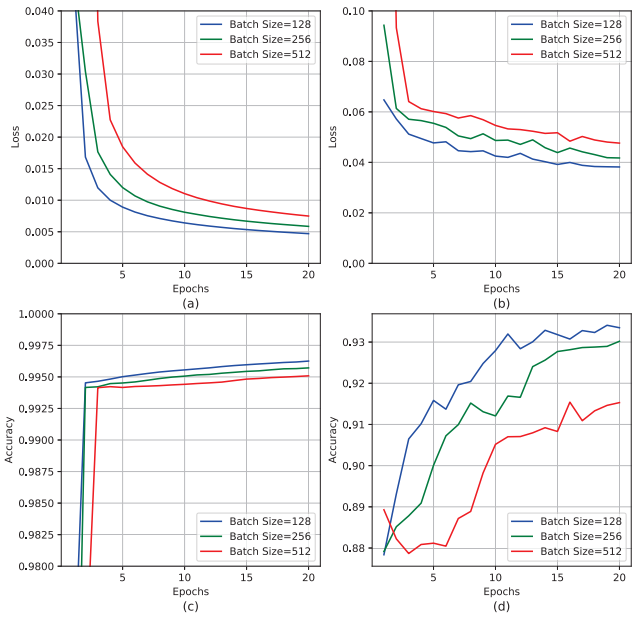


Figure 5: Tuning of batch_size

4.3 Model Analysis

4.3.1 Hyperparameters Study

There are various configurable hyperparameters in the model, such as batch_size, number of convolution kernels, convolutional kernel size, optimizer, etc. These hyperparameters can only be configured manually and cannot be automatically optimized through the training process, which will greatly affect the performance of the model. Among them, the two hyperparameters of batch_size and optimizer are tested. Batch_size is the number of training samples of the neural network after a forward and backward propagation operation, that is, how many samples are used to evaluate the loss in each optimization process. The optimizer is used to optimize the loss and then update the weight parameters. Therefore, we deeply analyzed the influence of these hyperparameters on the performance of our proposed model.

1) The influence of batch_size: As shown in Figure 5, we set the size of batch_size to 128, 256, 512. This is due to the better performance of the GPU to the power of 2 batches. In the experiment, we can see that when the batch_size is 128, the losses of training and verifying will converge faster in the same period, and the best result is 0.038. We know that a smaller batch_size can speed up the optimization in the same period, but it means that more calculation time is required. As the accuracy increases, the amplitude of the training vibration decreases.

2) The influence of optimizer: As shown in Figure 6, we have selected several commonly used optimizers Adam, Adamax, Rmsprop and Nadam for experimental comparison. The effect of using Adamax as an optimizer is the worst, the convergence is slow and the accuracy is only 0.9103. The other three optimizers have the approximative convergence speed. In comparison, Nadam has a higher verification accuracy of 0.9355, which is higher than Adam's 0.9334 and RMSProp's 0.9318. This is because Nadam uses nesterov momentum to replace the traditional momentum in the original Adam, making the gradient update more flexible.

4.3.2 Ablation Study

In order to analyze the effectiveness of each module, we conduct an ablation study on the XGBoost-TCN model. The details of the ablation study based on the AWID dataset are as follows:

1) w/o XGBoost: We remove the XGBoost module of XGBoost-TCN, but keep the TCN module.

2) w/o TCN: We remove the TCN module of XGBoost-TCN, but keep the XGBoost module.

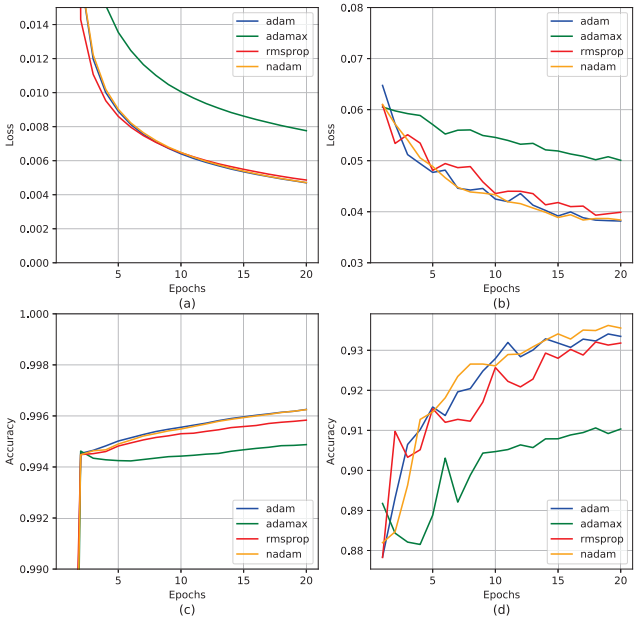


Figure 6: Tuning of optimizer

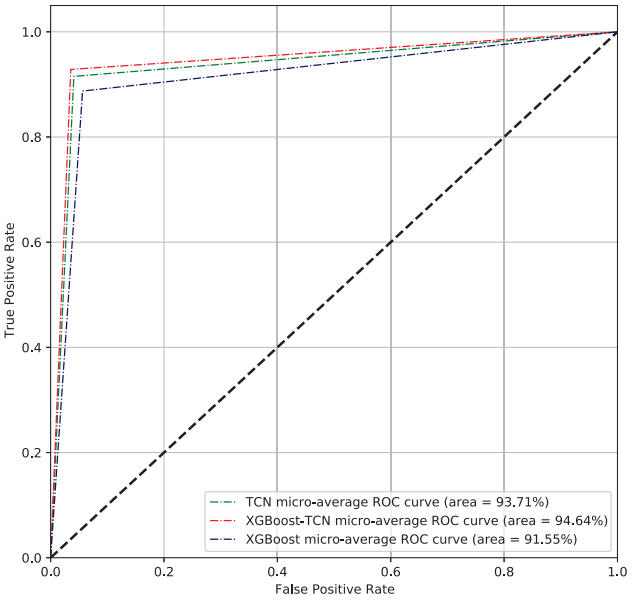


Figure 7: ROC curves of different methods on AWID dataset

We analyze the detailed performance of the XGBoost-TCN model in the ablation study, and the results are shown in Table 6. Compared with the single XGBoost model, the TCN network uses causal convolution and dilated convolution to extract the spatial and temporal features of the dataset. Therefore, the XGBoost model does not mine data well because of only extracting the spatial features of the dataset. However, the single TCN model directly trains the initial 154 traffic features without feature selection, which undoubtedly increases the model training time and redundant features also affect the detection accuracy of the model.

Table 6: The ablation study results on testing dataset

Model	Accuracy	Precision	Recall	F1_score
w/o XGBoost	0.8873	0.2957	0.3333	0.3134
w/o TCN	0.9005	0.4706	0.6229	0.5167
Proposed	**0.9396**	**0.6436**	**0.6622**	**0.6527**

In addition, we also compared the ROC (Receiver Operating Characteristic) curves of the three as shown in Figure 7. The ROC curve of XGBoost-TCN is the closest one to the upper left corner, indicating better generalization ability against the other methods. The results reported above demonstrate that XGBoost-TCN outperforms its competitors. We can conclude that XGBoost-TCN effectively handles the abnormal flow detection problem by the ability to compress the original data to more discriminative abstract features, and XGBoost-TCN is capable of efficient abnormal flow detection.

5 Conclusion and Future Works

This article proposes a new solution framework that combines software-defined security and machine learning to provide end-to-end protection for MEC. In the proposed framework, we use the public AWID dataset as our detection samples, using XGBoost-TCN to detect abnormal behavior from normal behavior. Experiments prove that our anomaly detection framework is effective and has better classification performance. We propose a complete anomaly-based intrusion detection process for IoT devices, which can be used for follow-up research for scholars in the field of research. In the future work, the model can also be improved by adjusting the hyperparameters with swarm intelligent optimization algorithm, such as Particle Swarm Optimization (PSO) algorithm [5] and Artificial Bee Colony (ABC) algorithm. These two algorithms used to automatically tune hyperparameters, and had been proved an efficient method to improve the accuracy of detection.

References

[1] S. A. Althubiti, E. M. Jones, and K. Roy, "Lstm for anomaly-based network intrusion detection," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 2018, pp. 1–3.

[2] S. D. Anton, L. Ahrens, D. Fraunholz, and H. D. Schotten, "Time is of the essence: Machine learning-based intrusion detection in industrial time series data," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2018, pp. 1–6.

[3] N. Ashraf, W. Ahmad, and R. Ashraf, "A comparative study of data mining algorithms for high detection rate in intrusion detection system," *Annals of Emerging Technologies in Computing*, vol. 2, pp. 49–57, 01 2018.

[4] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[5] J. C. Bansal, "Particle swarm optimization," in *Evolutionary and swarm intelligence algorithms*. Springer, 2019, pp. 11–23.

[6] E. Bisong, "Introduction to scikit-learn," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Springer, 2019, pp. 215–229.

[7] E. Carrizosa, C. Molero-Río, and D. R. Morales, "Mathematical optimization in classification and regression trees," *Top*, vol. 29, no. 1, pp. 5–33, 2021.

[8] A. Chawla, B. Lee, S. Fallon, and P. Jacob, "Host based intrusion detection system with combined cnn/rnn model," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 149–158.

[9] Y. Cui, Z. Jin, and J. Hu, "Research on intrusion detection method based on hierarchical self-convergence pca-ocsvm algorithm," *International Journal of Network Security*, vol. 22, no. 6, pp. 916–924, 2020.

[10] Z. Chen, Y. Huang, J. Li, and Y. Gong, "Improving mask learning based speech enhancement system with restoration layers and residual connection." in *INTERSPEECH*, 2017, pp. 3632–3636.

[11] P. Devan and N. Khare, "An efficient xgboost–dnn-based classification model for network intrusion detection system," *Neural Computing & Applications*, no. 3, 2020.

[12] R. H. Dong, H. H. Yan, and Q. Y. Zhang, "An intrusion detection model for wireless sensor network based on information gain ratio and bagging algorithm." *International Journal of Network Security*, vol. 22, no. 2, pp. 218–230, 2020.

[13] A. Harell, S. Makonin, and I. V. Bajić, "Wavenilm: A causal neural network for power disaggregation from the complex power signal," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8335–8339.

[14] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Computers & Security*, vol. 92, p. 101752, 2020.

[15] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1091–1100.

[16] S. Naseer and Y. Saleem, "Enhanced network intrusion detection using deep convolutional neural networks." *TIIS*, vol. 12, no. 10, pp. 5159–5178, 2018.

[17] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, "An anomaly-based intrusion detection system for the smart grid based on cart decision tree," in *Global Information Infrastructure and Networking Symposium (GIIS 2018)*, 2018.

[18] Y. L. S. Rui-Hong Dong and Q. Y. Zhang, "Intrusion detection model based on feature selection and random forest." *International Journal of Network Security*, vol. 23, no. 6, pp. 985–996, 2021.

[19] J. Tan, W. Liu, T. Wang, M. Zhao, A. Liu, and S. Zhang, "A high-accurate content popularity prediction computational modeling for mobile edge computing using matrix completion technology," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 6, p. e3871, 2021.

[20] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 648–656.

[21] Z. Wang, B. Du, and Y. Guo, "Domain adaptation with neural embedding matching," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2387–2397, 2019.

[22] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, pp. 35 365–35 381, 2018.

[23] C. Zhang, X. Wang, F. Li, Q. He, and M. Huang, "Deep learning–based network application classification for sdn," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 5, p. e3302, 2018.

Biography

Xubin Jiao. He received the bachelor's degree from the School of Information Engineering, Xuzhou University of Technology, Jiangsu, China, in 2019. His research interests include Information Security, Intrusion Detection.

Jinguo Li. He received the B.S. degree in Information Security, and the Ph.D. degree in Computer Science and Technology from Hunan University, China, in 2007 and 2014 respectively. His research activities are focused on information security and privacy, applied cryptography.

Mi Wen. She received the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2008, both in computer science. Her research interests include privacy preserving in wireless networks, big data, smart grid, etc.

A Quantitative Assessment Method for Security Risk of IoV Based on Combination Weighting

Peng-Shou Xie, Liang-Xuan Wang, Shuai Wang, Ying-Wen Zhao, Tao Feng, and Yan Yan
(Corresponding author: Liang-Xuan Wang)

School of Computer and Communications & Lanzhou University of Technology
No. 36, Peng Jia Ping Road, Lanzhou, Gansu 730050, China
Email: 2681559167@qq.com

(Received July 21, 2021; Revised and Accepted Jan. 28, 2022; First Online Feb. 26, 2022)

Abstract

Due to the dynamic and changeable network environment of the Internet of Vehicles, the risk value in the access control process and prediction of the risk in the access control process are not accurate enough. This paper proposes a quantitative assessment method of Internet of Vehicles security risks based on combination weighting to solve the above problems better. By analyzing the characteristics of the Internet of Vehicles environment, an Internet of Vehicles security risk assessment index system is established. Combined with the analytic hierarchy process, entropy weight method, and distance function method to determine the weight of each index. The fuzzy evaluation method is used for each index's fuzzy treatment, and each index's risk interval is solved from bottom to top. The simulation experiment shows the feasibility and accuracy of the quantitative assessment method of Internet of Vehicles security risk used in this paper.

Keywords: Blur Processing; Combination Weighting; Internet of Vehicles; Quantitative Assessment; Security Risk

1 Introduction

The Internet of Vehicles(IoV) is a large interactive network that contains information about vehicle location, speed, route, etc. Based on mobile communication and information science technology, the IoV uses wireless communication technology, automotive sensors technology, global-positioning technology, and automobile data recorder technology to complete the data collection of vehicle information and the surrounding environment, data transmission and processing, etc, in order to achieve effective intelligent monitoring, planning, and management of vehicles, people, roads and locations [20].

The IoV industry, on the whole, presents a development trend of ecological, networked, and increasingly close vertical connection among enterprises. Network security has become a prerequisite for the healthy and sustainable development of the IoV industry [12]. With the increasing penetration rate of intelligent and connected vehicles, cyber security incidents occur frequently. In 2019 alone, the proportion of automobile security incidents caused by cyber security issues is as high as 57-percent. At present, due to the dynamic and changeable network environment of the IoV, it is not accurate to identify the risk value in the process of access control and predict the risk in the process of access control.

After identifying the risks faced by the IoV, the risks need to be evaluated, the evaluation results will be used as the basis for risk strategy formulation and risk monitoring. Reference [17] for vehicle driver for operational risk, different driving the corresponding risk size is not the same problem, adopt the improved entropy weight - Analytic Hierarchy Process (AHP), to determine the weight of every evaluation index, combining the evaluation index of the comprehensive score and weight, it is concluded that the motorist driving behavior score. It can prompt drivers to develop good driving habits and reduce the incidence of traffic accidents. Reference [9], the combination of entropy weight method and order relation analysis method was used to determine the index weight coefficient, and then the fuzzy comprehensive evaluation method was used to carry out a comprehensive intelligent quantitative evaluation on the autonomous driving vehicles. The combined weight method is adopted to make full use of the acquired objective information to determine weights of various indexes of autonomous driving vehicles more accurately and objective, to obtain scientific and reasonable evaluation results. Reference [1] expands the work-based access control model by combining the risk assessment process with the system's trust level of visitors. The risk determination of this model is related to the user's trust level and the security level of the requested object and is weighted by the risk threshold of risk preference scenario conditions. Although researchers at home and abroad have done a lot of research on risk assessment algorithms, there are still problems of low accuracy of quantitative risk assessment methods.

In view of the above problems, this paper studies and analyzes the evaluation principles and methods in access control, and analyzes the situation of attribute-based access control in the IoV environment. By studying the influence of subject attributes, object attributes, and environmental attributes on risks in access control, and taking them as important analysis factors to establish a risk assessment index system, we can judge their influence on risks in the access control process. The risk assessment method is used to evaluate the risk index and provide a more accurate decision basis for access control.

2 Security Risk Assessment Model and Risk Assessment System of IoV

2.1 Security Risk Assessment Model of IoV

By analyzing the security status of the IoV and referring to the principles of scientifically, systematicity, hierarchy, and comprehensiveness, the IoV security risk assessment index system is constructed, and the IoV security risk assessment index is taken as the risk input. Using the AHP to calculate the subjective weight, entropy weight method to calculate the objective weight, distance function method to calculate the combination weight, the fuzzy evaluation method is used to establish the index evaluation set, and the risk interval of the underlying index is calculated, combined with the combination weight and the risk interval of the bottom index, the corresponding risk interval of the upper index is solved from bottom to top, and the comprehensive evaluation result of the whole risk assessment system is obtained. On this basis, the security risk assessment model of IoV established in this paper is shown in Figure 1.

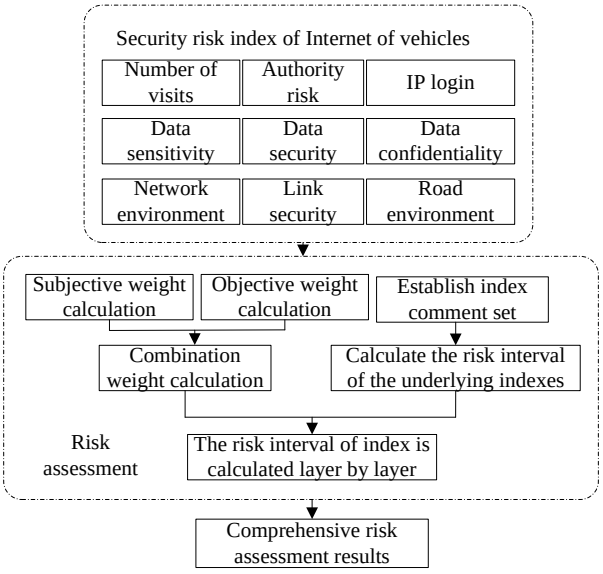


Figure 1: Model diagram of security risk assessment for IoV

2.2 Security Risk Assessment Index System of IoV

Nothing can be accurately controlled only by qualitative analysis, thus the security risks of the Internet of vehicles need to be quantified to improve accurate risk management, a comprehensive evaluation index system can be constructed. Only through quantitative analysis can various factors be quantified and the influence degree of things can be intuitively reflected, to effectively control different risk indexes. Based on the principles of scientificity, systematism, hierarchy, and comprehensivity [16],this paper establish a security risk assessment index system for IoV. Influential factors of the IoV are complex, this paper analyzes the main risk factors of the IoV from three aspects: subject attribute, object attribute, and environment attribute. The IoV security risk assessment indexes are taken as the target layer and decomposed successively. Subject attribute, object attribute, and environmental attribute are set as the criterion layer, and the criterion layer is decomposed and set as the index layer, which is mainly summarized as the following specific factors. It is divided into several visits, authority risk, and IP login under the main attributes. The object attributes can be divided into data sensitivity, data security, and data confidentiality. It can be divided into a network environment, link security, and road environment. Therefore, the security risk index system of the IoV established is shown in Figure 2.

3 Quantitative Assessment of Security Risk of IoV Based on CombinationWeight

3.1 Subjective Weight Calculation

3.1.1 Analytic Hierarchy Process

AHP [4, 5, 14, 15] is a decision-making method that uses quantitative analysis. It uses the same level of factors in the AHP to conduct a pairwise comparison of importance. Using each other as a ruler, the importance is compared, that is, the same level of elements. Compare i(i=1,2,...,n) with element j(j=1,2,...,n), get X_{ij} according to the importance, and determine the judgment matrix. The quantified value of importance is shown in Table 1.

After the judgment matrix is established, the specific maximum eigenvector λ_{max} of each matrix is solved, and the eigenvector is normalized to get. The consistency index was calculated CI as shown in Equation (1).

$$CI = \frac{\lambda_{\max} - n}{n-1} \tag{1}$$

Table 1: Quantitative value of factor comparison

Factor i is better than factor j	Meaning	Quantized value
Equally important	Factor i is as important as factor j	1
Slightly important	Factor i and factor j are slightly more important	3
Stronger important	Factor i and factor j are more important	5
Strongly important	Factor i and factor j are highly important	7
Extremely important	Factor i and factor j are very important	9
The middle value of two adjacent judgments	The middle value of two adjacent judgments	2,4,6,8

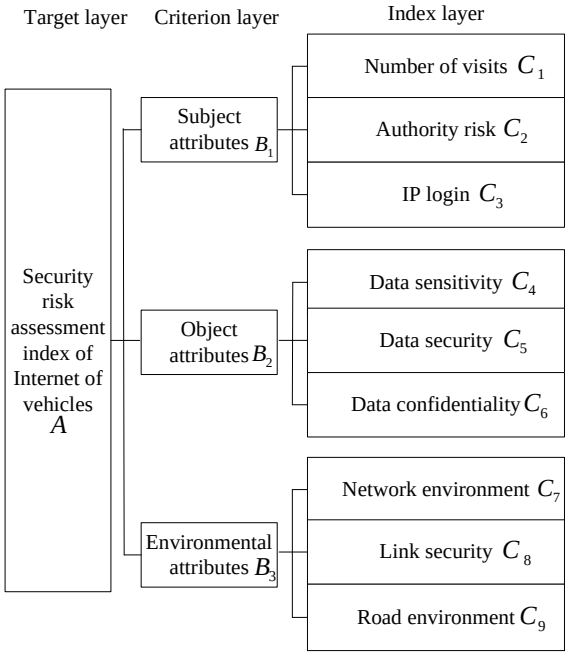


Figure 2: Security risk index system of IoV

The consistency of the matrix can be judged by the introduction of CR the tested coefficient when judging the consistency of the matrix. CR is the ratio of consistency index to random consistency index , as shown in Equation (2), the values of RI are shown in Table 2. When $CR < 0.1$ the consistency of the matrix is considered to be satisfied and the subjective weight Wsi is obtained. Otherwise, the consistency test is not established and the judgment matrix is reconstructed.

$$CR = \frac{CI}{RI} \tag{2}$$

3.1.2 Subjective Weight Calculation Based on Analytic Hierarchy Process

Next, calculate the subjective weight W_{si} of the feature in the order of B_j to A, C_i to B_j (j=1,2,3), and C_i to A, as follows. The judgment matrix of the criterion layer B_1-B_3 to target layer A is shown in Equation (3).

$$A = \begin{bmatrix} 1 & 2 & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ 2 & 2 & 1 \end{bmatrix} \tag{3}$$

The maximum eigenvalue λ_{max}=3.0356 can be obtained from the Equation $A\zeta = \lambda_{\max}\zeta$ the normalized feature vector B_j(j=1,2,3) is shown in Equation (4).

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} 0.3108 \\ 0.1958 \\ 0.4934 \end{bmatrix} \tag{4}$$

Calculate $CR = 0.0516 < 0.1$ and pass the consistency test to determine that the subjective weight of the indexes required by the model criterion layer B to the target layer A is $\begin{bmatrix} W_{sB_1} & W_{sB2} & W_{sB_3} \end{bmatrix} = \begin{bmatrix} 0.3108 & 0.1958 & 0.4934 \end{bmatrix}$

The judgment matrix of criterion level C_1-C_3 to B_1 is shown in Equation (5).

$$B_1 = \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{4} \\ 3 & 1 & \frac{1}{2} \\ 4 & 2 & 1 \end{bmatrix} \tag{5}$$

The maximum eigenvalue is λ_{max}=3.0183, and the eigenvector is as shown in Equation (6).

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 0.1220 \\ 0.3196 \\ 0.5584 \end{bmatrix} \tag{6}$$

Calculate $CR = 0.0176 < 0.1$ and pass the consistency test to determine the subjective weight of the actually required index of the model index level C to the criterion level B_1 is $\begin{bmatrix} W_{sC_1} & W_{sC_1} & W_{sC_3} \end{bmatrix} = \begin{bmatrix} 0.122 & 0.3196 & 0.5584 \end{bmatrix}$.

The judgment matrix of criterion level C_4-C_6 to B_2 is shown in Equation (7).

$$B_2 = \begin{bmatrix} 1 & 6 & 3 \\ \frac{1}{6} & 1 & \frac{1}{3} \\ \frac{1}{3} & 3 & 1 \end{bmatrix} \tag{7}$$

The maximum eigenvalue is λ_{max}=3.0183, and the eigenvector is shown in Equation (8)

$$\begin{bmatrix} C_4 \\ C_5 \\ C_6 \end{bmatrix} = \begin{bmatrix} 0.6548 \\ 0.0953 \\ 0.2499 \end{bmatrix} \tag{8}$$

Table 2: The values of RI

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

$CR = 0.0176 < 0.1$, through the consistency check, it is determined that the subjective weight of the actual indexes required by the model index level C to the criterion level B_2 is $\left[\begin{array}{ccc} W_{sC_4} & W_{sC_5} & W_{sC_6} \end{array}\right] = \left[\begin{array}{ccc} 0.6548 & 0.0953 & 0.2499 \end{array}\right]$

The judgment matrix of criterion level C_7-C_9 versus $B_3$is shown in Equation (9).

$$B_3 = \left[\begin{array}{ccc} 1 & \frac{1}{3} & \frac{1}{2} \\ 3 & 1 & 3 \\ 2 & \frac{1}{3} & 1 \end{array}\right] \tag{9}$$

The maximum eigenvalue is λ_{max}=3.0536, and the eigenvector is shown in Equation (10).

$$\left[\begin{array}{c} C_7 \\ C_8 \\ C_9 \end{array}\right] = \left[\begin{array}{c} 0.1571 \\ 0.5936 \\ 0.2493 \end{array}\right] \tag{10}$$

$CR = 0.0516 < 0.1$, through the consistency check, it is determined that the subjective weight of the actually required indexes of the model index level C to the criterion level B_3 is $\left[\begin{array}{ccc} W_{sC_7} & W_{sC_8} & W_{sC_9} \end{array}\right] = \left[\begin{array}{ccc} 0.1571 & 0.5936 & 0.2493 \end{array}\right]$.

It is necessary to check the consistency of the overall ranking of levels, and the specific detection process is shown in Equations (11) (12) (13).

$$CI = \sum_{j=1}^{3} W_{sB_j} \times CI_j = 0.0178 \tag{11}$$

$$RI = \sum_{j=1}^{3} W_{sB_j} \times RI_j = 0.5801 \tag{12}$$

$$CR = \frac{CI}{RI} = 0.0307 < 0.1 \tag{13}$$

Therefore, the consistency test of the total ranking of the levels is passed, and the weight of the criterion layer C_i to the target layer A is set as the subjective weight W_{si}, and the specific calculation method is shown in Equation (14).

$$W_{si} = \begin{cases} C_i \times W_{sB_1}; i = 1,2,3 \\ C_i \times W_{sB_2}; i = 4,5,6 \\ C_i \times W_{sB_3}; i = 7,8,9 \end{cases} \tag{14}$$

The subjective weights calculated according to Equation (14) are shown in Table 3.

Table 3: Subjective weights

Index	B_1	B_2	B_3	W_{si}
	0.3108	0.4934	0.1958	
C_1	0.1220	/	/	0.0379
C_2	0.3196	/	/	0.0993
C_3	0.5584	/	/	0.1736
C_4	/	0.6548	/	0.3231
C_5	/	0.0953	/	0.0470
C_6	/	0.2499	/	0.1233
C_7	/	/	0.1571	0.0308
C_8	/	/	0.5936	0.0488
C_9	/	/	0.2493	0.1162

3.2 Objective Weight Calculation

3.2.1 Entropy Weight Method

In order to eliminate the subjective factor in the weight of the AHP solution, the entropy weight method [7, 18, 19, 22, 23] should be introduced to determine the weight coefficient of each index, which should be determined according to the information provided by each index.

1) Construct judgment matrix

If samples are evaluated through n evaluation indexes, and the corresponding index value is r_{ij}(i=1,2,...,m;j=1,2,...,n), then the index matrix is $R = (r_{ij})_{m\times n}$, which represents the evaluation value of thej-th expert on the i-th index, as shown in Equation (15) Shown.

$$R = \left[\begin{array}{ccc} r_{11} & ... & r_{1n} \\ ... & ... & ... \\ r_{m1} & r_{m2} & r_{m3} \end{array}\right] \tag{15}$$

2) Normalized index matrix

Due to the differences in the nature and magnitude of the evaluation indexes, the index matrix R should be normalized to obtain the dimensionless index matrix as shown in Equations (16) and (17).

$$X = (x_{ij})_{m\times n} \tag{16}$$

$$x_{ij} = \frac{r_{ij}}{\sqrt{\sum_{i=1}^{m} (r_{ij})^2}}, i = 1,2,...,m; j = 1,2,...,n \tag{17}$$

3) Determine the weight of objective indexes

The information entropy value e_j of the j-th index of the sample is shown in Equation (18).

$$e_j = \frac{1}{\ln m} \sum_{i=1}^{m} a_{ij} \ln a_{ij}, i = 1,2,...,m; j = 1,2,...,n \tag{18}$$

In the above Equation, a_{ij} represents the proportion of the i-th sample under the j-th index, as shown in Equation (19).

$$a_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}}, i = 1, 2, ..., m; j = 1, 2, ..., n \quad (19)$$

Then the entropy weight W_{oi} of the j-th index is shown in Equation (20).

$$W_{oi} = \frac{1 - e_j}{n - \sum_{j=1}^{n} e_j}, i = 1, 2, ..., m; j = 1, 2, ..., n \quad (20)$$

3.2.2 Objective Weight Calculation Based on Entropy Weight Method

In this paper, the established security risk assessment indexes of the IoV are taken as the research object, and the entropy weight method is used to evaluate by using Matlab software. the objective weights are shown in Table 4.

Table 4: Objective weights

Index	B_1	B_2	B_3	W_{si}
	0.2445	0.0633	0.6923	
C_1	0.3124	/	/	0.1110
C_2	0.4357	/	/	0.1548
C_3	0.2519	/	/	0.0895
C_4	/	0.4441	/	0.15801
C_5	/	0.3617	/	0.1287
C_6	/	0.1942	/	0.0691
C_7	/	/	0.3326	0.0961
C_8	/	/	0.3008	0.0869
C_9	/	/	0.3666	0.1059

3.3 Combination Weight Calculation

3.3.1 Distance Function Method

The expression of the distance function [21] for both subjective and objective is shown in Equation (21).

$$\mathrm{d}(W_{\mathrm{si}}, W_{\mathrm{oi}}) = \sqrt{\frac{1}{2}\sum_{\mathrm{i}=1}^{\mathrm{n}} (W_{si} - W_{oi})^2} \quad (21)$$

Suppose the combination weight is W_{ci}, α is the subjective weight distribution coefficient, and β is the objective weight coefficient, then the calculation Equation of the combination weight is shown in (22).

$$W_{ci} = \alpha W_{si} + \beta W_{oi} \quad (22)$$

In order to reduce the difference between subjective and objective weights, make the distribution coefficient equal to the distance function, as shown in Equation (23).

$$\begin{cases} \alpha + \beta = 1 \\ \mathrm{d}(W_{si}, W_{oi})^2 = (\alpha - \beta)^2 \end{cases} \quad (23)$$

3.3.2 Combination Weight Calculation Based on Distance Function Method

The values of α and β can be obtained from Equations (21), (22), and (23) as shown in Equation (24).

$$\alpha = \sqrt{\frac{1}{8}\sum_{i=1}^{n} (W_{si} - W_{\mathrm{oi}})^2} + \frac{1}{2} = 0.59 \quad (24)$$

Combine Equation α+β=1and Equation (24) to get β=1-α=1-0.59=0.41. Put the values of α and β into Equation (22) to get the combineation weights as shown in Table 5.

Table 5: Combination weights

Index	B_1	B_2	B_3	W_{si}
	0.2836	0.3171	0.3994	
C_1	0.2001	/	/	0.0679
C_2	0.3672	/	/	0.1221
C_3	0.4327	/	/	0.1391
C_4	/	0.5684	/	0.2554
C_5	/	0.2045	/	0.0805
C_6	/	0.2271	/	0.1011
C_7	/	/	0.2291	0.0576
C_8	/	/	0.4736	0.0644
C_9	/	/	0.2974	0.1120

3.4 Calculation of Comprehensive Evaluation Results of Security Risk Index of IoV

3.4.1 Fuzzy Evaluation Method

The fuzzy evaluation method [6, 8, 10] is used to carry out fuzzy processing [0,1] on the evaluation of each risk index. For each index, the quantitative evaluation set is designed asV=[V_1,V_2,...,V_t], The t here is the number of ratings. Suppose there are n experts participating in an evaluation process, and there are t evaluation grades V=[V_1,V_2,...,V_t] for one of the indexes x_j at the bottom of the evaluation system, the index corresponding to the comment is assigned a value of continuous spacing [a_0,a_1],[a_1,a_2],...,[a_{t-1},a_t], is the number of experts who choose an evaluation grade of l (l=1,2,...,t,$\sum_{l=1}^{t} c_l$=n). Then the evaluation interval of the evaluation index $x_j$is V_{x_j}, which is expressed by Equation (25).

$$V_{x_j} = \sum_{l=1}^{t} (\frac{c_l}{n} \times [a_{l-1}, a_l]) \quad (25)$$

3.4.2 Calculation of Comprehensive Evaluation Results Based on Fuzzy Evaluation

Through the above process, we obtained the combined weight of each index layer [13] and then used the fuzzy

evaluation method to design the evaluation set for the importance of indexes. Considering that people's subjective evaluation results are more suitable to be expressed in the form of the interval, this paper uses interval fuzzy numbers to construct the fuzzy evaluation set of indexes, as shown in Equation (26).

$$\begin{aligned} V &= \{V_1, V_2, V_3\} \\ &= \{lowrisk,\ mediumrisk,\ highrisk\} \\ &= \{[0, 0.3], [0.3, 0.6], [0.6, 1]\} \end{aligned} \quad (26)$$

In this evaluation, the risk assessment questionnaire is constructed based on the index layer in the risk system, and 100 experts were used to evaluate 9 indexes. The evaluation results of risk indexes are shown in Table 6.

Table 6: Risk index evaluation results

Index	Low risk	Medium risk	High risk
C_1	40	52	8
C_2	37	58	5
C_3	49	41	10
C_4	53	43	4
C_5	46	48	6
C_6	23	57	20
C_7	31	58	12
C_8	19	60	21
C_9	54	37	9

Data processing was carried out on the evaluation results to solve the risk interval of each risk assessment index, as shown in Equation (27). Similarly, the risk interval of each index in the index layer is shown in Table 7.

$$\begin{aligned} V_{C_1} &= \{\frac{40}{100}V_1 + \frac{52}{100}V_2 + \frac{8}{100}V_3\} \\ &= [0.2040, 0.5120] \end{aligned} \quad (27)$$

According to the risk evaluation interval of the index level and the combination weight, through grouping calculation, the risk evaluation interval of the criterion level index can be obtained from the bottom up, as shown in Equation (28) (29) (30).

$$V_{B_1} = (V_{C_1}, V_{C_2}, V_{C_3}) \begin{bmatrix} W_{C_1} \\ W_{C_2} \\ W_{C_3} \end{bmatrix} = [0.1949, 0.5027] \quad (28)$$

$$V_{B_2} = (V_{C_4}, V_{C_5}, V_{C_6}) \begin{bmatrix} W_{C_4} \\ W_{C_5} \\ W_{C_6} \end{bmatrix} = [0.1899, 0.7026] \quad (29)$$

$$V_{B_3} = (V_{C_7}, V_{C_8}, V_{C_9}) \begin{bmatrix} W_{C_7} \\ W_{C_8} \\ W_{C_9} \end{bmatrix} = [0.2476, 0.5617] \quad (30)$$

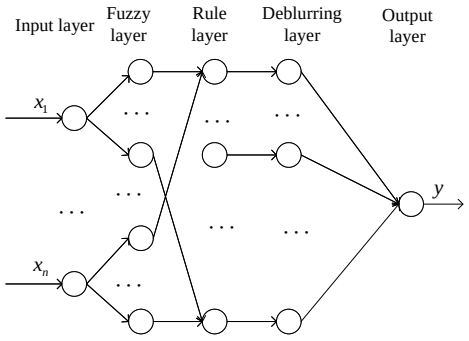


Figure 3: Structure diagram of fuzzy neural network

The final risk index evaluation interval is shown in Equation (31).

$$V_A = (V_{B_1}, V_{B_2}, V_{B_3}) \begin{bmatrix} W_{B_1} \\ W_{B_2} \\ W_{B_3} \end{bmatrix} = [0.2144, 0.5914] \quad (31)$$

Through calculation up layer by layer, the overall risk value evaluation interval that can be finally calculated into the security risk evaluation index system of the IoV is V_A=[0.2144,0.5914]. The risk degree of the system is below the medium level, and there are certain security risks. Appropriate measures can be taken to enhance the security protection of the system.

4 Simulation Experiment

4.1 Adaptive Fuzzy Neural Network Model

Fuzzy neural network [10,11] combines fuzzy system with the neural network, and fully considers their complementarity. The fuzzy system relies heavily on the experience and knowledge of experts or operators, and lacks the key abilities of self-learning and self-adaptation, while the neural network can directly learn effectively from the samples. An adaptive neuro-fuzzy reasoning system is to establish fuzzy rules and membership function by learning verified data and using a data-based modeling method. The adaptive fuzzy neural network used in this paper realizes risk adaptability, the fuzzy structure of the neural network based on the T-S model is shown in Figure 3.

4.2 Adaptive Fuzzy Neural Network Model

MATLAB R 2016b was used in the experiment. AMD Ryzen 5 4600H processor with Radeon Graphics 3.00GH, 16GB memory, and Microsoft Windows 10 64-bit operating system were used in the equipment. The results of Table 6 and each index risk interval are fuzzily processed into the form of point values for the convenience of

Table 7: Subjective weights

Index	C_1	C_2	C_3
Risk range	V_{C_1} = [0.2040,0.5120]	V_{C_2} = [0.2040,0.5090]	V_{C_3} = [0.1830,0.4930]
Index	C_4	C_5	C_6
Risk range	V_{C_4} = [0.1530,0.8170]	V_{C_5} = [0.1800,0.4860]	V_{C_6} = [0.2910,0.6110]
Index	C_7	C_8	C_9
Risk range	V_{C_7} = [0.2460,0.5610]	V_{C_8} = [0.3000,0.6170]	V_{C_9} = [0.1650,0.4740]

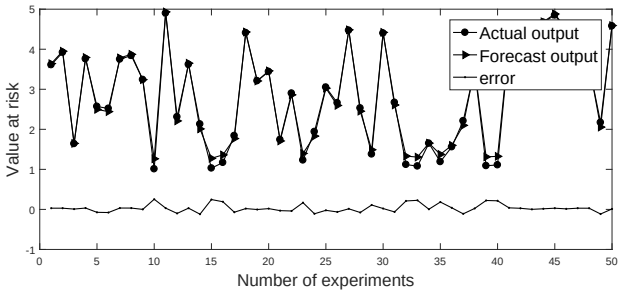


Figure 4: Comparison of experimental results and predicted results

training and testing, the sample will evaluate the subject attribute, object attribute, and environment attribute in the access control process. According to the sample data and prior knowledge, the three input language components of risk prediction are divided into three categories. The experiment adopts three input data with three language variables each. In order to make the results closer to the displayed value, 1000 groups of samples are selected as the training values and 500 groups as the test group.

4.3 Experimental Results

After several risk assessment tests based on fuzzy neural network, the relevant parameters of the network are set as follows after comparing the experimental process and results: (1) Gaussian membership function was selected, the number of fuzzy language variables was 5, and the training times were 100. (2) The T-S model of the fuzzy neural network is adopted in the experiment, the number of language variables in the input layer is 3, that is, there are 3 risk-related elements in the structural model, the output number is 1. A comparison between the value obtained from multiple experiments and the actual risk value is shown in Figure 4.

As can be seen from Figure 4, the error between the predicted value of the simulation experiment result and the actual risk assessment result is very small, which indicates that the quantitative assessment method of the security risk of the IoV used in this paper is feasible.

In order to further test the performance of the algorithm in this paper, the same data were used to conduct comparative tests on the four methods in reference [2,3,10,11], and the average result of ten experiments of each algorithm was taken as the final result, as shown in Table 8.

It can be seen from the experimental results that the prediction result of the algorithm in this paper has a relatively small error compared with that in reference [2, 3, 10, 11], and the risk can be evaluated more accurately, which indicates that the algorithm used in this paper has good accuracy in risk assessment.

5 Conclusion

The main content of this paper is to build the security risk index system of the IoV based on the principles of scientificity, systematicness, hierarchy, and comprehensibility. On this basis, a security risk assessment model of the IoV is established. AHP, entropy weight method, and distance function method are used to weight the security risk indexes of the IoV, and fuzzy evaluation method is used to establish the index evaluation set and calculate the risk interval of each index, so as to make an accurate decision basis for access control. Through the simulation experiment, the feasibility and accuracy of the quantitative assessment method of the security risks of the IoV used in this paper are verified. Although this paper has carried out relevant research work, there are still the following deficiencies: In the application process after the construction of the risk assessment model, due to the lack of a large number of information data, it is impossible to use the actual data for statistical analysis of the influence degree of each factor. Instead, it can only use the way of expert rating to score the influence degree of each factor and construct the judgment matrix. The establishment of the security risk index system of the IoV is not comprehensive enough, and a more perfect security risk index system of the IoV needs to be established.

Acknowledgments

This research is supported by the National Natural Science Foundations of China under Grants No.61862040, No.61762059 and No.61762060.The authors gratefully acknowledge the anonymous reviewers for their helpful comments and suggestions.

Table 8: Comparison results

Number	Target value	Reference 2	Reference 3	Reference 11	Reference 12	This paper
1	3.3	3.264	3.322	3.272	3.278	3.319
2	3.8	3.786	3.901	3.78	3.823	3.802
3	2.4	2.366	2.371	2.404	2.406	2.405
4	6.4	6.417	6.354	6.402	6.41	6.415
5	4.6	4.566	4.692	4.576	4.578	4.589
6	6.1	5.893	6.201	5.991	6.071	6.121
7	5.3	5.278	5.321	5.241	5.239	5.311
8	3.4	3.241	3.412	3.33	3.361	3.378
9	4.6	4.521	4.69	4.417	4.641	4.59
10	4.2	4.276	4.198	4.272	4.23	4.189
Average error	/	6.78%	5.16%	3.88%	2.83%	1.27%

References

[1] M. Abomhara, G. M. Kien, and V. A. Oleshchuk, "Towards risk-aware access control framework for healthcare information sharing," in *4th International Conference on Information Systems Security and Privacy*, 2018.

[2] P. Cao and H. L. Yu, "Initial safety risk assessment of special stratum shield based on ahp-fuzzy comprehensive evaluation method(in chinese)," *Construction Technology*, vol. 50, no. 17, pp. 80–86, 2021.

[3] D. C. Chen, "Risk assessment of anti-explosion safety performance of buildings in chemical industrial park based on improved fuzzy analytic hierarchy process(in chinese)," *Safety and Environmental Engineering*, vol. 28, no. 06, pp. 52–60+66, 2021.

[4] D. D. Chen. "Research on water resources security of zhengzhou city based on ahp-entropy comprehensive evaluation method(in chinese),". Master's thesis, North China University of Water Resources and Hydropower, 2020.

[5] W. Chen and X. J. Liu, "Multi-factor reputation evaluation model based on analytic hierarchy process in vehicle ad-hoc networks (in chinese)," *JOURNAL OF ZHEJIANG UNIVERSITY (ENGINEERING SCIENCE)*, vol. 54, no. 4, p. 10, 2020.

[6] Y. K. Chen. "Research on risk management of a construction project based on fuzzy comprehensive evaluation (in chinese),". Master's thesis, Suzhou University, 2020.

[7] T. L. Gao, T. Li, and R. Jiang, "Research on cloud service security measurement based on information entropy," *International Journal of Network Security*, vol. 21, no. 6, pp. 1003–1013, 2019.

[8] L. Li. *Research on Risk Assessment Model of Water Traffic Meteorological Disaster Risk(in Chinese)*. PhD thesis, Hunan Normal University, 2019.

[9] R. Li, Y. L. Ma, and H. Tian, "Comprehensive intelligent quantitative evaluation of autonomous vehicle based on entropy and g1 methods(in chinese)," *AUTOMOTIVE ENGINEERING*, vol. 42, no. 10, p. 8, 2020.

[10] S. Y. Li. *Research on Information Security Risk Assessment Method Based on Improved Neural Network (in Chinese)*. PhD thesis, China University of Mining and Technology.

[11] S. R. Liu. *Research on Risk Adaptive Access Control of Large Data Applications(in Chinese)*. PhD thesis, Inner Mongolia University of Science and Technology.

[12] W. Meng, P. Cofta, and T. Grandison, "Editorial for special issue on security, trust, and privacy in internet of things: Challenges and solutions," *International Journal of Network Management*, 2020.

[13] A. Nk and B. Wm, "An integrated approach of fuzzy risk assessment model and data envelopment analysis for route selection in multimodal transportation networks," *Expert Systems with Applications*, vol. 171, 2020.

[14] H. Song, X. Lu, and Q. Wu, "Weight calculation method for consumer goods risk assessment indexes based on analytic hierarchy process," in *IOP Conference Series: Earth and Environmental Science*, vol. 440, p. 042001. IOP Publishing, 2020.

[15] G. Wang, "Research on network security risk assessment method based on improved analytic hierarchy process," *International Journal of Network Security*, vol. 23, no. 3, pp. 515–521, 2021.

[16] M. N. Wang, X. H. Guo, and G. B. Ni, "Ahp entropy power method based on the railroad tunnel single and double hole selection decision research(in chinese)," *Journal of Railway Engineering*, vol. 36, no. 11, p. 6, 2019.

[17] Z. H. Wang. *Research on Auto Insurance Pricing based on UBI Driving Behavior Score(in Chinese)*. PhD thesis, Hunan University, 2019.

[18] L. B. Wen, "Security evaluation of computer network based on hierarchy," *International Journal of Network Security*, vol. 21, no. 5, pp. 735–740, 2019.

[19] O. Y. Wu, Y. Jin, and Z. L. Liu, "Performance evaluation of water lubricated stern bearings based on entropy weight fuzzy comprehensive evaluation method(in chinese)," *CHINA MECHANICAL ENGIN EERING*, vol. 31, no. 12, p. 8, 2020.

[20] P. S. Xie, H. J. Fan, and T. Feng, "Adaptive access control model of vehicular network big data based on xacml and security risk," *International Journal of Network Security*, vol. 22, no. 2, pp. 347–357, 2020.
[21] P. S. Xie, C. Fu, and T. Feng, "Malicious attack detection algorithm of internet of vehicles based on cw-knn," *International Journal of Network Security*, vol. 22, no. 6, pp. 1004–1014, 2020.
[22] M. Yang, R. Jiang, and T. L. Gao, "Research on cloud computing security risk assessment based on information entropy and markov chain," *International Journal of Network Security*, vol. 20, no. 4, pp. 664–673, 2020.
[23] M Yang and J. Li, "Research on privacy security risk evaluation of mobile commerce users based on information entropy and markov theory," *International Journal of Network Security*, vol. 23, no. 5, pp. 807–816, 2021.

Biography

Peng-shou Xie was born in Jan. 1972. He is a professor and a supervisor of master student at Lanzhou University of Technology. His major research field is Security on Internet of Things.E-mail: xiepsh_lut@163.com .

Liang-xuan Wang was born in Mar.1995. She is a master student at Lanzhou University of Technology. Her major research field is network and information security. E-mail: 2681559167@qq. com

Shuai Wang was born in Aug.1994. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 627858493@qq. com

Ying-wen Zhao was born in Feb.1996. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 1376144882@qq. com

Tao Feng was born in Dec. 1970. He is a professor and a supervisor of Doctoral student at Lanzhou University of Technology. His major research field is modern cryptography theory, network and information security technology. E-mail: fengt@lut.cn

Yan Yan was born in Oct.1980.She is a ass-ociate professor and a supervisor of master student at Lanzhou University of Technology.Hermajor research field is privacy protection, multimedia information security.E-mail:yanyan@lut.cn

Mobile RFID Authentication Protocol Based on Permutation Cross Synthesis for Anti Counterfeit Attack

Shan-Hua Zhan[1] and Chun-Qiang Yu[2]

(Corresponding authors: Shan-Hua Zhan and Chun-Qiang Yu)

Department of Information Management, Guangdong Justice Police Vocational College[1]
Guangzhou 510006, China
Email: zsh_87@126.com
Guangxi Key Laboratory of Multi-Source Information Mining & Security, Guangxi Normal University[2]
Guilin 541004, China
Email: yu_chunqiang@126.com

(Received Dec. 18, 2020; Revised and Accepted Dec. 27, 2021; First Online Feb. 26, 2022)

Abstract

With the development of society and the increment of human needs, many new technologies have been produced. The mobile RFID system has gradually approached the scene view of humans—the wire communication between reader and database limits modern communication in a traditional RFID system. While wireless communication between these two characters can satisfy the need in a mobile RFID system, specific safety risk still exists. An ultra-lightweight two-way authentication protocol of mobile RFID system has been designed in this paper. The protocol is based on permutation cross synthesis operation, which is from bit operation to encrypt sent messages. Besides, permutation cross synthesis includes two steps, permutation operation, and cross synthesis operation. With Hamming Weights of encrypted messages, permutation and cross-location can be identified. Communication entities in this protocol are identified first before privacy information is exchanged. Meanwhile, one end of the reader stores the communication keys for the first and second rounds, harmonizing communication entities. Compared with other protocols of this type on safety and function, the protocol in this paper has characteristics of lower computation, higher safety performance, and higher applicability on mobile RFID systems.

Keywords: Counterfeit Attack; Internet of Things; Mobile Authentication; Mobile System; RFID; Permutation Cross Synthesis (Pcs)

1 Introduction

RFID has been used in many areas. During operation, by using this technology, information can be identified without touching entities, which is suitable for many near field communication, especially the application of RFID system. Usually, there are total three parts included in RFID system: tag, reader, and server [6,9]. In traditional RFID system, reader is fixed, which means communication between reader and server is based on wire transmission, considered safe and reliable. With technologies develop and society advances, human need has increased, and traditional RFID system has gradually not satisfied people's complex demand, which causes the production of mobile RFID system. Instead of being fixed, reader in mobile RFID system is mobile, which means communication between reader and server is changed into wireless. Wireless communication is a dangerous way due to its own inherent property, which is easily spied by attackers or by other types attack, considered as low safety [3, 11, 17].

It is necessary to assure the safety of information during communication in order to make sure that mobile RFID system can be promoted and applied. However, the two-way authentication protocol in traditional RFID system cannot be applied perfectly in mobile RFID system so that a new two-way authentication protocol should be designed for mobile RFID system [1, 2, 13].

In [14], an authentication protocol which is suitable for mobile system and is based on shared key mechanism is put forward. By analyzing the protocol, it can be seen that authentication from one end of tag to one end of reader is lacked, leading to that attackers are able to send messages to tag, pretending to be a reader so that counterfeit attack can be carried out.

In [12], based on cross bit operation, a mobile two-way authentication protocol is provided, in which bit operation is applied and information encryption is achieved so that whole computation of system can be greatly lowered. By analyzing the protocol, attackers can gain all messages

of a complete conversation by the means of eavesdropping. It is allowed to analyze messages gained previously, however, the protocol is not able to provide backward safety guarantee so that attackers can gain messages to analyze private messages in the previous round of conversation.

In [20], according to impossible clone physically, a mobile two-way authentication protocol is designed. The protocol is of high safety for the reason that shared key produced by Physical Unclonable Functions (PUF) cannot be copied. Although it is impossible to copy important messages in the protocol, information of shared key used between tag and database will lose its consistency after repeatedly replaying messages gained, which leads to the situation that the protocol cannot resist desynchronization attack from attackers.

In [18], a two-way authentication protocol is put forward which is suitable for mobile RFID system after considering many elements. Analyzing the protocol, although two-way authentication between mobile reader and server is increased, the one between tag and server is ignored, which offers attackers chances to communicate pretending to be tag or server so that the protocol cannot resist counterfeit attack from attackers.

In [**?**], a mobile two-way authentication protocol is presented by using bit operation to do encryption. In order to lower system computation, bit operation is used to do encryption instead of using Hash Function to encrypt. However, physical trespass is not considered during the design process of protocol, causing that attackers are able to gain shared key by the means of physical trespass. Thus, counterfeit attack is produced by using reversing clone technique so that the protocol cannot resist it.

In [15], based on Hash function, a mobile two-way authentication protocol is designed, however, there are many safety flaws in it, see the third part. Since the paper length is limited, no more two-way authentication protocols can be illustrated [4, 5, 8, 10, 16, 19].

In conclusion, recent two-way authentication protocols, more or less, embrace some safety flaws. In the paper, after analyzing safety flaws existed in [15], an improved protocol has been raised up. The improved protocol should endure recognition verification among communication entities first, followed by other subsequent operations. Otherwise, once the protocol stops, replay attack and counterfeit attack would be efficiently resisted.

The paper develops as the following: In the first part, drawbacks in traditional RFID system are illustrated and study problems which are suitable for mobile RFID system two-way authentication protocol will be put forward, some classic two-way authentication protocol in recent years have been introduced and disadvantages as well as flaws exist in them are indicated. In the second part, there is deep analysis of the two-way authentication protocol in [15] and safety problems in the protocol are pointed out. In the third part, includes an improved protocol based on analysis of disadvantages in [15] protocol and meanwhile, detailed steps are provided. In the fourth part, from the perspective of two-way authentication and replay attack, detailed safety analysis of the improved protocol is involved. In the fifth part, performance analysis of the improved protocol will be compared with other recent classic protocols. In the sixth part, concludes the whole paper and points out next step of study.

2 Analysis of Protocol

Main steps in this section: Certification process of protocol in [15] will be illustrated first, followed by detailed analysis of safety flaws in the protocol, and lastly an improved two-way authentication protocol will be put forward based on drawbacks existed in the protocol.

2.1 llustration of Authentication Process

Authentication process of protocol in [15] is as follows (For the meanings of symbols involved in the following illustration, please refer to [15], which will not be explained here). The specific process is shown in Figure 1.

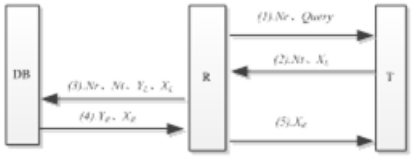


Figure 1: Protocol Diagram in [15]

1) Mobile reader sends $< Nr, Query >$ to tag, and two-way authentication will starts.

2) Tag makes calculation on X which is divided into data with left and right part, X_L and X_R. $< Nt, X_L >$ will be sent to mobile reader.

3) According to Nt, mobile reader will make sure whether it is necessary to filter information sent by tag. Y is calculated by mobile reader and is divided into data with left and right part, Y_L, Y_R. $< Nr, Nt, Y_L, X_L >$ will be sent to server.

4) Sever will make sure whether it is necessary to filter messages sent by mobile reader, according to Nr.

   The first step is to verify whether tag is legal or not, details are as bellow: A Y_L which is computed from the server will be compared to the Y_L sent by mobile reader. If they are the same, verification is valid and authenticity of tag will be soon verified; if not, verification on mobile reader will be failed and the protocol will end.

   The next step is to check legality of tag, details are as follow: X_L which is computed from the server making use of $< IDt_{new}, Kt >$ will be compared to the X_L which is sent by mobile reader. If they are the same, verification of tag will be valid and messages will be

updated; if not, $< IDt_{new}, Kt >$ will be replaced by $< IDt_{old}, Kt_{old} >$, calculating into a X'_L which will be compared to the X_L again sent by mobile reader. If they are the same, it means that verification of tag becomes valid and messages will be updated; if they are still different, verification of tag will be failed, followed by the end of protocol.

Server will send $< Y_R, X_R >$ to mobile reader after all above operations.

5) Mobile reader will compare the Y_R computed from itself to the Y_R sent by server. If they are the same, verification on server will be positive and X_R will be sent to tag; otherwise, verification will be failed and protocol will end.

6) Tag compares the X_R computed from itself to the X_R sent by mobile reader. If they are the same, verification on server and mobile reader will be valid, and messages will be updated; if not, protocol ends.

2.2 Specific Analysis for the Protocol

Analyzing the above protocol, it can be found that: The protocol cannot provide authentication from tag to mobile reader; meanwhile, it cannot resist counterfeit attacking from attackers.

Specific analysis why the protocol cannot provide authentication from tag to mobile reader is as follows: Combining Step 5 with Step 6 in above, server sends $< Y_R, X_R >$ message to mobile reader which will directly transfer X_R to tag without any other operations after finishing verifying Y_R. After receiving X_R, the tag verifies X_R with only authenticity of server rather than that of reader being tested for the reason that computing process of X_R is totally irrelevant to mobile reader. Therefore, the protocol cannot provide authentication from tag to mobile reader in the last step.

Specific analysis why the protocol cannot resist counterfeit attacking from attackers is as follows:
It can be concluded from the above analysis: The protocol cannot provide authentication from tag to mobile reader, therefore, attackers are able to acquire communication messages of Step 4 between mobile reader and server by the means of ear-dropping, which means attackers can listen for the message $< Y_R, X_R >$. After gaining the above messages, attackers will block communication of Step 5 between legal mobile reader and legal tag; meanwhile , pretending to be mobile reader, attackers send the gained Message X_R to tag. After tag receives the message, Step 6 will be operated and authentication of mobile reader which is pretended by attacker will be also positive. Thus, the protocol cannot resist counterfeit attacking from attackers.

Focusing on safety flaws in protocol in [15], an improved mobile two-way RFID authentication protocol that can resist counterfeit attacking from attackers will be provided. The improved protocol is mainly modified from the following aspects: First, all data used in the protocol is transferred as cipher-text, which means all messages cannot be transferred without encryption; second, messages from mobile reader to tag are transferred to tag after double encryption rather than are transferred in a simple way to resist counterfeit attacking from attackers and to provide authentication from tag to mobile reader; third, in order to decrease the whole computing amount of system, bit operation which has less computing amount is adopted to do encryption instead of applying Hash Function.

3 Improvement of Mobile RFID Authentication Protocol

3.1 Explanation of Protocol's Initialization and Symbols

In present mobile RFID system, both two communication entities server and mobile reader own strong computing capability, profound storage space and powerful inquiry capability; while tag does not have the above advantages, and it has weaker computing capability, is not able to operate complicated calculation, has a limited storage space and cannot store large data [7].

Before authentication protocol starts, there is initialization process with the purpose of finishing operations including secret key distribution and secure storage. Specifically, in the protocol designed in the paper, one end of server will produce the following data: $IDS_R, IDS_T, k, k_T, k_R$. Before the protocol starts working, assign $IDS_{T_{new}} = IDS_{T_{old}} = IDS_T$. Server will send $< IDS_R, k, k_R >$ to mobile reader through a safe path and store it safely in mobile reader; meanwhile, server will send $< IDS_T, k, k_T >$ to tag and store it safely in tag; while server will store Message $< IDS_R, IDS_T, k, k_T, k_R, k_{T_{old}}, IDS_{T_{old}} >$ in itself and at the same time, assign $k_{T_{old}} = 0$.

Meanings of symbols in protocol can be seen specifically in Table 1.

3.2 Permutation Cross Synthesis

Symbol $Pcs(X, Y)$ stands for Permutation Cross Synthesis. The specific definition of $Pcs(X, Y)$ is as follows:

1) X, Y, Z, and W are all binary sequence of length L which is even;

2) When the ith bit of the binary sequence X is 0, flip the value of the ith bit of binary sequence Y and put the value has been inverted on the ith bit of binary sequence Z; when the ith bit of the binary sequence X is 1, the ith bit of binary sequence Y will not be issued at all and put the value of this bit on the ith bit of binary sequence Z. According to (1) and (2) in above, permutation operation can be finished;

Table 1: Meaning of symbols in the protocol

Symbols	Description
Server	A database
Reader	A mobile reader
Tag	A tag
IDS_R	A pseudonyms for mobile reader
IDS_T	A pseudonyms for tag
k	A shared key between the three
k_T	A shared key between tag and server
$k_{T_{old}}$	A last shared key between tag and server
k_R	A shared key between server and mobile reader
$IDS_{T_{new}}$	The current pseudonyms of the deposit
$IDS_{T_{old}}$	The last authenticated pseudonym stored
rR	A random number generated by mobile reader
rT	A random number generated by tag
$\oplus$	XOR operation
$Pcs(X,Y)$	A permutation cross synthesis

3) Calculating Hamming weight and Hamming weight difference of binary sequence X and Y separately. When Hamming weight of X is greater than or equal to that of Y, cross synthesis should be operated under the circumstance of above Hamming weight difference of binary sequence Y and Z; oppositely, cross synthesis should be operated under the Hamming weight difference of binary X and Z;

4) Extract Hamming weight difference of binary sequence Y and put it on the right side of binary sequence W; extract back-end Hamming weight difference of binary sequence Z and put it on the left side of binary sequence W. Repeat the above operations so that a complete binary sequence W which is the operating result of $Pcs(X,Y)$ will be given out.

   $H(X)$ stands for Hamming weight of binary sequence X; $H(Y)$ stands for Hamming weight of binary sequence Y; and H(X-Y) stands for absolute value of Hamming weight difference between binary sequence X and Y, which is $H(X-Y)=|H(X)-H(Y)|$.

3.3 Illustration of Protocol

The improved protocol flow chart can be referred to Figure 2.

Combined with the above flow chart, specific steps of the improved protocol can be illustrated as follows:

Step 1. Combining with the right side k_R of stored shared key k and the random number rR which is produced by mobile reader, $m1$ will be produced by calculation; combining with the left side k_L of stored shared key k and the random number rR, result $m2$ will be produced. Then send the above results together with $\{m1, m2, Query\}$ to tag and start verifying the protocol. Query stand for two-way authentication query. $m1$ and $m2$ are calculated out as the following rules: $m1 = rR \oplus k_R$, $m2 = rR \oplus k_L$.

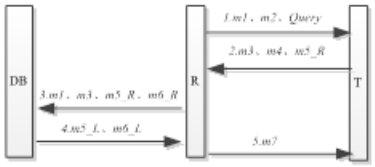


Figure 2: The improved protocol flow chart

Step 2. After receiving the message $\{m1, m2, Query\}$, the tag will verify the mobile reader first. Only if the authentication is valid, the following operations can be continued, otherwise, the protocol will be stopped. The tag calculates $m1 \oplus k_R$ by making use of received $m1$ and the right side k_R of stored shared key k; calculates $m2 \oplus k_L$ by making use of received $m2$ and the left side k_L of stored shared key k, followed by comparison between value of $m1 \oplus k_R$ and that of $m2 \oplus k_L$ to see whether they are the same.

If they are different, authentication on mobile reader from tag would be negative, which leads to the end of protocol; otherwise, the authentication would be valid, followed with random number rR from reader by calculating. Tag produces a random number rT together with calculated random number rR which calculate into $m3$; using random number rT and reserved share key k to calculate into $m4$; combining rT, k_T, and IDS_T, $m5$ can be calculated out. Then $m5$ is divided into two parts $m5\_R$ and $m5\_L$ with same left and right bit. Next is to send $\{m3, m4, m5\_R\}$ to mobile reader as the response from tag to mobile reader.

rR, m3, m4 and m5 are calculated out from the following rules: $rR = m1 \oplus k_R$ (or $rR = m2 \oplus k_L$), $m3 = rR \oplus rT$, $m4 = Pcs(rT, k)$ and $m5 = Pcs(rT \oplus k_T, IDS_T)$.

Step 3. After mobile reader receives message $\{m3, m4, m5\_R\}$, tag will be verified first. Only if the authentication passes, operations left can be continued, otherwise, protocol quits.

Making use of received $m3$ and its own random number rR, the mobile reader can get a random number rT‘; making use of the random number rT‘ and reserved shared key k, m4‘ can be calculated out, followed by comparison between $m4$‘ and $m4$.

If they are different, authentication on tag is invalid and the protocol quits. Otherwise, authentication on tag passes and rT‘ $= rT$ and $m4$‘ $= m4$; then the reader calculates its own random number rR, reserved shared key k_R and reserved pseudonym IDS_R

into $m6$ which is separated into two parts $m6\_R$ and $m6\_L$ whose left and right bit are the same. In the end, $\{m1, m3, m5\_R$,m6_R$\}$ will be sent to server together.

rT‘, $m4$‘, and $m6$ are calculated out from the following rules: rT‘ =m3$\oplus rR$, $m4$‘ $= Pcs(rT$‘$, k)$, and $m6 = Pcs(rR, k_R \oplus IDS_R)$.

Step 4. After server receives message, the mobile reader will be verified first and the tag will be authenticated only if the verification on reader passes. Left operations will be continued only if the authentication both on reader and tag passes; otherwise, the protocol quits.

1) Server verifies mobile reader:
Making use of received $m1$ and k_R which is right part of reserved shared key k , server calculates out a random number rR‘; making use of the random number rR‘, reserved shared key k_R and reserved pseudonym IDS_R, $m6$‘ will be calculated out, followed by comparison between received $m6\_R$ and $m6\_R$ which is right part of $m6$‘.
If they are different, authentication on mobile reader would be invalid so the protocol quits. If they are the same, verification of mobile reader passes and rR‘ $= rR$ and $m6$‘ $= m6$. Then server starts verifying authenticity of tag.
rR‘ and $m6$‘ are calculated out obeying the following rules: rR‘ =m1$\oplus k_R$ and $m6$‘ $= Pcs(rR$‘$, k_R \oplus IDS_R)$.
2) Server verifies tag:
Making use of the calculated random number rR and received $m3$, server calculates out a random number rT‘; making use of the random number rT‘, reserved shared key k_T and reserved pseudonym IDS_T, $m5$‘ will be calculated out. Then compare received $m5_R$ with $m5_R$‘ which is right part of $m5$‘ to see whether they are the same.
The same, authentication on tag passes and rT‘ $= rT$ and $m5$‘ $= m5$, following updating messages: $IDS_{T_old} = IDS_{T\_new}$, $k_{T\_old} = k_T$, $k_T = Pcs(rT, k_T)$ and $IDS_{T_new} = Pcs(rT, IDS_{T_old})$.
If different, server will replace k_T, IDS_T with k_{T_old}, IDS_{T_old} which is used in the last authentication, getting the result $m5$“. Compare the value of calculated result $m5\_R$“ with received $m5\_R$ by making use of $m5\_R$“ which is right part of $m5$“. If they are still different, the authentication on tag fails, and the protocol quits; otherwise, authentication on tag passes, and tag becomes synchronous with server again. Meanwhile, information is updating: $k_T = Pcs(rT, k_T)$ and $IDS_{T\_new} = Pcs(rT, IDS_{T\_old})$.
After updating information, server will take out $m6\_L$ which is left part of calculated result $m6$ in (1) and $m5\_L$ which is left part of calculated result $m5$ in (2), and sends $\{m5\_L$,m6_L$\}$ to mobile reader.
rT‘, $m5$‘, $m5$“ are calculated out obeying the following rules: rT‘ =m3$\oplus rR$, $m5$‘ $= Pcs(rT$‘ $\oplus k_T, IDS_T)$ and $m5$“ $= Pcs(rT$‘ $\oplus k_{T\_old}, IDS_{T\_old})$.

Step 5. After receiving the message $\{m5\_L$,m6_L$\}$, mobile reader will verify server first. Only the authentication passes, the left operation can be continued; otherwise, the protocol quits.

Mobile reader will compare received $m6\_L$ with its own calculated result $m6\_L$. Different, authentication on server fails, and the protocol quits. The same, the authentication passes. Mobile reader calculates out $m7$ by integrating received $m5\_L$, its own random number rR and calculated random number rT. Lastly, $\{m7\}$ will be sent to tag.

$m7$ is calculated out obeying the following rule: $m7 = m5_L \oplus$rR$\oplus rT$.

Step 6. After receiving the message $\{m7\}$, the tag will verify mobile reader and server first. When it passes, left operations will be continued; otherwise, the protocol quits.

Making use of its own calculated result $m5\_L$, its own random number rT and calculated random number rR, the tag calculates out the result $m7$‘, followed by comparison between received $m7$ and calculated result $m7$. Different, authentication on server and mobile reader fails, which means at least, server or mobile reader is forged, leading to the end of the protocol. The same, the authentication passes, followed by message updating: $k_T = Pcs(rT, k_T)$ and $IDS_{T\_new} = Pcs(rT, IDS_{T\_old})$. After updating messages, two-way authentication among tag, mobile reader and server these three communication entities will be finished. $m7$‘ is calculated out as the following rule: $m7$‘ $= m5_L \oplus$rR$\oplus rT$.

4 Safety Analysis of Improved Protocol

1) Two-way Authentication: The basic function of protocol is able to provide authentication among communication entities. There are three communication entities in the protocol in the paper, and mutual authentication among them should be provided. The protocol is able to provide this function. Two-way authentication between tag and mobile reader.

In the protocol, there are two parts included when tag is verifying mobile reader. First, in Step 2, tag undertakes the first authentication on mobile reader

using received $m1$ and $m2$. The detailed methods have been illustrated in Step 2; second, in Step 6, tag undertakes the second authentication on mobile reader using received $m7$, which can be found in Step 6. Authentication on tag from mobile reader is finished in Step 3 through received $m3$ and $m4$, which can be seen in Step 3.

Two-way authentication between tag and server.

In the protocol, tag's authentication on server is finished in Step 6 through received $m7$. While server's authentication on tag is finished in Step 4 through received $m3$ and $m5\_R$, which has been illustrated in Step 4.

Two-way authentication between mobile reader and server.

Authentication on server from mobile reader is finished in Step 5 through received $m6\_L$, which has been explained in detail in Step 5. While server's authentication on mobile reader is finished in Step 4 through received $m1$ and $m6\_R$, which has been specifically illustrated in Step 4.

How tag finished verifying both mobile reader and server through $m7$ at the same time has been emphasized in the last step. $m7$ involves the message $m5\_L$ which is calculated out by server; if server is forged, the message $m5\_L$ cannot be calculated out correctly. Therefore, tag can identify the authenticity by comparing calculated result $m7$ and received $m7$. Meanwhile, $m7$ involves messages rR and rT which are from mobile reader itself and its calculation; if mobile reader is forged, messages rR and rT cannot be figured out so that calculated result $m7$ is also wrong. Therefore, based on the above illustration, as long as calculated result $m7$ from one end of tag is different from received $m7$, one of mobile reader and database is forged, at least.

2) Replay Attack: Random numbers have been added in the process of encryption on all messages in the protocol so that it can be assured that communication messages of two separated times are different; with random numbers added, it can also be assured that communication messages are real-time. Even though attackers can steal one round communication messages by ear-dropping and they are ready to acquire private messages of communication entities by means of replaying, they cannot achieve success. Messages attackers acquire from replaying and ear-dropping can pass authentication from one of communication entities, however, when communication entities text back to attackers, attackers cannot decode private messages for the reason that they are lacking of correspondent parameter information. Therefore, replay attack from attackers fails and the protocol can resist attackers' replay attack.

3) De-synchronization Attack: De-synchronization attack refers to the situation that shared key used in authentication between tag and server loses its consistency. Specifically, one of tag and server updates the key while the other does not, which leads to the inconsistency of the shared key used in the second authentication. In order to resist de-synchronization attacks from attackers, in the protocol designed in the paper, not only authentication shared key k_T and tag pseudonym IDS_{T_new} of the present communication round but also authentication shared key k_{T_old} and tag pseudonym IDS_{T_new} of the last authentication round are reserved in one end of server. When it is unsuccessful for server to identify the authenticity of tag, $< k_T, IDS_{T\_new} >$ will be replaced with $< k_{T\_old}, IDS_{T\_old} >$ to verify the authenticity of tag. Tag is estimated to be forged if and only if the above two authentication fail. Therefore, the protocol can resist the de-synchronization attack from attackers.

4) Brute Force Attack: Attackers would like to use exhaustive method to squeeze all private information instead of any other attack methods for the reason that the present computer has a powerful computing ability so that the protocol can resist brute force attack from attackers. During design process of the protocol, all messages are sent out after encryption in which random numbers are added so that attackers are not able to decode any useful private messages with brute force attack. For instance, in $m1 = rR \oplus k_R$, attackers can acquire the message $m1$ by ear-dropping. However, $m1$ is private text and attackers can acquire private messages only after decoding it. Attackers cannot squeeze all private messages because in $m1$, messages rR and k_R which are not approachable to attackers; besides, message random number rR is produced randomly which cannot be known in advance, still, message $m1$ of two times are different. Based on the above illustration, the protocol is able to resist brute force attack from attackers.

5) Counterfeit Attack: Counterfeit attack means that attackers acquire communication messages by using some other methods, followed by using the method of replaying to muddle through authentication. The point whether the protocol can resist counterfeit attack or not is whether the protocol can provide two-way authentication among communication entities. Based on the specific analysis in the part (1) that the improved protocol can provide two-way authentication among communication entities, even though attackers acquire some communication messages by ear-dropping and by replaying, authentication of one of three communication entities passes, getting response from communication entities, they are not able to analyze out any useful private messages from response because they are lacking of necessary parameter information. Therefore, the protocol is able to resist attackers' counterfeit attack.

Table 2: Safety comparison of protocols

Attack type	[14]	[12]	[20]	[18]	[13]	[15]	Our protocol
Two-way authentication	V	V	V	V	V	×	V
Replay attack	×	V	V	×	×	×	V
De-synchronization attack	V	V	V	V	V	V	V
Brute force attack	V	V	×	V	V	V	V
Counterfeit attack	V	V	V	V	V	V	V
Backward security	V	×	V	V	V	V	V

Notes: V means can resist, × means cannot resist.

6) Backward Security: Backward security refers to the situation that attackers can acquire some present communication messages by specific method from which they can infer next time's communication messages so that private messages can be analyzed out. During design process of the protocol, all communication messages are sent after encrypting to ensure communication safe in which random numbers are added. For instance, in $m1 = rR \oplus k_R, rR$ and k_R are unknown and even though attackers acquire the message $m1$ of the present round by ear-dropping to infer the one of next round, they cannot succeed for the reason that with random number rR added, attackers cannot infer the following points. First, random numbers are produced randomly in each round, which is unpredictable; second, random numbers are different when they are produced. Based on the above two points, attackers cannot predict specific value of random numbers in encryption process of next round so that they cannot infer the message $m1$ of next round. Therefore, the protocol embraces forward safety.

Table 2 is safety comparison between the protocol in this paper and other protocols.

5 Performance Analysis of the Improved Protocol

According to the above illustration: there are three communication entities in RFID System, however, tag does not have the advantages including strong computing ability and enough reservation space which both mobile reader and server have. Therefore, tag is chosen to be the object of performance analysis for performance analysis section, and from the perspective of computing amount and reservation amount of one end of tag, functions of each protocol is deeply analyzed, whose specific analysis can be seen in Table 3.

Explanation of symbols in Table 3 is as follows: H refers to computing amount of Hash Function; PR refers to random computing amount; Mm refers to computing amount of modular multiplication algorithm; PUF refers to computing amount of Physical Unclonable Function (PUF); P refers to computing amount of random number function; M refers to computing amount of modular algorithm; C refers to computing amount of crossover operation; XOR refers to computing amount of Exclusive OR Operation; Pcs refers to computing amount of permutation cross synthesis. l is the symbol of all communication messages length.

Table 3: Performance comparison of protocols

Index	Amount of calculation	Storage capacity
[14]	3PR+H	2l
[12]	8C+14XOR	3l
[20]	PR+7H+2PUF	2l
[18]	5P+3M	5l
[13]	P+C	3l
[15]	3P+2M+2PUF	4l
Our protocol	5XOR+4Pcs	3l

The above operations could be divided into two types, one is with lightweight class computing amount, the other is with super lightweight class computing amount. H, PR, Mm, PUF, P, and M belong to lightweight class computing; while C, XOR and Pcs all belong to super lightweight class computing. According to different operation definitions of different weight classes, computing amount of lightweight class is usually times of that of super lightweight class, which means computing amount of lightweight class is several times of that of super lightweight class. It can be found in performance analysis in Table 3 that apart from the protocol in the paper, other protocols are all encrypted with lightweight class operation. Therefore, with regard to computing amount of one end of tag, the protocol in the paper stands out. From the perspective of reservation space, compared to other protocols, the protocol in this paper has an advantage in reserving data amount. Based on performance and safety analysis, improvement and advance have been made to computing amount of one of tag, which can also make up safety problems in other protocols.

6 Conclusion

When traditional RFID system is applied conversely, new problems appear. Mobile RFID system can make up disadvantages of traditional one. However, mobile RFID system is unable to ensure communication message safe, therefore, in order to make sure the safety of communication messages, two-way authentication protocol has been designed for mobile RFID system. After analyzing flaws of the protocol put forward by Wang and others, the improved protocol is provided. The improved protocol only can continue left operations after finishing two-way authentication among communication entities so that it can avoid replay attack and counterfeit attack from attackers; meanwhile, to resist other types of attack, communication messages are sent after encryption in which random numbers are added so that attackers cannot infer communication messages of last round and next round through messages ear-dropped in the present. Undertaking safety analysis of the protocol means the protocol can satisfy the safety need of mobile RFID system; undertaking performance analysis means the protocol is suitable to present mobile RFID system with regard to computing amount. The next research orientation of the paper: it is available to make prototype of mobile RFID system with the protocol come true and study some specific parameters such as time as well as computing amount of a complete communication process.

Acknowledgments

This paper is supported by the Science and Technology Planning Project of Guangdong Province (China) (2019B020208001), the Guangdong Science and Technology Collaborative Innovation Center for Judicial Administration (China) (2019B110210002), the Natural Science Project of Guangxi Universities under Grant (2021KY0051).

References

[1] Y. P. Duan, "Lightweight RFID group tag generation protocol," *Control Engineering of China*, vol. 27, no. 4, pp. 751–757, 2020.

[2] K. Fan, W. Jiang, H. Li, *et al.*, "Lightweight RFID protocol for medical privacy protection in IoT," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1656–1665, 2018.

[3] P. Gope, J. Lee, T. Q. S. Quek, "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831–2843, 2018.

[4] T. Li, Y. L. Liu, "A new double PUF-based RFID authentication protocol," *Journal of Computer Research and Development*, Mar. 4, 2021. (`https://kns.cnki.net/kcms/detail/11.1777.tp.20210303.1726.006.html`)

[5] W. Liang, S. Xie, J. Long, *et al.*, "A double PUF-based RFID identity authentication protocol in service-centric Internet of Things environments," *Information Sciences*, vol. 50, no. 3, pp. 129–147, 2019.

[6] D. W. Liu, J. Ling, "An improved RFID authentication protocol with backward privacy," *Computer Science*, vol. 43, no. 8, pp. 128-130, 2016.

[7] T. Liu, L. F. Jia, P. Guo, "Research on an RFID security authentication protocol against quantum attack," *Control Engineering of China*, vol. 27, no. 3, pp. 513–519, 2020.

[8] Y. L. Liu, M. F. Ezerman, H. X. Wang, "Double verification protocol via secret sharing for low-cost RFID tags," *Future Generation Computer Systems*, vol. 90, pp. 118–128, 2019.

[9] Y. L. Liu, X. C. Yin, Y. Q. Dong, *et al.*, "Lightweight authentication scheme with inverse operation on passive RFID tags," *Journal of the Chinese Institute of Engineers*, vol. 42, no. 1, pp. 74-79, 2019.

[10] Z. H. Liu, C. J. Huang, H. Suo, "Modified mobile RFID bidirectional authentication protocol against counterfeiting attack," *Computer Applications and Software*, vol. 37, no. 6, pp. 309–315, 2020.

[11] S. Q. Mei, X. R. Deng, "Mobile RFID bidirectional authentication protocol based on shared private key and bitwise operation," *Computer Applications and Software*, vol. 37, no. 7, pp. 302-308, 2020.

[12] F. Rahman, M. E. Hoque, S. I. Ahamed, "Anonpri: A secure anonymous private authentication protocol for RFID systems," *Information Sciences*, vol. 379, no. 2, pp. 195–210, 2017.

[13] S. Sundaresan, R. Doss, S. Piramuthu, *et al.*, "A secure search protocol for low cost passive RFID tags," *Computer Networks*, vol. 122, pp. 70–82, 2017.

[14] F. Tan, "An improved RFID mutual authentication security hardening protocol," *Control Engineering of China*, vol. 26, no. 4, pp. 783–789, 2019.

[15] J. Wang, X. M. Wang, "Improved lightweight mutual authentication protocol for mobile RFID," *Computer Engineering and Design*, vol. 39, no. 4, pp. 912-917, 2018.

[16] J. Q. Wang, Y. F. Zhang, D. W. Liu, "Provable secure for the ultra-lightweight RFID tag ownership transfer protocol in the context of IoT commerce," *International Journal of Network Security*, vol. 22, no. 1, pp. 12–23, 2020.

[17] P. Wang, Z. P. Zhou, J. Li, "Improved server-less RFID security authentication protocol," *Journal of Frontiers of Computer Science and Technology*, vol. 12, no. 7, pp. 1117-1125, 2018.

[18] C. H. Wei, M. S. Hwang, A. Y. H. Chin, "Security analysis of an enhanced mobile agent device for RFID privacy protection," *IETE Technical Review*, vol. 32, no. 3, pp. 183–187, 2016.

[19] R. Xie, B. Y. Jian, D. W. Liu, "An improved ownership transfer for RFID protocol," *International Journal of Network Security*, vol. 20, no. 1, pp. 149–156, 2018.
[20] R. Xie, R. Ling, D. W. Liu, "Wireless key generation algorithm for RFID system based on bit operation," *International Journal of Network Security*, vol. 20, no. 5, pp. 938–949, 2018.

Biography

Shan-Hua Zhan received his doctorate from the school of computer science, Guangdong University of technology in December 2019. He is now a lecturer, working in Guangdong judicial police vocational college. At present, his research interests mainly include information security, data mining, etc.

Chun-Qiang Yu (Member, IEEE) is with the Guangxi Key Laboratory of Multi-Source Information Mining & Security, Guangxi Normal University. His research interests include Multimedia information security and Data hiding,etc.

Study on Network Video Image Encryption Based on an Optimized Algorithm Combined with High-Efficiency Video Coding

Wenwen Li[1], Hongfei Xiao[2], Shiqi Tang[3]

(Corresponding author: Hongfei Xiao)

School of Media and Design, Chuzhou Polytechnic, Chuzhou, Anhui 239000, China[1]

School of Information Engineering, Chuzhou Polytechnic[2]

Chuzhou, Anhui 239000, China

Email: hongfxiao@126.com

Information Center of Ministry of Science and Technology, Beijing 100862, China[3]

(Received Sept. 19, 2020; Revised and Accepted Dec. 11, 2021; First Online Feb. 28, 2022)

Abstract

High-definition video needs to be encoded and compressed for transmission on the Internet, and transmission security is ensured. This paper briefly introduces High-Efficiency Video Coding (HEVC) technology and the video encryption algorithm based on the entropy coding of HEVC improved the encryption algorithm by the codeword sensitivity in entropy coding and conducted simulation experiments. The results showed that the video encrypted by the encryption algorithm improved by the codeword sensitivity of entropy coding had a lower peak signal-to-noise ratio, showing better encryption effect; the increase of video resolution led to the increase of encoding encryption time, but the improved encryption algorithm took less time for encoding encryption.

Keywords: Codeword Sensitivity; Entropy Coding; High Efficiency Video Coding; Video Encryption

1 Introduction

The development of communication technologies has enabled the expansion of the Internet and accelerated the speed of information delivery. The emergence of multimedia technologies has increased the variety of ways users can view information on the Internet [1]. Internet video is one of the multimedia technologies on the Internet, and it is one of the multimedia commonly used by users. The popularity of smart mobile terminals allows users to watch videos at any time, and users' pursuit of video clarity is getting higher and highert [2]. The increase in the clarity of network video will lead to a large increase in the amount of data, increasing the pressure on the storage and transmission of the Internet; hence, it is necessary to encode and compress high-definition video to reduce the amount of storage and improve the transmission speedt [8].

Videos contain a large amount of information, including much important private information; thus, they need to be encrypted in the transmission process. Xu *et al.* [9] designed an efficient chaotic pseudo-random number generator for generating a key stream for encrypting H.264/AVC video syntax elements. The experimental results showed that the encryption process did not affect the coding efficiency of H.264/AVC and the design was resistant to malicious attacks. Saini *et al.* [10] proposed a chaotic mapping-based real-time optical video encryption technique and verified through numerical simulations that the method was effective. Jeyamala *et al.* [3] improved the speed of encryption by chaos-based S-box design and found through experiments that chaos-based S-box design and key generation significantly reduced the computational cost of video encryption without compromising security. This paper briefly introduces High Efficiency Video Coding (HEVC) technology and the entropy coding-based video encryption algorithm. The encryption algorithm was improved by using the codeword sensitivity in entropy coding. Simulation experiments were conducted.

2 HEVC and Video Encryption Algorithm

2.1 HEVC Technology

The basic process framework for encoding a video with HEVC technology used is shown in Figure 1. One frame in the video is taken as an example, and the steps are as follows.

1) The input video screen is divided equally into large blocks. Every block is a coding tree unit (CTU).

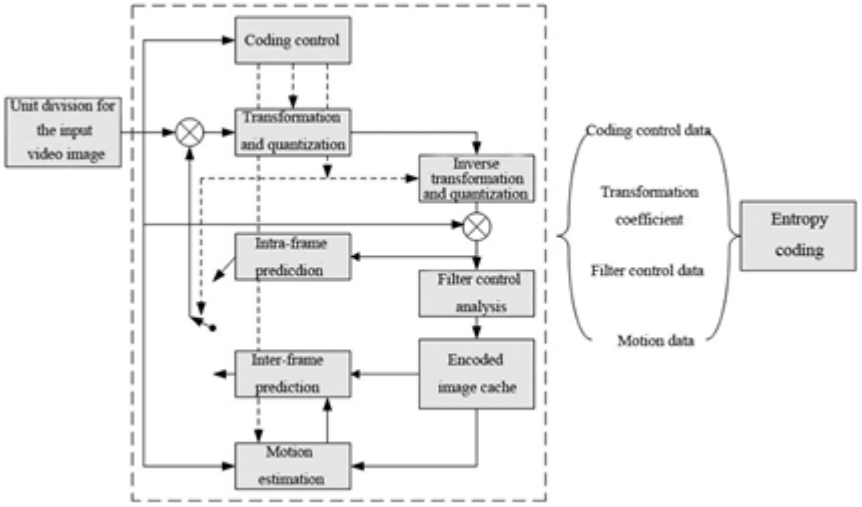


Figure 1: Basic flow of HEVC framework

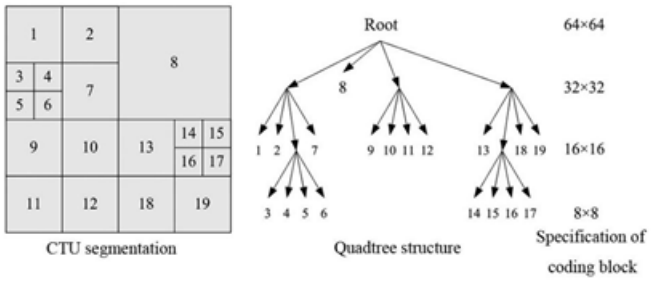


Figure 2: Schematic diagram of CTU segmentation

Every CTU is further divided according to the structure of a quadtree [7] to obtain coding units (CU), as shown in Figure 2. The maximum size of the CU is 64×64, the minimum size is 8×8. The more complex the image texture is, the smaller the CU is. When scanning and encoding the CU, a zigzag scan is used, and Figure 2 shows the encoding result under this scan method. In addition to CU, the structure of frame segmentation also includes prediction unit (PU) and transform unit (TU). Both PU and TU are obtained by subdividing CU. The role of PU is to predict coding. CU can be divided into intra-frame and inter-frame PU, but only one of the division modes can be selected in the same CU. The role of TU is to transform and quantize the pixel data in the unit.

2) Intra-frame or inter-frame prediction is performed on the PU obtained from the previous division step according to the demand. The inter-frame prediction requires the motion estimation information in the front and rear frames in the cache of the coded image and their code division units [5].

3) The predicted signal is compared with the original video signal, and the residual signal is transformed and quantized in the TU obtained from the coding division. After that, the residual signals form the reconstructed values together with the intra- or inter-frame prediction signals after inverse transformation and quantization at the decoding end. The reconstructed values are filtered to obtain the coded image cache. These caches are then used for inter-frame prediction and motion estimation.

4) Entropy coding is performed on the syntactic elements such as coding control data (how to divide the image), transformation coefficients (for transforming and quantizing the residual signal), filter control data, and motion data obtained in the processing of the above steps [13].

Context-based adaptive binary arithmetic coding (CABAC) is used for entropy coding in HEVC technology, whose framework is divided into a binarization module, a context modeling module, and a binary arithmetic module. The binary module is to convert the non-binary elements in the input syntax elements into binary elements; the context modeling module designs a context model for the input syntax elements to ensure that the model can efficiently represent the probability distribution of the sequence of syntax elements and can be updated with the coding process; the binary arithmetic module inputs the binary syntactic elements sequentially into the context model for encoding in normal mode and encodes the input stream in bypass encoding mode [6].

2.2 A Video Encryption Algorithm Based on The Entropy Coding of HEVC

Encryption is performed in the binarization module. The encryption process is as follows.

1) Syntactic elements such as coding control data, transformation coefficients, filter control data and motion data obtained when the video is encoded by HEVC are input.

2) Whether the input syntactic elements are transformation and quantization residuals belonging to transformation coefficients, motion vector difference belonging to motion data, or coding control coefficients and filter control coefficients belongings to other kinds is determined.

3) When the syntactic element is the motion vector difference, it is encoded by exponential Golomb coding [14]. The form of exponential Golomb coding is $[M][1][INFO]$, where M is prefix M composed of a string of 0 (the number of 0 is M), 1 is the intermediate number, $INFO$ is the suffix, whose encoding length determines the length of the prefix. The calculation formulas of M and $INFO$ are:

$$\begin{aligned} M &= floor(\log_2(CodeNum+1)) \\ INFO &= CodeNum - 2^M + 1 \end{aligned} \quad (1)$$

$$CodeNum = \begin{cases} V & V \text{ is an unsigned number} \\ 2V-1 & V \text{ is a signed number and } V>0 \\ -2V & V \text{ is a signed number and } V \le 0 \end{cases}$$

where $floor(\cdot)$ is the integer function, $CodeNum$ is the index number of the Columbian code, and V is the data to be encoded, which is divided into signed and unsigned numbers. In order not to destroy the format information of the video, only the sign of the lowest bit in $INFO$ is processed by 0-1 scrambling, *i.e.*, only the sign of the motion vector difference is changed, and the numerical magnitude remains unchanged. Then, the $[M][1][INFO]$ of the scrambled motion vector difference is encoded in the arithmetic encoder [15].

4) When the syntactic element is the transform and quantization residual, the data are encoded by exponentially Columbian coding using Equation (1), the encoded sequence in $INFO$ is encrypted using the key of the AES encryption algorithm, and the encrypted $[M][1][INFO]$ is encoded in the arithmetic encoder.

5) When the syntactic element is of other kinds, the element is encoded using fixed-length encoding, i.e., the value of the syntactic element is expressed using a fixed-length codeword. Then, it is encoded in the arithmetic encoder.

2.3 Improving HEVC-based Video Encryption Algorithm Based on Codeword Sensitivity

When encoding a video with HEVC, a frame is first divided into multiple CTUs in a size of 64×64, the CTUs are divided into CUs of different sizes according to the quadtree structure, the CTUs in the same frame are scanned in the order from left to right and top to bottom, and the CUs in the same CTU are encoded in a zigzag pattern. The same order is followed when encrypting the entropy code. The full encryption of the data in the entropy code is not only inefficient but also lacks specificity. Therefore, in order to improve the encryption efficiency and the targeting of encryption, this paper used the codeword sensitivity [4] in entropy encoding to perform targeted encryption.

The codeword sensitivity in entropy coding refers to the importance of the codeword to the reconstructed video; the higher the sensitivity is, the greater the importance is. Once there is an error, the quality of the video reconstructed according to the codeword will be poor. Table 1 shows the six codewords that can be used for encryption at the CU level during entropy coding, among which 1) 4) codewords work when entropy coding is performed on I frame and P frame [12] and 5) and 6) codewords only work when entropy coding is performed on P frame.

Before using the sensitivity of codewords for selective encryption of entropy code, it is first necessary to rank the sensitivity of the six codewords in I frame and P frame. After obtaining the sensitivity ranking, the entropy encoding of the syntactic elements of CU prioritizes to encrypt the codewords with high sensitivity, and the codewords with low sensitivity are not encrypted for the time being, achieving targeted encryption and reducing the encryption computation. The encryption process of the improved entropy coding is shown in Figure 3.

1) Whether the input CT belongs to I frame or P frame is determined, and then the sensitivity ranking of the encoded codewords within the CT is selected according to the frame type they belong to.

2) The corresponding encoding method is selected according to the type of syntactic elements within CT. The syntactic elements related to transform and quantization and motion vectors are encoded by exponential Columbus encoding, and the rest types of syntactic elements are encoded by fixed-length codes.

3) The codeword with the highest sensitivity is selected from the syntactic elements that have been encoded as Columbian code or fixed-length code [11] according to the sensitivity ranking of the codeword within the CT of the corresponding frame and encrypted. The encryption method is the same as the encryption process in the section of a video encryption algorithm based on entropy coding.

Table 1: Six codewords available for encryption at the CU level during entropy coding

Number	Code Type	Codeword	Scope of Action
1	Transformation coefficient symbol	coef_sign_flag	I frame, P frame
2	Quantization parameter sign	cu_qp_delta_sign	I frame, P frame
3	Quantization parameter value	suffix of cu_qp_delta_abs	I frame, P frame
4	Intra-frame prediction mode	rem_intra_luma_pre_mode	I frame, P frame
5	Sign of motion vector difference	mvd_sign_flag	P frame
6	Value of motion vector difference	suffix of abs_mvd_minus2	P frame

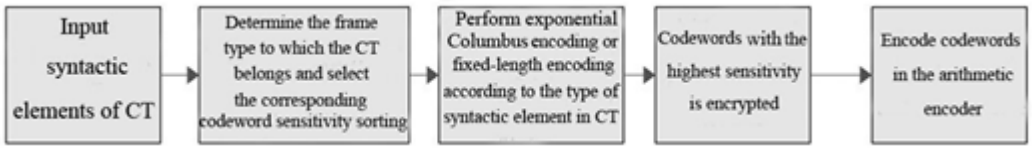


Figure 3: Encryption flow of entropy coding improved by codeword sensitivity

4) The encrypted and unencrypted codewords in Step 3 are encoded in the arithmetic encoder, and the encrypted entropy code is output.

3 Simulation Experiments

3.1 Experimental Environment

Experiments were conducted in a lab server with configurations of 16G memory, Core i7 processor, and Windows 7 operating system. The software used in the experiments was Vista Studio 2010. The video was encoded by HEVC HM10.0.

3.2 Experimental Data

Videos in a resolution of 416×240, 832×480, 1280×720, 1920×1080 and 2560×1600 were collected from the Internet. The basic parameters for encoding videos with HEVC HM10.0 are as follows: the maximum size of the encoding unit was 64×64, the segmentation depth of the quadratic tree was 3, the frame rate was set as 30, the size of group of pictures (GOP) in the video was 8, the encoding frame was 10, and the quantization coefficient was 32.

3.3 Experimental Projects

Code Sensitivity Test:
When using HEVC HM10.0 for entropy coding of videos in different resolutions, the six CU codewords in I and P frames of every video were scrambled separately, so there were 12 scrambling schemes for every video, and only one CU codeword in one frame was scrambled in every scheme. The peak signal-to-noise ratios (PSNRs) of the original video before scrambling and the video after scrambling were calculated, and the sensitivities of the six CU codewords in I and P frames were ranked.

Video Encryption Performance Test Based on Entropy Coding of HEVC:
In order to verify the performance of the encryption algorithm improved by the codeword sensitivity of entropy coding, it was compared with the non-improved encryption algorithm. The video was encrypted using the improved and non-improved encryption algorithms. The process of the non-improved encryption algorithm was as described above, while the improved encryption algorithm encrypted the video following the sensitivity ranking obtained from the previous test project, i.e., encrypting the codewords with high sensitivity in priority.

The algorithm performance was evaluated by PSNR. Firstly, the PSNR of the video before and after encryption was evaluated; then, the PSNR of the video before encryption and after decryption was evaluated. In addition, the average time consumption of the two encryption algorithms was tested.

3.4 Experimental Results

In order to test the sensitivity of the six types of codewords in I and P frames in the video, they were individually scrambled. The PSNR before and after the scrambling was calculated, and the results are shown in Figure 4. It was seen from Figure 4 that the PSNRs of No. 5 and 6 in the I frame, which were the sign and value of motion vector difference, were nearly unchanged after the scrambling, which meant that these two codewords

had the lowest sensitivity in the I frame; No. 1 and 4, which were the sign of transformation coefficients and the intra-frame prediction pattern, had the highest PSNR reduction after the scrambling, which meant that they had the highest sensitivity in the I frame; No. 2, which was the sign of quantization parameters, had the second-highest PSNR reduction after the scrambling. No. 3, i.e., the value of quantization parameters, had the third-highest PSNR reduction after the scrambling. In the P frame, the PSNR was nearly unchanged when 3 and 6 were scrambled; the PSNRs reduced the most when No. 1, 2 and 4 were scrambled; the PSNR reduced the second most when No. 5 was scrambled.

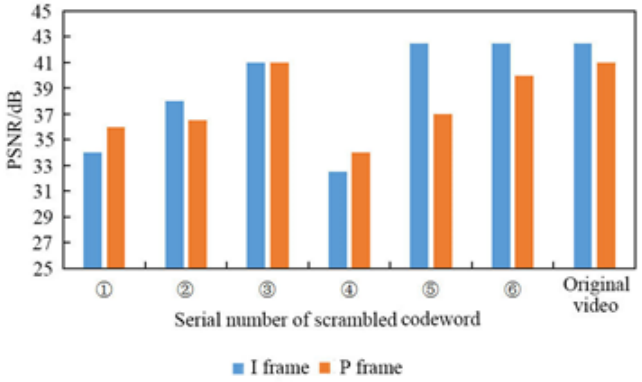


Figure 4: PSNRs after scrambling of different codewords in I and P frames in the video

Therefore, the encryption priorities in I and P frames were divided into three levels according to the sensitivities of the six codewords. In the I frame, the codewords of the first level encryption were the sign of transformation coefficients and the intra-frame prediction pattern, the codeword of the second level encryption was the sign of quantization parameters, and the codeword of the third level encryption was the value of quantization parameters. In the P frame, the codewords of the first level encryption were the sign of transformation coefficients and the intra-frame prediction pattern, the codewords of the second level encryption were the sign of quantization parameters and the sign of motion vector differences, and the codewords of the third level encryption were the value of quantization parameters and the value of motion vector differences.

Figure 5 shows some results of the video encrypted by the two encryption algorithms. It was seen from Figure 5 that the image processed with the non-improved encryption algorithm was blurred compared to the original image, but the general outline was visible, but the content of the image processed with the improved encryption algorithm was almost impossible to be seen. Figure 6 shows the PSNR of video images in different resolutions after encryption by the two encryption algorithms. It was seen from Figure 6 that the PSNR of videos in different resolutions encrypted by the same encryption algorithm was nearly the same, which indicated that both encryption algorithms were relatively stable. The comparison of the PSNR of the video in the same resolution after encryption by different encryption algorithms demonstrated that the PSNR of the video encrypted by the improved encryption algorithm was significantly lower regardless of the resolution, which meant that the encryption performance of the improved encryption algorithm was better.

Figure 7 shows the time consumed by the two encryption algorithms for encrypting videos in different resolutions. It was seen from Figure 7 that the time consumed by the two encryption algorithms increased with the increase of the video resolution. Higher resolution meant more pixels in every frame, i.e., the number of CTs to be processed was larger, leading to increased calculation amount. For the video in the same resolution, the time consumed by the improved encryption algorithm was less. It was because the improved encryption algorithm prioritized the encryption of the codeword with higher sensitivity, which not only improved the targeting but also reduced the number of codewords to be encrypted.

4 Conclusion

This paper briefly introduces HEVC technology and the video encryption algorithm based on the entropy coding of HEVC, improved the encryption algorithm based on the codeword sensitivity in entropy coding, and then conducted simulation experiments. The results are as follows.

1) The sign of transformation coefficients and the intra-frame prediction pattern were the first-level encryption sensitivity of I and P frames, the sign of quantization parameters was the second-level encryption sensitivity of I and P frames, the sign of motion vector differences was the second-level encryption sensitivity of P frame, the value of quantization parameters was the third-level encryption sensitivity of P frame, and the value of motion vector differences was the third-level encryption sensitivity of P frame.

2) The original video content was almost invisible after the video was encrypted by the improved encryption algorithm, and the PSNR of the video encrypted by the improved encryption algorithm was significantly lower than that by the non-improved encryption algorithm in the face of videos in different resolutions.

3) The time consumption of both encryption algorithms increased with the increase of video resolution, while the improved encryption algorithm took less time to encrypt the video in the same resolution than the non-improved encryption algorithm.

Figure 5: The original video image and the images after processing by two encryption algorithms

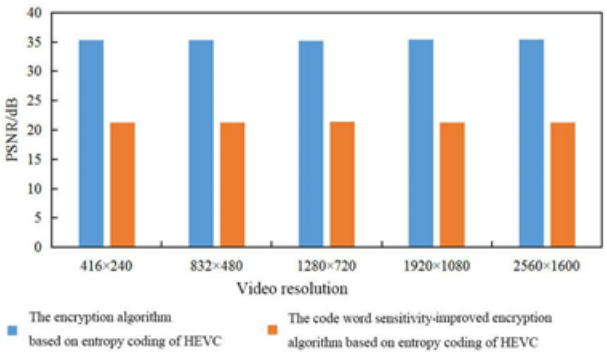


Figure 6: PSNR of videos in different resolutions after encryption by two encryption algorithms

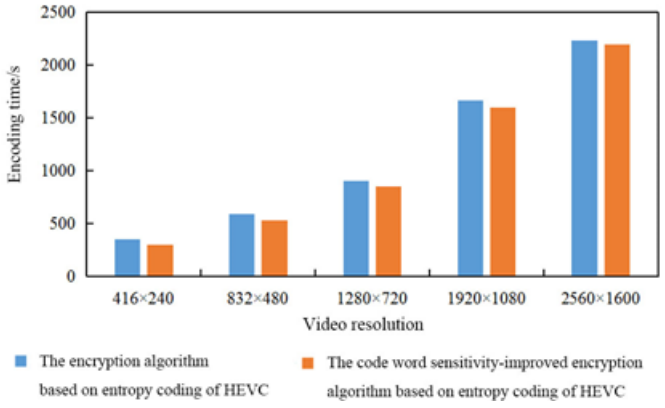


Figure 7: Encryption time of two encryption algorithms for videos in different resolutions

References

[1] A. Alfalou, C. Brosseau, N. Abdallah, "Simultaneous compression and encryption of color video images," *Optics Communications*, vol. 338, pp. 371-379, 2015.

[2] A. Chadha, S. Mallik, A. Chadha, R. Johar, M. M. Roja, "Dual-layer video encryption using RSA algorithm," *International Journal of Computer Applications*, vol. 116, no. 1, pp. 33-40, 2015.

[3] J. Chandrasekaran, S. J. Thiruvengadam, "Ensemble of chaotic and naive approaches for performance enhancement in video encryption," *The Scientific World Journal*, vol. 2015, pp. 1-11, 2015.

[4] M. Farajallah, W. Hamidouche, O. Deforges, S. E. Assad, "ROI encryption for the HEVC coded video contents," in *IEEE International Conference on Image Processing (ICIP'15)*, vol. 2015, pp. 3096-3100, 2015.

[5] B. Guan, D. Xu, Q. Li, "An efficient commutative encryption and data hiding scheme for HEVC video," *IEEE Access*, vol. 8, pp. 60232-60245, 2020.

[6] B. Jovanovic, S. Gajin, "An efficient mechanism of cryptographic synchronization within selectively encrypted H.265/HEVC video stream," *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 1537-1553, 2018.

[7] M. Long, F. Peng, H. Y. Li, "Separable reversible data hiding and encryption for HEVC video," *Journal of Real Time Image Processing*, vol. 14, no. 1, pp. 171-182, 2018.

[8] D. Valli, K. Ganesan, "Chaos based video encryption using maps and Ikeda time delay system," *European Physical Journal Plus*, vol. 132, no. 12, pp. 542, 2017.

[9] H. Xu, X. Tong, X. Meng, "An efficient chaos pseudo-random number generator applied to video encryption," *Optik - International Journal for Light and Electron Optics*, vol. 2016, pp. 9305-9319, 2016.

[10] N. Saini, A. Sinha, "Video encryption using chaotic masks in joint transform correlator," *Journal of Optics*, vol. 17, no. 3, pp. 73-93, 2015.

[11] A. I. Sallam, E. S. M. El-Rabaie, O. S. Faragallah, "Efficient HEVC selective stream encryption using chaotic logistic map," *Multimedia Systems*, vol. 24, no. 4, pp. 419-437, 2018.

[12] R. A. Shah, M. N. Asghar, S. Abdullah, M. Fleury, N. Gohar, "Effectiveness of crypto-transcoding for H.264/AVC and HEVC video bit-streams," *Multimedia Tools and Applications*, vol. 78, no. 15, pp. 21455-21484, 2019.

[13] N. F. Soliman, M. I. Khalil, A. D. Algarni, S. Ismail, R. Marzouk, W. El-Shafai, "Efficient HEVC steganography approach based on audio compression and encryption in QFFT domain for secure multimedia communication," *Multimedia Tools and Applications*, vol. 2020, no. 2, 2020.

[14] K. Thiyagarajan, R. Lu, K. El-Sankary, H. Zhu, "Energy-aware encryption for securing video transmission in internet of multimedia things," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 29, no. 3, pp. 610-624, 2019.

[15] M. A. Usman, M. R. Usman, S. Y. Shin, "Quality assessment for wireless capsule endoscopy videos compressed via HEVC: From diagnostic quality to visual perception," *Computers in Biology and Medicine*, vol. 91, no. 4, pp. 112-134, 2017.

Biography

Wenwen Li has received the master's degree of engineering from Wuhan University in 2013. She is an associate professor in Chuzhou Polytechnic. Her research interests include digital media, industrial design and software engineering.

Hongfei Xiao has received the master's degree of engineering from Hefei University of Technology in 2016. He is an associate professor. He is a doctoral student of Chongqing University and is working in Chuzhou Polytechnic. His research interests include information security and data mining.

Shiqi Tang has received the master's degree from Inner Mongolia Normal University in 2019. He is working in Information Center of Ministry of Science and Technology. His research interests include digital image processing and information security.

A Fusion Malicious Social Bots Detection Model Based on Static and Dynamic Features

Hongling Jiang, Dan Liu, Haiyan Kang, and Yilin Wang
(Corresponding author: Hongling Jiang)

School of Information Management, Beijing Information Science and Technology University
No.12 Xiaoying East Qinghe Road, Haidian District, Beijing 100192, China
Email: jhl@mail.bistu.edu.cn

(Received Aug. 4, 2021; Revised and Accepted Jan. 31, 2022; First Online Feb. 26, 2022)

Abstract

Since social networks have brought great convenience to people, there are also many security threats. One of the threats is the malicious social bot which seriously pollutes the social network environment and disturbs the regular operation of the social network. This paper proposes a fusion model to detect the malicious social bot. We design dynamic features that can better distinguish malicious social bots from normal users, and we combine static features with dynamic features to better detect social bots. Our detection model fuses three machine learning models, including Random Forest, Decision Tree, and Multi-layer Perceptron, and makes a final decision. A new dataset is proposed to evaluate our proposal and compare it with others. The experimental results show that our proposal has achieved higher precision, recall, and F1-score.

Keywords: Dynamic Features; Fusion Model; Malicious Social Bots; Social Network; Static Features

1 Introduction

Social networks, such as Twitter, Sina Weibo (one of the most popular Chinese social networks), have become a very important part of human life. People can get information, show themselves, share opinions and experiences, and engage in online marketing through the social network. But at the same time, the social network becomes a new platform for criminals to derive great benefits. There are a lot of malicious social bots on social networks. The malicious social bots are mostly used for malicious tasks. They disrupt the natural marketing of social platforms, threaten the credibility of social platforms, infringe on the public interests, and pollute the environment of social networks [13, 16, 17].

Social bots generally refer to malicious social bots which often carry out malicious activities. Social bots are usually automated software that are used to mislead, mine, and even manipulate social networks [8, 20]. For example, social bots can spread rumors, spam, phish, viruses, carry out Sybil attacks, act as Zombie users, and manipulate the stock market, etc [10,18,22]. To evade the detection system, social bots usually imitate the behavior of human users, which makes the detection of social bots more challenging. Therefore, the research on social bots detection has become an important field of network security.

This paper aims to propose a novel social bot detection approach to improve the performance of the detector. The main contributions of our work are as follows:

1) Since there are few public datasets in the field of social bot detection, we build a new dataset to evaluate the detection model. The dataset is built by crawling the basic and dynamic information of users of Sina Weibo.

2) Several dynamic features are designed to better distinguish social bots and normal users. Dynamic features are features extracted from the user's recent behavior. The experimental results show that dynamic features have better performance than static features.

3) A fusion detection model is proposed which merges the detection results of multiple models and gets the final detection result. The experimental results show that the fusion model performs better than a single model, and it can identify social bots effectively.

The paper is organized as follows. Section 2 introduces the related work. Section 3 describes the method proposed in this paper. Section 4 presents the experiments and results. Section 5 concludes this paper and foresees the future work.

2 Related work

Nowadays, social bot detection techniques have been widely studied by many research groups. We divide social bot detection approaches into 3 categories: (1) The

approaches based on crowd-sourcing [3, 6]; (2) The approaches based on graph [1, 11, 27]; (3) The approaches based on machine learning models [7, 9]. Among these three categories, machine learning-based approaches are the most powerful effective techniques. In this paper, we focus on the machine learning-based approaches, and some of them are introduced below.

Beskow and Carley [4] detect social bots by their names. They propose a random string detection model to label the string as random or not random. They develop a combination of heuristic filtering and traditional machine learning models. The features they extract from the strings are based on character n-gram. They merge several features together and include the Shannon string entropy in their feature set. The models they use are Naive Bayes, Logistic Regression, and Support Vector Machines.

Wu *et al.* [26] propose a framework for detecting social bosts in Sina Weibo. Their approach is based on deep neural networks and active learning. They extract 30 features to distinguish social bots and normal users. A deep neural network model is built to detect social bots.

Shi *et al.* [23] propose a method to detect malicious social bots. Their method includes both features selection and semi-supervised clustering. They analyze the transition probability of user behavior clickstreams and the time feature of behavior. Their experimental platform is the online media social network platform CyVOD.

Wang *et al.* [24] propose a social bot detection framework that is based on a combination of a Variational AutoEncoder and an anomaly detection algorithm. They first use the Variationla AutoEncoder to encode and decode sample features. Then they combine the decoding representation and the original features. Finally, they use an anomaly detection method to detect bots. They use the CLEF2019 dataset [19], which aims to identify the nature of the Twitter account.

Loukas *et al.* [12] introduce two methods to detect bots in social network. The first method is a feature extraction approach to identify accounts posting automated messages. The second method is a deep learning architecture to identify whether tweets are posted by real users or bots. They use the attention mechanism to identify bots. They use two Twitter datasets to evaluate their methods.

Lingam *et al.* [15] design a Particle Swarm Optimization (PSO) based deep Q-learning algorithm to detect social spam bots. They integrate PSO with the Q-value function. A spam influence minimization model is proposed to identify the spam-influential users. Besides, an influential community detection algorithm is proposed to reduce the spreading of spam content through influential communities in the Twitter network. Two real-time Twitter datasets, social honeypot dataset [14] and the fake project dataset [6], are used to evaluate their algorithms.

Abulaish *et al.* [2] present a machine learning-based approach to detect social bots. They classify social bots into three categories: active, reactive, and inactive users. They use three machine learning classifiers: Naive Bayes, REP Decision Tree, and Random Forest. They evaluate their approach by using a dataset collected from a live social bot injection experiment conducted on Twitter.

Wu *et al.* [25] propose an improved conditional generative adversarial network (improved CGAN) to detect social bots. They first use a modified gaussian kernel density peak clustering algorithm (GKDPCA) to generate an auxiliary condition. Then they introduce the wasserstein distance with a gradient penalty to improve the CGAN convergence judgement condition.

Schuchard *et al.* [21] introduce an ensemble social bot detection coverage framework. Their framework utilizes the power of multiple detection sources to detect a wider variety of bots in a given online Twitter social network corpus. They find that the incorporated social bot detection platform is effective.

Beskow *et al.* [5] propose the bot-hunter to detect social bots. The bot-hunter applies a Random Forest model on Twitter data in a multi-tiered fashion. They compare bot-hunter to other similar models and find that the Random Forest model performs best.

Among the above approaches, some of them are based on textual information, such as user name or published content. However, some intelligent bots leverage artificial intelligence techniques to generate textual content. In this way, the textual contents of social bots will be similar to those written by normal users. Therefore, in our work, we will not consider the feature of textual content. Other approaches extract features of users' behavior or profile, but some of them extract too many features. If there are too many features, it will increase the amount of calculation and reduce the detection efficiency. In our approach, we extract fewer features, which can better distinguish social bots and normal users. Some approaches use a single machine learning algorithm to detect social bots. A single detection algorithm will have a great impact on detection performance. In our work, we propose a novel approach that could merge multiple algorithms and improve the performance of the detection.

3 The Proposed Method

3.1 Framework

The framework of the proposed social bot detection approach is shown in Figure 1. It contains three phases: building the dataset, extracting features, and detecting social bots.

Building the dataset is to build dataset D_1. We first obtain the nicknames of social bots and normal users in Sina Weibo. Then we crawl the basic and dynamic information of Sina Weibo users. After that, we insert the basic information of each user in front of each blog the user publishes. Finally, we preprocess the data, including dealing with missing values, deleting outliers, and removing duplicate data in the dataset D_1. For more details about building the dataset, see 3.2.

Extracting features is divided into two parts, including extracting static features and calculating dynamic fea-

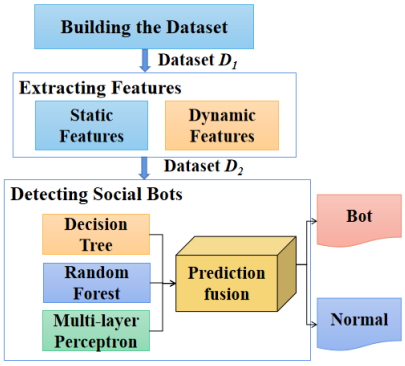


Figure 1: The framework of our approach

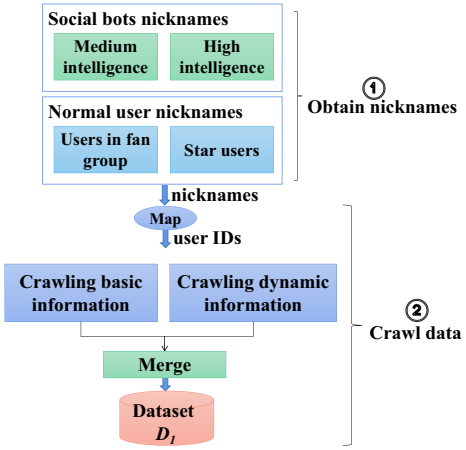


Figure 2: The process of building dataset D_1

tures. After features extraction, dataset D_2 is obtained. Dataset D_2 will be fed into the detection model. We divide the dataset D_2 into training set and testing set. The ratio of training set and testing set is 3:1. For more information about extracting features, see 3.3.

Detecting social bots is to fuse the detection results of multiple models. After analysis and comparison in the experiments, we choose Decision Tree, Random Forest, and Multi-Layer Perceptron models, and merge their prediction results to get the final detection result. For more details, see 3.4.

3.2 Building the Dataset

The process of building the dataset D_1 is shown in Figure 2. The dataset D_1 is merged by the basic and dynamic information of Sina Weibo users. The process of building D_1 contains the following two stages:

The first stage is to obtain nicknames.

To evaluate the detection model, there should be both normal users and malicious social bot users in D_1.

In order to make the dataset used in our paper more diverse, the normal users contain two parts: 1) Users in fan groups. The users in fan groups gathered because of their common interests. Users need to apply to join the fan group. A user can join the fan group only after the fan group owner passes the user's application. Therefore, users in the fan group can be regarded as normal users. We pick up 425 users from the fan group. 2) Star users. There are a large number of star users in Sina Weibo. Compared with ordinary users, such star users are more influential. For example, they have more fans, and their number of likes, comments and forwards is also larger. We select 100 star users as part of the normal users.

To get the nicknames of malicious social bots, we have purchased 500 nicknames of malicious social bots from Sina Weibo fan vendors. These bot nicknames have different intelligence levels, including 300 medium level bots and 200 high level bots. The medium level bots have medium intelligence, while the high level bots have advanced intelligence. Compared with medium level bots, high level bots can better imitate the behavior of normal users. Due to Sina Weibo has its own malicious social bots detection mechanism, by the time of crawling the data of these malicious social users, two medium level bots have been banned. In the end, the real number of bots in D_1 is 498.

All the data we collected in this stage are nicknames, including normal users' nicknames and social bots' nicknames. The statistics of nicknames are shown in Table 1.

Table 1: The statistics of nicknames

Category	Subcategory	Number of users
Social bots	Medium intelligence	298
	High intelligence	200
Normal users	Users in a fan group	425
	Star users	100
Total	-	1023

The second stage is to crawl the data of these users.

This stage can be completed in four steps:

1) We map nicknames to user IDs. The user ID is the unique identifier of a user. Because nicknames are not unique, and they will be changed by users. The user ID is unique and unchanged, so it can uniquely identify a user.
2) We crawl the user's basic information based on the user ID. The basic information includes userID, nickname, regtime, gender, birthday,

describe, address, credit and authentication, as shown in Table 2.

Table 2: The basic information of users

#	Name	Description
1	*userID*	Unique identification
2	*nickname*	The nickname of the user
3	*regtime*	The register time
4	*gender*	The gender set by the user
5	*birthday*	The birthday set by the user
6	*describe*	The describe set by the user
7	*address*	The address set by the user
8	*credit*	The Sunshine Credit of Sina Weibo
9	*authentication*	Authentication of the user

3) We crawl the dynamic information of users. We set our crawler to crawl 10 screens of blogs (about 100 pieces of blogs). If some users publish few blogs, such as less than 100, we crawl all their blogs. Firstly, we crawl the dynamic information of each user, including the number of blogs recently published by the user (denoted as *blogs_count*), the number of fans owned by the user (denoted as $fans\_count$), the number of friends owned by the user (denoted as $friends\_count$). Secondly, for each blog published by the user, we crawl the following information: *device*, *publish_time*, *link*, *content*, *forwards_count*, *comments_count* and *likes_count*. The dynamic information crawled in this step is shown in Table 3.

Table 3: The dynamic information crawled

#	Name	Description
1	*userID*	Unique identification
2	*blogs_count*	The number of blogs recently published by the user
3	*fans_count*	The number of fans owned by the user
4	*friends_count*	The number of friends owned by the user
5	*device*	Device for publishing blogs
6	*publish_time*	The time of publishing the blog
7	*link*	The link to the blog
8	*content*	The content of the blog
9	*forwards_count*	The number of forwards
10	*comments_count*	The number of comments
11	*likes_count*	The number of likes

4) We merge the basic and dynamic information of users. After the above two steps, we obtain the basic and dynamic information of users. There are 498 malicious social bots and 525 normal users in the dataset, a total of 1023 users. These users have recently published 93412 blogs. For each blog, we insert the information of the user to the front of the blog and build the dataset D_1. The dataset D_1 as the original dataset is the basis for subsequent work.

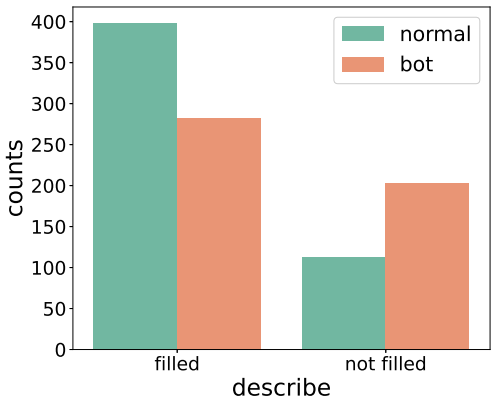


Figure 3: Distribution of feature *describe*

3.3 Extracting Features

Features are the key factors that affect the performance of the detection model. In this paper, we extract effective features from both static and dynamic perspectives.

3.3.1 Static Features Extraction

Static features will be selected from the basic information of the user (see Table 2). Since static features usually do not change, they are stable. We have selected three static features from the dataset D_1. These three static features could distinguish social bots and normal users better than others. These three static features are *describe*, *address* and *authentication*.

1) *describe*
We found that normal users usually fill in the description. Because the description is used to introduce themselves and attract others' attention. While most malicious social bots will not fill in the description. We compare the distribution of feature *describe* between malicious social bots and normal users, as shown in Figure 3. We can see that about 80 percent of normal users fill in the description. While fewer malicious social users pay attention to the description.

2) *address*
When registering, a user can fill in the address selectively. The distribution of feature *address* is shown in the Figure 4. It shows that malicious social bots have

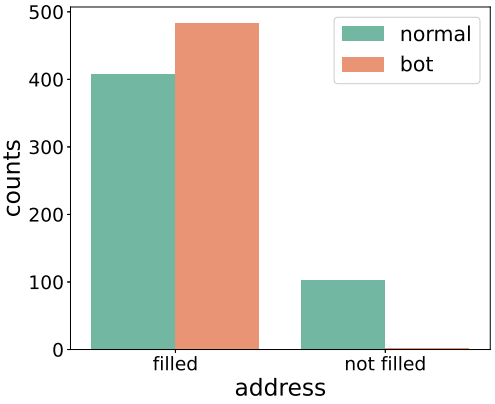


Figure 4: Distribution of feature *address*

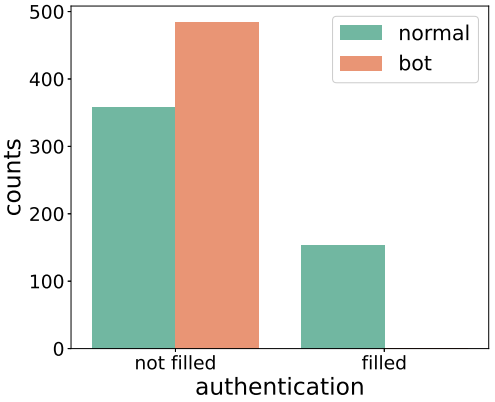


Figure 5: Distribution of feature *authentication*

a certain level of intelligence, and most of them fill in the address. However, due to privacy protection, some normal users do not fill it. It is proved that this feature can play a certain role in the classification.

3) *authentication*
The authentication can increase user's influence, credibility and obtain certain privileges. The authentication of Sina Weibo includes personal identity authentication, enterprise authentication, and so on. The assessment content of each kind of authentication is different. Users need to meet certain conditions to be certified. So usually only normal users will consider getting authentication, while malicious social bots will not consider it.

The distribution of feature *authentication* is shown in the Figure 5. It shows that many normal users have been authenticated, while malicious social bots usually do not apply for authentication. It is proved that feature *authentication* has a good effect on the classification.

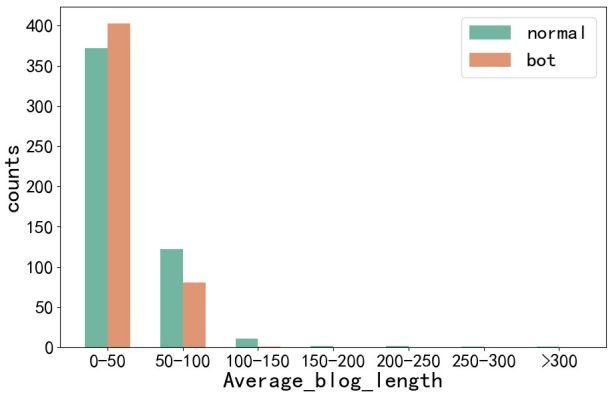


Figure 6: The distribution of feature *Average_blog_length*

3.3.2 Dynamic Features Extraction

In order to detect the social bot more efficiently, this paper proposes some dynamic features, which are calculated according to the dynamic information crawled (see Table 3). Since dynamic features will change with the user's behavior, they are unstable. Dynamic features may have different values in different time periods. We extract a total of 9 dynamic features. The following will explain why they were chosen and how they were obtained.

1) *Average_blog_length*
An important difference between social bots and normal users is the content of blogs. Normal users publish blogs to show themselves. However, the blog content of social bots is automatically generated by the bot program. One of the metrics is the length of the blog. We propose a feature named as *Average_blog_length* that reveals the average length of recently published blogs. *Average_blog_length* is computed as Equation (1).

$$Average\_blog\_length = \frac{len\_sum}{blogs\_count} \tag{1}$$

The *len_sum* is the sum of lengths of all blogs the user recently published, and *blogs_count* is the number of blogs recently published by the user. The distribution of feature *Average_blog_length* is shown in the Figure 6. It shows that the length of blogs published by social bots is mostly short, while the blogs published by normal users have different lengths.

2) *Forward_proportion*
Since social bots usually have a large number of forwarding behaviors, measuring forwarding behavior can distinguish bots from normal users. We propose a feature called *Forward_proportion* that is the proportion of forwarding behavior according to the recently published blogs. We use Equation (2) to extract *For-*

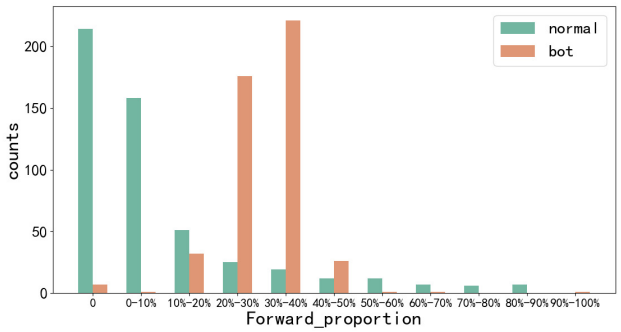


Figure 7: The distribution of feature *Forward_proportion*

ward_proportion.

$$Forward\_proportion = \frac{forwards\_count}{blogs\_count} \quad (2)$$

The *forwards_count* is the number of forwarded blogs, and *blogs_count* is the number of blogs recently published by the user (see Table 3). The distribution of feature *Forward_proportion* is shown in the Figure 7. It shows that the proportion of social bots' forwarding behavior is significantly higher than normal users. Most normal users do not have a high proportion of forwarding behavior.

3) *Average_blog_counts_per_day*
The frequency of publishing blogs is also an important indicator. We propose a feature *Average_blog_counts_per_day* to measure the frequency. *Average_blog_counts_per_day* is calculated as Equation (3). The *dur* is the time interval between the first blog and the last blog, and it is measured in days.

$$Average\_blog\_counts\_per\_day = \frac{blogs\_count}{dur} \quad (3)$$

The distribution of feature *Average_blog_counts_per_day* is shown in the Figure 8. It can be seen that the frequency of blog publishing by normal users is relatively random, while the frequency of social bot publishing is within a certain range.

4) Interaction related features: *Average_forwards*, *Average_comments*, *Average_likes*
The interaction indicators reflect the interaction behavior between a certain user and other users. Forward, comment, and like are all interactive behaviors. We focus on the user's recent average number of forwards, comments and likes. We propose three features, *Average_forwards*, *Average_comments*, *Average_likes*, to measure the average number of forwards, comments and likes respectively. They are

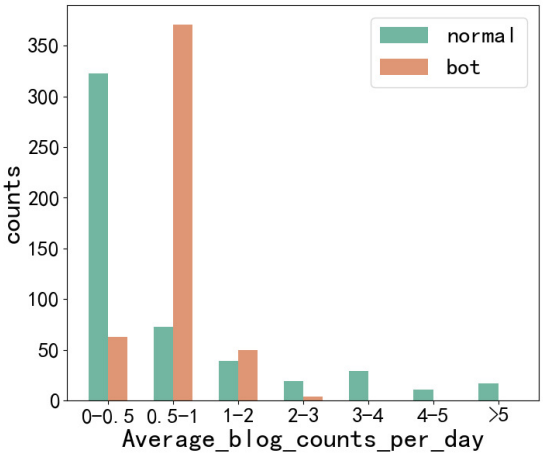


Figure 8: The distribution of feature *Average_blog_counts_per_day*

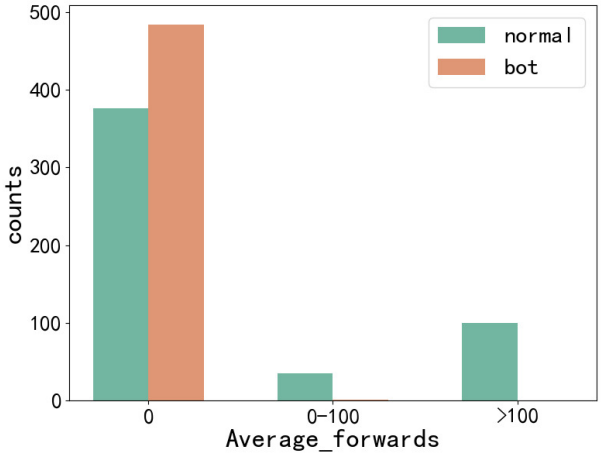


Figure 9: The distribution of feature *Average_forwards*

calculated as the Equation (4), (5), (6) respectively.

$$Average\_forwards = \frac{forwards\_count}{blogs\_count} \quad (4)$$

$$Average\_comments = \frac{comments\_count}{blogs\_count} \quad (5)$$

$$Average\_likes = \frac{likes\_count}{blogs\_count} \quad (6)$$

The *forwards_count*, *comments_count*, *likes_count* are the number of forwards, comments, and likes respectively. The distributions of *Average_forwards*, *Average_comments*, *Average_likes* are shown in Figures 9, 10, 11. Obviously, the average number of forwards, comments, and likes of social bots is low, because social bots have less interaction than normal users.

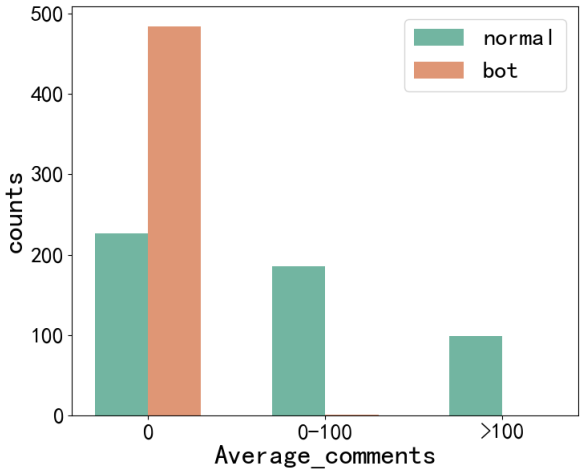


Figure 10: The distribution of feature *Average_comments*

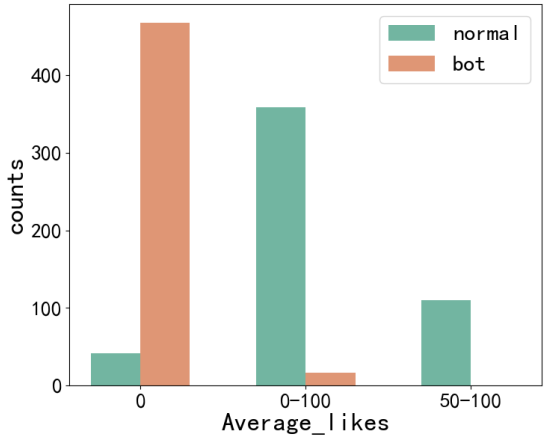


Figure 11: The distribution of feature *Average_likes*

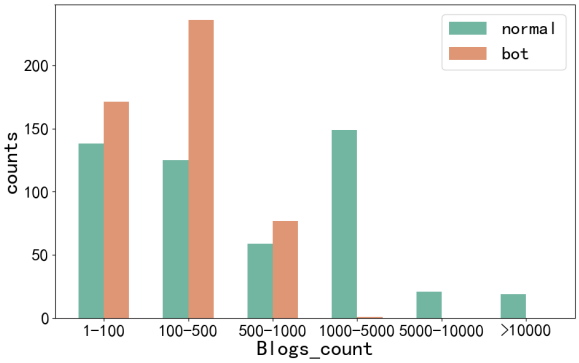


Figure 12: The distribution of feature *Blogs_count*

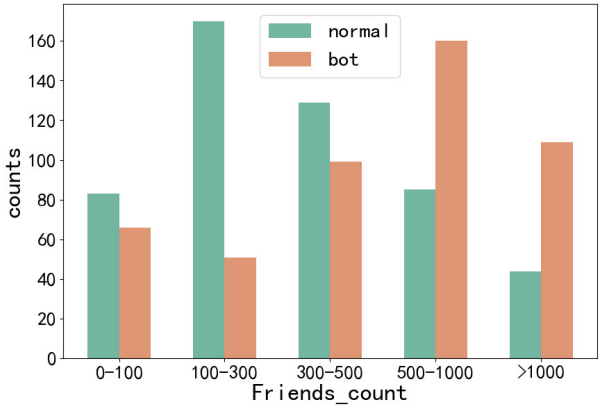


Figure 13: The distribution of feature *Friends_count*

5) Impact related features: *Blogs_count*, *Friends_count*, *Fans_count*
The Impact related indicators reflect the influence of users, including the number of blogs (*Blogs_count*), the number of friends (*Friends_count*) and the number of fans (*Fans_count*).

The distribution of the three features are shown in Figures 12, 13, 14. We can see that the number of blogs published by social bots is within a certain range, while the number of normal user blogs is widely distributed. The social bot usually follow many other users. The peak number of friends followed by social bots is between 500 and 1000, while the peak value of normal users is between 100 and 300. The number of fans of social bots is relatively small, while the number of fans of normal users is relatively large. It shows that these features have a good classification effect.

3.3.3 Feature Vectors Formation

For each user u in dataset D_1, we first extract the above static features and dynamic features of it. Then we merge static features and dynamic features into a feature vector F^u. Finally, we put all the feature vectors into the dataset D_2. D_2 will be used as the input of the detection model. The process of extracting features and building D_2 is shown in Algorithm 1. Where f^u_{s1}, f^u_{s2}, f^u_{s3} are the static features of user u, and f^u_{d1},f^u_{d2},...,f^u_{d9} are the dynamic features of u.

3.4 Detecting Social Bots

In this phase, we design a social bots detection model. Our model fuses the results of three basic models, namely Decision Tree (DT), Random Forest (RF), and Multilayer Perceptron (MLP). The social bots detection phase contains the following steps.

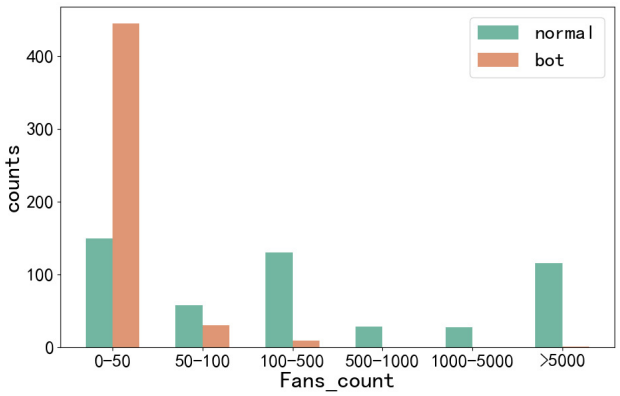


Figure 14: The distribution of feature *Fans_count*

Algorithm 1 Feature extraction

Input: Dataset D_1 (The data fields of D_1 are shown in Table 2 and Table 3).
Output: Dataset D_2.
Begin
for each user u in D_1 **do**
  $f_{s1}^u = describe_u$
  $f_{s2}^u = address_u$
  $f_{s3}^u = authentication_u$
  $blogs^u$ = All blogs recently published by u
  len_sum_u =Sum of lengths of all blogs in $blogs^u$
  $Average\_blog\_length_u = \frac{len\_sum_u}{blogs\_count_u}$
  $f_{d1}^u = Average\_blog\_length_u$
  $Forward\_proportion_u = \frac{forwards\_count_u}{blogs\_count_u}$
  $f_{d2}^u = Forward\_proportion_u$
  dur_u = Time interval between the first and the last blog published by u
  $Average\_blog\_counts\_per\_day_u = \frac{blogs\_count_u}{dur_u}$
  $f_{d3}^u = Average\_blog\_counts\_per\_day_u$
  $Average\_forwards_u = \frac{forwards\_count_u}{blogs\_count_u}$
  $f_{d4}^u = Average\_forwards_u$
  $Average\_comments_u = \frac{comments\_count_u}{blogs\_count_u}$
  $f_{d5}^u = Average\_comments_u$
  $Average\_likes_u = \frac{likes\_count_u}{blogs\_count_u}$
  $f_{d6}^u = Average\_likes_u$
  $f_{d7}^u = blogs\_count_u$
  $f_{d8}^u = friends\_count_u$
  $f_{d9}^u = fans\_count_u$
  $F^u = \{f_{s1}^u, f_{s2}^u, f_{s3}^u, f_{d1}^u, f_{d2}^u, f_{d3}^u, f_{d4}^u, f_{d5}^u, ..., f_{d9}^u\}$
  insert F^u into D_2
end for
Return D_2
End

1) Normalization
We get the dataset D_2 in the above phase, then we normalize the samples in D_2. Normalization can improve the performance of detection models. In this paper, we use the Min-Max method, as shown in Equation (7).

$$F_i^{u\prime} = (F_i^u - F_i^u{}_{min}) \frac{b-a}{F_i^u{}_{max} - F_i^u{}_{min}} + a \quad (7)$$

While F_i^u is the value of the ith feature of the original feature vector F^u, and $F_i^{u\prime}$ is the normalized value of F_i^u. $F_i^u{}_{min}$ and $F_i^u{}_{max}$ are the initial minimum and maximum values of the ith feature of F^u respectively, a and b are the new minimum and maximum values. In our paper, $a = 0, b = 1$. All the features are converted to the range within [0,1].

2) Basic Models Training
We choose three basic models, Decision Tree (DT), Random Forest (RF), and Multi-layer Perceptron (MLP) as the basic models. The normalized samples are divided into training and testing datasets, and the ratio is 3:1. The normalized samples in the training dataset are fed into each model, and the detection result of each model is the probability of normal user or social bot. After the training is completed, the parameters of these three basic models will be saved for subsequent detection of social bots.

3) Social Bots Detection
The final detection model combines the detection results of the three basic models. The social bot detection algorithm is shown in Algorithm 2. We use the soft voting method, which is also known as weighted average probability voting. The final model takes the average of the detection probabilities of the three basic models. The final detection result p is calculated as Equation (8).

$$p = \frac{p_d + p_r + p_m}{N} \quad (8)$$

Where p_d, p_r, and p_m are the detection probability of DT, RF, and MLP model respectively, and N is the number of models, here $N = 3$.

4 Experiments

4.1 Environment and dataset

In this section, we present the experiments. We first explain the metrics used for evaluation. Then we demonstrate the performance of our proposed model on the dataset we build. The method is implemented using pycharm. All experiments are performed on a 64-bit Windows 10 with an Intel i5 CPU, 4 cores, 8GB RAM, and the CPU clock rate is 1.8GHz.

In the experiments, we use the proposed dataset D_1. The process of building the dataset D_1 is presented in 3.2. The data fields of D_1 contain *userID*, *nickname*, *regtime*, *gender*, *birthday*, *describe*, *address*, *credit*, *authentication*, *blogs_count*, *fans_count*, *friends_count*, *device*, *publish_time*, *link*, *content*, *forwards_count*, *comments_count* and *likes_count*.

Algorithm 2 Social bots detection

Input:

1) Dataset D_2.
2) Trained Decision Tree classifier $C_d()$.
3) Trained Random Forest classifier $C_r()$.
4) Trained Multi-layer Perceptron classifier $C_m()$.

Output: The detection result *Res* of Dataset D_2.
Begin
for each sample F^u in D_2 **do**
  $F^{u\prime} = normalization(F^u)$
  $p_d = C_d(F^{u\prime})$
  $p_r = C_r(F^{u\prime})$
  $p_m = C_m(F^{u\prime})$
  $p = avg(p_d, p_r, p_m)$
  Insert p into *Res*
end for
Return *Res*
End

4.2 Metrics

The main metrics used in our paper include *precision*, *recall*, *F1-score*, and ROC curve. We first introduce the confusion matrix on which these metrics are based, and then introduce these metrics.

1) Confusion Matrix
A confusion matrix is a specific table as shown in Table 4. The confusion matrix allows visualization of the performance of an algorithm. The terms *True Positives* (*TP*), *True Negatives* (*TN*), *False Positives* (*FP*), and *False Negatives* (*FN*) compare the results of the classifier under predictive with actual class.

Table 4: Confusion matrix

Predictive \ Actual	Positive	Negative
Positive	*TP*	*FP*
Negative	*FN*	*TN*

2) *precision*
The *precision* refers to the proportion of real social bots in the samples predicted to be social bots. The calculation is shown in Equation (9). The greater the *precision*, the higher the proportion of correctly detected social bots.

$$precision = \frac{TP}{TP + FP} \tag{9}$$

3) *recall*
The *recall* refers to the proportion of all real social bots that are predicted to be social bots. The calculation is shown in Equation (10). The larger the *recall*, the more social bots that are correctly detected.

$$recall = \frac{TP}{TP + FN} \tag{10}$$

4) *F1-score*
The *F1-score* is the harmonic mean of *precision* and *recall*. The calculation is shown in Equation (11).

$$F1 - score = \frac{2 * precision * recall}{precision + recall} \tag{11}$$

5) Receiver Operating Characteristic Curve (ROC)
The Receiver Operating Characteristic Curve (ROC) is a graphical plot that illustrates the diagnostic ability of a binary classifier system. The ROC curve is created by plotting the *True Positive Rate* (*TPR*) against the *False Positive Rate* (*FPR*) at various threshold settings. The *TPR* is also known as the recall of detection model. The *FPR* is also known as the probability of false alarm.

ROC space defines the *FPR* as the X axis and the *TPR* as the Y axis. Given the binary classification model and its threshold value, the (X=*FPR*, Y=*TPR*) coordinate points can be calculated from the (positive/negative) true and predicted values of all samples.

The AUC (Area Under Curve) is defined as the area under the ROC curve and the coordinate axis. Obviously, the value of AUC will not be greater than 1. Since the ROC curve is generally above the straight line $y = x$, the value range of AUC is between 0.5 and 1. The closer the AUC is to 1, the better the performance of the detection model.

4.3 Comparison of Static and Dynamic Features

We first evaluate the influence of static and dynamic features on detection models. We choose five commonly used models in the field of social bot detection, including Random Forest (RF), k-Nearest Neighbors (kNN), Logistic Regression (LR), Decision Tree (DT), and Multi-layer Perceptron (MLP). In the experiment, we compare the performance of detection models in the following three situations:

1) Use static features alone.
2) Use dynamic features alone.
3) Use both static and dynamic features.

In the experiment, *precision*, *recall* and *F1-score* are used as evaluation metrics of detection models.

In the first situation, only static features are used. The results of different models are shown in Table 5. It shows

that the performance of models built with only static features is not very good. Except for the metrics of Logic Regression exceeding 0.8, the metrics of other models are around 0.7.

Table 5: Results using static features alone

Model	*precision*	*recall*	*F1-score*
RF	0.7373	0.7324	0.7316
kNN	0.7102	0.7090	0.7083
LR	0.8385	0.8194	0.8164
DT	0.7148	0.7090	0.7077
MLP	0.7327	0.7157	0.7115

In the second situation, only dynamic features are used. The results of different models are shown in Table 6. It shows that the performance of the models has been significantly improved. The *precision*, *recall* and *F1-score* of RF and DT are more than 0.97, the metrics of MLP are above 0.9, and the metrics of kNN and LR are around 0.8. It shows that models built with only dynamic features have better performance than the first situation.

Table 6: Results using dynamic features alone

Model	*precision*	*recall*	*F1-score*
RF	0.9777	0.9766	0.9766
kNN	0.8471	0.8462	0.8460
LR	0.8101	0.7957	0.7620
DT	0.9733	0.9732	0.9732
MLP	0.9133	0.9130	0.9130

In the third situation, both static and dynamic features are used. The results of different models are shown in Table 7. By observing Table 7, we find that the *precision*, *recall* and *F1-score* of RF model are 0.995, the *precision*, *recall* and *F1-score* of DT model are about 0.98, and the *precision*, *recall* and *F1-score* of kNN and MLP model are all above 0.9. The three models have achieved the best results.

Table 7: Results using both static and dynamic features

Model	*precision*	*recall*	*F1-score*
RF	0.9951	0.9950	0.9950
kNN	0.9134	0.9100	0.9102
LR	0.7985	0.7700	0.7687
DT	0.9802	0.9800	0.9800
MLP	0.9200	0.9200	0.9199

To understand the impact of the three situations on the model performance intuitively, the comparison diagrams of these five models are drawn respectively, as shown in Figure 15. We can see that except LR model, the performances of the other detection models with all features are better than that with static features or dynamic features alone. In addition, the performance with dynamic features alone is better than that with static features alone.

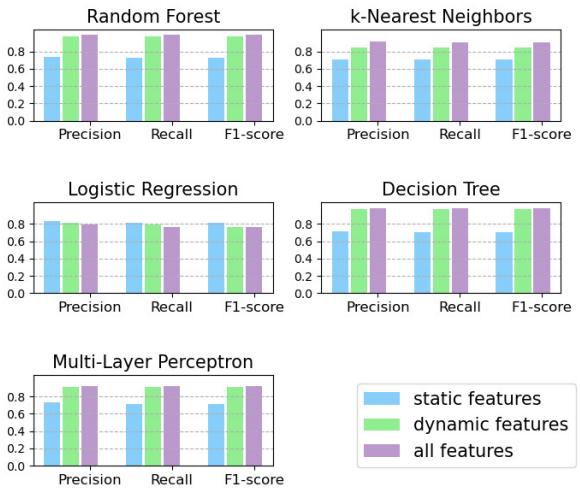


Figure 15: Comparison of models on the three situations

4.4 Comparison of Social Bots Detection Models

The comparison contains two parts. The first part is to compare the performance of different traditional machine learning models. The purpose of this comparison is to select models with better performance. The selected models will be used in our fusion model. The traditional machine learning models are Random Forest (RF), k-Nearest Neighbors (kNN), Logistic Regression (LR), Decision Tree (DT), Multi-Layer Perceptron (MLP). We select the three best performing models by comparing the detection results of them. Both static and dynamic features are used to train the five models. An important metrics to evaluate the performance of detection model is the ROC. The ROC of the five detection models is shown in Figure 16. The figure shows that the AUC area of Random Forest, Decision Tree and Multi-layer Perceptron is more than 0.98. The detection performance of K-Nearest Neighbors and Logistic Regression is worse than the other three models. Therefore, our fusion model uses RF, DT and MLP as the basic models.

The second part is to compare the performance of our proposal with the selected machine learning models, RF, DT and MLP. In the experiment, we train the three models use both static and dynamic features. The metrics we used are *precision*, *recall* and *F1_score*. The experimental results are shown in Table 8. As shown from Table 8, all metrics of our model are all above 0.996, which

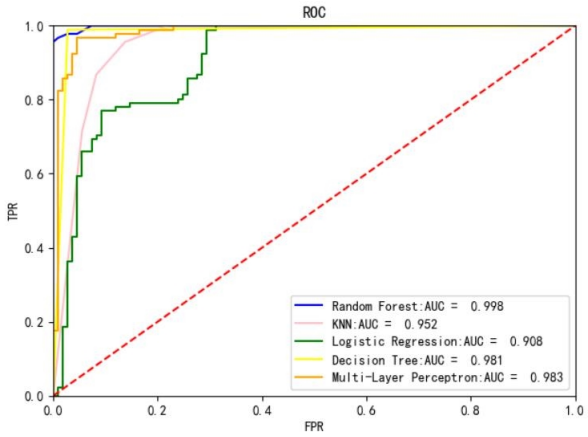


Figure 16: ROC of different detection models

is further improved compared with the single model in the previous section. Our model further improves the performance of the detection and enables it to better classify the social bot.

Table 8: Comparison of performances

Algorithm	*precision*	*recall*	*F1-score*
RF	0.99505	0.99500	0.99500
DT	0.98023	0.98000	0.98002
MLP	0.92001	0.92000	0.91991
Ours	**0.99668**	**0.99666**	**0.99666**

5 Conclusions

In this paper, we propose a fusion model to detect malicious social bots. We first obtain malicious social bot nicknames and normal user nicknames, and crawl the basic and dynamic information of users. Then, we propose 9 dynamic features to measure the recent behavior of users, and combine static and dynamic features to form feature vectors. Finally, a fusion model is proposed to detect social bots. Our fusion model merges the results of three models, Decision Tree, Random Forest, and Multilayer Perceptron. The experimental results show that the proposed dynamic features have better performance than static features. Furthermore, when combining dynamic and static features, the detection model achieves the best results. The comparison experimental results show that our fusion model achieves higher *precision*, *recall* and *F1-score* values than a single model, and can detect social bots more efficiently. In the future, we will continue to discover the features which can better distinguish the social bots and normal users, and design novel detection models to improve the detection performance.

6 Acknowledgments

This study was supported by the Beijing Municipal Education Commission Applied Basic Research Project (No.KM202011232022) and the National Social Science Fund Project (No.21BTQ079). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] N. Abu-El-Rub and A. Mueen, "Botcamp: Bot-driven interactions in social campaigns," in *The world wide web conference*, 2019, pp. 2529–2535.

[2] M. Abulaish and M. Fazil, "A machine learning approach for socialbot targets detection on twitter," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 3, pp. 4115–4133, 2021.

[3] A. Alarifi, M. Alsaleh, and A. Al-Salman, "Twitter turing test: Identifying social machines," *Information Sciences*, vol. 372, pp. 332–346, 2016.

[4] D. M. Beskow and K. M. Carley, "Its all in a name: detecting and labeling bots by their name," *Computational and Mathematical Organization Theory*, vol. 25, no. 1, pp. 24–35, 2019.

[5] D. M. Beskow and K. M. Carley, "Bot-hunter: a tiered approach to detecting & characterizing automated activity on twitter," in *Conference paper. SBP-BRiMS: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, vol. 3, 2018, p. 3.

[6] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," in *Proceedings of the 26th international conference on world wide web companion*, 2017, pp. 963–972.

[7] K. E. Daouadi, R. Z. Rebaï, and I. Amous, "Bot detection on online social networks using deep forest," in *Computer science on-line conference*. Springer, 2019, pp. 307–315.

[8] M. R. Faghani and U. T. Nguyen, "Mobile botnets meet social networks: design and analysis of a new type of botnet," *International Journal of Information Security*, vol. 18, no. 4, pp. 423–449, 2019.

[9] M. Fazil and M. Abulaish, "A hybrid approach for detecting automated spammers in twitter," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2707–2719, 2018.

[10] E. Ferrara, "Measuring social spam and the effect of bots on information diffusion in social media," in *Complex spreading phenomena in social systems*. Springer, 2018, pp. 229–255.

[11] S. Hurtado, P. Ray, and R. Marculescu, "Bot detection in reddit political discussion," in *Proceedings of the fourth international workshop on social sensing*, 2019, pp. 30–35.

[12] L. Ilias and I. Roussaki, "Detecting malicious activity in twitter using deep learning techniques," *Applied Soft Computing*, vol. 107, 2021.

[13] M. Latah, "Detection of malicious social bots: A survey and a refined taxonomy - sciencedirect," *Expert Systems with Applications*, vol. 151, no. 113383, pp. 1–21, 2021.

[14] K. Lee, B. D. Eoff, and J. Caverlee, "Seven months with the devils: A long-term study of content polluters on twitter," in *Fifth international AAAI conference on weblogs and social media*, 2011.

[15] G. Lingam, R. R. Rout, D. Somayajulu, and S. K. Ghosh, "Particle swarm optimization on deep reinforcement learning for detecting social spam bots and spam-influential users in twitter network," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2281–2292, 2020.

[16] A. Mueen, N. Chavoshi, and A. Minnich, "Taming social bots: Detection, exploration and measurement," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2967–2968.

[17] I. Pozzana and E. Ferrara, "Measuring bot and human behavioral dynamics," *Frontiers in Physics*, vol. 8, p. 125, 2020.

[18] M. Orabi, D. Mouheb, Z. A. Aghbari, and I. Kamel, "Detection of bots in social media: A systematic review," *Information Processing & Management*, vol. 57, no. 102250, pp. 1–23, 2020.

[19] F. Rangel and P. Rosso, "Overview of the 7th author profiling task at pan 2019: bots and gender profiling in twitter," in *Proceedings of the CEUR Workshop, Lugano, Switzerland*, 2019, pp. 1–36.

[20] L. Rheault and A. Musulan, "Efficient detection of online communities and social bot activity during electoral campaigns," *Journal of Information Technology & Politics*, pp. 1–14, 2021.

[21] R. J. Schuchard and A. T. Crooks, "Insights into elections: An ensemble bot detection coverage framework applied to the 2018 us midterm elections," *Plos one*, vol. 16, no. 1, pp. 1–19, 2021.

[22] C. Shao, G. L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," *Nature communications*, vol. 9, no. 1, pp. 1–9, 2018.

[23] P. Shi, Z. Zhang, and K. K. R. Choo, "Detecting malicious social bots based on clickstream sequences," *IEEE Access*, vol. 7, pp. 28 855–28 862, 2019.

[24] X. Wang, Q. Zheng, K. Zheng, Y. Sui, S. Cao, and Y. Shi, "Detecting social media bots with variational autoencoder and k-nearest neighbor," *Applied Sciences*, vol. 11, no. 12, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/12/5482

[25] B. Wu, L. Liu, Y. Yang, K. Zheng, and X. Wang, "Using improved conditional generative adversarial networks to detect social bots on twitter," *IEEE Access*, vol. 8, pp. 36 664–36 680, 2020.

[26] Y. Wu, Y. Fang, S. Shang, J. Jin, L. Wei, and H. Wang, "A novel framework for detecting social bots with deep neural networks and active learning," *Knowledge-Based Systems*, vol. 211, pp. 1–16, 2021.

[27] X. Zhang, H. Xie, and J. C. Lui, "Sybil detection in social-activity networks: Modeling, algorithms and evaluations," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*. IEEE, 2018, pp. 44–54.

Biography

Hongling Jiang received her Ph.D. in the Computer Science College of Nankai University, Tianjin, China, in 2013. She is currently a lecturer in the School of Information Management at Beijing Information Science and Technology University, Beijing, China. She has published more than ten papers in recent years. Her research interests focus on network security, artificial intelligence, and the Internet of things.

Dan Liu is a student of Beijing Information Science and Technology University, Beijing, China. She has participated in many research projects and has strong scientific research ability. Her research interests include social network, network security, big data analysis and mining.

Haiyan Kang was born in China in 1971. He is a supervisor, senior member of the China Computer Federation (No. E200028533M), ACM Membership (No. 9495204), and standing committee member of the privacy protection committee of China Confidentiality Association. He received his Ph.D. in computer application technology from Beijing Institute of Technology, China in 2005. His research interest fields include information system security, privacy preserving, and natural language processing (NLP). He is currently working as a professor at the Department of Information Security, School of Information and Management, Beijing Information Science and Technology University, Beijing, China.

Yilin Wang is currently a student of Beijing Information Science and Technology University, Beijing, China. She has strong programming and writing skills. Her research interests include network security, data mining, and social network.

Montgomery Algorithm Based on Co_Z Operation on Edwards Curves over Prime Field

Shuang-Gen Liu and Rong Lu
(Corresponding author: Shuang-Gen Liu)
School of Cyberspace Security, Xi'an University of Posts and Telecommunications
Xi'an 710121, China
Email: liusgxupt@163.com
(Received Aug. 4, 2021; Revised and Accepted Jan. 29, 2022; First Online Feb. 26, 2022)

Abstract

Based on the formula of point addition and point doubling on the Edwards elliptic curve on the prime field, combined with the method of calculating only y coordinate, the formula in the affine coordinate system was deduced, and Montgomery Ladder Algorithm Using Only-y-Coordinate has been given in combination with Lopez-Dahab's ideas. To reduce the inversion operation, we proposed transforming affine coordinate into standard projective coordinates and designing a Co_Z Montgomery Algorithm 5 to improve the Edwards elliptic curve. At the same time, a new algorithm, four based on the Markov chain, is proposed. Theoretical analysis shows that Co_Z Algorithm 5 is 21.4% faster than Yu W Weierstrass-form Elliptic Curves Montgomery algorithm. Algorithm 4 is 35.9% faster than Yu W Algorithm 3. The efficiency of the two algorithms is improved remarkably. The theoretical analysis is supported by elliptic curve experiments in the different binary prime fields.

Keywords: Co_Z Montgomery Algorithm; Edwards Curves; Scalar Multiplication

1 Introduction

Elliptic curve cryptography (ECC) was first proposed by Koblitz [11] and Miller [19] in 1985. The security and implementation efficiency of ECC has been widely studied by many mathematicians and cryptographers. ECC has an efficiency advantage over other public-key cryptography. For instance, a key length of 283-bits in ECC is regarded as secure as 3072-bits in RSA public-key cryptography [14]. This significant difference makes ECC particularly attractive for applications in restricted environments since a shorter key size can be converted to less power, storage, and computational time overhead.

Scalar multiplication kP is the most expensive ECC operation that adds P to itself k times such that $kP = P + P + ... + P$, where P is a point on an elliptic curve over finite fields and k is a large positive integer [21]. The efficiency of scalar multiplication in our framework relies on the method and the cost of formulas (*i.e.* point doubling ($2P$), point addition ($P + Q$)). The cost of $2P$ and $P + Q$ formulas varies in different coordinate systems over different finite fields.

We use inversions (I), multiplications (M) and square (S) to represent the calculation cost of the formula. We discuss a special class of curves in the prime fields, called Edwards curve $E_{a,d}$ from equation (1). To simplify the comparison over prime fields, multiplication by a curve constant is denoted by D, assume D$\approx$ 0 [2, 9]. Add/subtract operations will be ignored because they are cheap operations in prime and binary fields.

One technique for accelerating scalar multiplication is to use projective coordinate systems. Using standard projective coordinates to avoid inversion completely. Firstly, we convert an affine point into a projective point. Then, we perform the scalar multiplication algorithm without any inversion. Finally, we invert a projective point into an affine point. The cost of converting affine points to projective points or projective points to projective points is small compared with the cost of directly calculating scalar multiplication algorithms. The number of operations is significantly reduced, especially for devices with high I/M ratios. For this reason, many studies have proposed projective coordinate systems with different elliptic curves [4, 7, 9, 10].

In recent years, great progress has been made in the research of the Montgomery algorithm in the elliptic curve cryptosystem. In different coordinate systems, the representation of points on the elliptic curve is different, and the calculation formulas of point addition and point doubling are also different. In 1987, Montgomery [20] proposed an algorithm that can resist SPA, called the Montgomery algorithm. In 1999, Lopez and Dahab [18] optimized the Montgomery algorithm on $GF(2^m)$ elliptic curves. They gave the optimized Montgomery algorithm and a new formula for calculating point addition and point doubling. The new formula eliminates the calculation of y coordinates and improves the calculation speed of the

algorithm. In 2002, Brier and Joye [6] proposed how to rewrite the addition on the general Weierstrass form of elliptic curves so that the same formula applies equally to add two different points or to double a point. It also shows how to generalize to the Weierstrass form a protection method previously applied to a specific form of elliptic curves due to Montgomery. In 2007, Bernstein *et al.* [5] proposed fast explicit formulas for group operations on an Edwards curve. And it contains an extensive comparison of different forms of elliptic curves and different coordinate systems for the basic group operations as well as higher-level operations. In 2008, Wang *et al.* [27] proposed the formula of point addition and point doubling, which omitted the y-coordinate. It eliminated the inversion operation, thus improving the Montgomery algorithm in the finite field of feature 3. In the same year, Bernstein and Birkner [3] promoted the Twisted Edwards curve and put forward the formula for fast calculation of point addition and point doubling in the projection coordinate, and eliminated the inversion operation. In 2008, Patrick *et al.* [17] proposed a novel method to accelerate the elliptic curve point formula on the prime number field. This flexible technique takes advantage of the fact that domain squares are generally cheaper than multiplication, replacing multiplication with squares and other cheaper operations. In 2011, Rivain [24] introduced the regular implementation of the Montgomery ladder algorithm and a new binary algorithm and discussed this algorithm's security concerning for to side-channel attacks. And the author's work gives a clear view of the currently best time-memory trade-offs for regular implementation of scalar multiplication over prime-field elliptic curves. In the same year, Raveen *et al.* [23] presented further Co_Z addition formula for various point additions on Weierstrass elliptic curves. Moreover, they also explained how the use of conjugate point addition and other implementation tricks allow the developing of efficient scalar multiplication algorithms making use of Co_Z arithmetic. In 2014, Oliveira *et al.* [22] presented new arithmetic formulas for a projective version of the affine point representation, which leads to efficient computation of the scalar multiplication operation over binary elliptic curves. In 2015, Tang *et al.* [26] proposed that the Twisted Edwards curve could resist SPA attack well by using a unified point addition and point doubling formula. In 2017, Liu *et al.* [16] proposed a Markov point addition multiple point chains based on the Edwards curve. Based on this chain, the scalar multiplication algorithm performs point addition and point doubling operations every time, which can resist simple energy attacks. In 2017, Yu *et al.* [28] improved the point addition and point doubling formula by using the unified Z-coordinate technique and not calculating the y coordinate in the middle stage of the cycle and constructing the SPA-resistant Co_Z Montgomery algorithm. In 2019, Liu *et al.* [15] combined the ternary of k with the original Montgomery algorithm and proposed the ternary Montgomery algorithm. In 2019, Yu *et al.* [29] proposed an improved Montgomery algorithm, which has faster operation efficiency than Brier and Joye's Montgomery algorithm. In 2020, Abarzua *et al.* [1] showed good safety in Edwards elliptic curve.

The purpose of the study in this paper is whether the efficiency of the Montgomery algorithm on different curves can be improved? Therefore, we chose the Edwards curve for research. The biggest characteristic of the Edwards curve is that the rules in group operation are relatively simple, and the formula of point addition and point doubling is the same. In this paper, we adopt Lopez and Dahab's ideas and standard projection coordinates to improve the addition and multiplication formulas of points in the Edwards Curve in the prime number field. Combined with the unified Z-coordinate method, we also propose a Co_Z Montgomery Algorithm 5 of Edwards Curve. And we propose a new Algorithm 4 combined with the Markov chain, which is not only better than Yu W Algorithm 3, but also better than Co_Z Algorithm 5. Theoretical analysis shows that the cost of Co_Z Montgomery Algorithm 5 is 21.4% faster than Yu W Algorithm 3. Algorithm 4 is 35.9% faster than Yu W Montgomery Algorithm 3 [1].

This paper is organized as follows. In Section 2, we give some basic information about Edwards elliptic curves, Markov Chain, and Montgomery ladder algorithm. In Section 3, we introduce the Montgomery Ladder algorithm using only-y-coordinate on Edwards. In Section 4, we propose the Montgomery Ladder algorithm based on MADC and Co_Z Montgomery algorithm using only-y-coordinate. In Section 5, we analyze the complexity of Co_Z Algorithm 5 and Algorithm 4 in comparison with Yu W Montgomery Algorithm 3. Finally, we have made a summary.

2 Background Knowledge

2.1 Edwards Curves

Bernstein, Birkner, Joye, Lange, and Peters introduced Twisted Edwards curves over prime fields F_p [3]. Twisted Edwards curves $E_{a,d}$ is represented by

$$E_{a,d} : ax^2 + y^2 = 1 + dx^2y^2 \tag{1}$$

where a,$d \in F_p$, $p > 3$ is a prime number, $a \neq d$ and a,$d \neq 0$. The denotation $E_{a,d}(F_p)$ is the set of all points (x, y) where x,$y \in F_p$ that satisfies the above equation $E_{a,d}(F_p)$ is an abelian group under the point addition operation [12]. The identity point for the group is the point (0, 1).

The negative of point $P = (x, y) \in E_{a,d}(F_p)$ is another point $-P = (-x, y) \in E_{a,d}(F_p)$. $E_{a,d}$ has a unified formula for both $P+Q$ and $2P$. Let $P = (x_1, y_1) \in E_{a,d}(F_p)$ and $Q = (x_2, y_2) \in E_{a,d}(F_p)$. Then $P + Q = (x_3, y_3) \in E_{a,d}(F_p)$ can be computed by

$$x_3 = \frac{x_1y_2 + y_1x_2}{1 + dx_1y_1x_2y_2}, y_3 = \frac{y_1y_2 - ax_1x_2}{1 - dx_1y_1x_2y_2}$$

Let $P = (x_1, y_1) \in E_{a,d}(F_p)$. Then $2P = (x_3, y_3) \in E_{a,d}(F_p)$ can be computed by

$$x_3 = \frac{2x_1y_1}{1 + dx_1^2y_1^2}, y_3 = \frac{y_1^2 - ax_1^2}{1 - dx_1^2y_1^2}$$

The projective closure of the twisted Edwards curve $E_{a,d}$ in F_p includes the projective points(X, Y, Z) in F_p satisfying the curve equation

$$aX^2Z^2 + Y^2Z^2 = Z^4 + dX^2Y^2$$

The special points are the infinitely distant points $(1, 0, 0)$ and $(0, 1, 0)$ and therefore we find its singularities at infinity in the corresponding affine components $A^1 := aZ^2 + Y^2Z^2 = Z^4 + dY^2$ and $A^2 := aX^2Z^2 + Z^2 = Z^4 + dX^2$. These points are singular.

Edwards curves are a special class of twisted Edwards curves. Edwards curves E_d are represented by

$$E_d : x^2 + y^2 = 1 + dx^2y^2 \quad (2)$$

where $d \in F_p$ and $d \neq 0, 1$. In other words, Edwards curves are Twisted Edwards curves with the coefficient $a = 1$ [4].

2.2 Expansion of Markov Chain for Integer K

Markov Chain, also known as discrete-time Markov Chain (DTMC), is a special case of the Markov process. It is a random process with Markov properties and Discrete-Time state. In this process, given current knowledge or information, only the current state is used to predict the future. The past (*i.e.* the historical state before the present) is irrelevant for predicting the future (*i.e.* the future state after the present).

Algorithm 1 Expand the Markov Chain for Integer k [29]

Input: k
Output: $MADC(k)$
$i = 1, j = 1$
$a_1 \leftarrow \lfloor \frac{k}{2} \rfloor, a_2 \leftarrow \lceil \frac{k}{2} \rceil$
$m \leftarrow$
for $s \leftarrow \{a_1, a_2\}$ to $\lfloor \frac{a_i}{2} \rfloor \neq 0$, $i = i + 2, j++$ **do**
  if $a_i \bmod 2 \neq 0$ **then**
    $a_{i+2} \leftarrow \lfloor \frac{a_i}{2} \rfloor, a_{i+3} \leftarrow a_i - a_{i+2}, e_j = 0$
  else
    $a_{i+2} \leftarrow \frac{a_i}{2}, a_{i+3} \leftarrow a_{i+1} - a_{i+2}, e_j = 1$
  end if
  $s_i \leftarrow \{a_{i+2}, a_{i+3}\}$
  $m \leftarrow \{m, e_j\}, s \leftarrow \{s, s_i\}$
end for
$max \leftarrow j$
Return s, m
End

At each step of the Markov chain, the system can change from one state to another or stay in the current state according to the probability distribution.

Markov chain is a random process that satisfies the following two assumptions:

1) The probability distribution of the system state at time $t+1$ is only related to the state at time t, independent of the state before time t.

2) The state transition from time t to time $t+1$ is independent of the value of t.

Any integer K can be represented as $\lfloor \frac{k}{2} \rfloor + \lceil \frac{k}{2} \rceil$, the specific algorithm is Algorithm 1. For example, 29 can be turned into: 29 →(14,15)→(7,8)→(3,4)→(1,2), the current state of the chain is only related to the previous state and has nothing to do with the other states, conforming to the Markov chain. The algorithm is defined as markov point addition-point double chain(Markov Addition-Double Chain, MADC).

2.3 Montgomery Ladder Algorithm

The left-to-right Montgomery algorithm [20], which has a point addition and a point doubling in each cycle, is shown as Algorithm 1. Yin and Zhang [30] proposed a formula for calculating only-y-coordinate, ignoring the x coordinate and restoring the x-coordinate in the last step. Montgomery Ladder algorithm executes the point addition and points doubling operation in each cycle, so it can resist the simple power attack very well. Also, the difference between R_0 and R_1 is fixed as P.

Algorithm 2 Montgomery Ladder Algorithm [5]

Input: $P = (x, y)$, $k = (k_{n-1}, k_{n-2}, ..., k_1, k_0)_2 \in N$
Output: $Q = kP \in E(F_p)$
$R_0 = P, R_1 = 2P$
for $i = n - 2$ to 0 **do**
  if $k_i = 1$ **then**
    $R_0 = R_0 + R_1, R_1 = 2R_1$
  else
    $R_1 = R_0 + R_1, R_0 = 2R_0$
  end if
end for
Return $Q = R_0$
End

3 Montgomery Ladder Algorithm Using Only-y-Coordinate

According to the algorithm proposed in the literature, [29] that only-x-coordinate over prime fields, we proposed an algorithm that only-y-coordinate in Edwards curves over prime fields. Using calculating only-y-coordinate idea, we

Algorithm 3 Montgomery Ladder Algorithm Using Only-y-Coordinate

1: Input: $P=(x,y)$, $k=(k_{n-1},k_{n-2},...,k_1,k_0)_2\in N$
2: Output: $Q=kP\in E(F_p)$
3: set $y_1\leftarrow y, y_2\leftarrow \frac{y^2-x^2}{1-dx^2y^2}, x_0,y_0\in E(F_p)$
4: **for** $i=n-2$ to 0 **do**
5: **if** $k=0$ **then**
6: $y_2\leftarrow Maddy(y_1,y_2), y_1\leftarrow Mdoubley(y_1)$
7: **else**
8: $y_1\leftarrow Maddy(y_1,y_2), y_2\leftarrow Mdoubley(y_2)$
9: **end if**
10: **end for**
11: $x_1=Mx_0y_0(x_0,y_0,y_1,y_2)$
12: **Return** $(Q=x_1,y_1)$
13: End

optimize the Montgomery Ladder algorithm on Edwards elliptic curves in the prime field, given as Algorithm 2.

In Algorithm 3, $Maddy$ and $Mdoubley$ are the point addition and point doubling formulas in (3) and (4) by Lemma 1. Mx_0y_0 is the formula in (5) by Lemma 2. In the affine coordinate, the computational cost of point addition and point doubling is $I+2M+2S+2D$ and $I+2S+2D$, and the computational cost of restoring the x-coordinate is $I+3M+1D$. We prove Lemma 1 and 2 by computing only-y-coordinate.

Lemma 1. *Let $P_0=(x_0,y_0)$, $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$, $P_3=(x_3,y_3)\in E_{a,d}(F_p)$, and $P_3=P_1+P_2, P_0=P_2-P_1$. $y_3,x_i(i=1,2,3)$ are undiscovered. Then the y-coordinate of $P_3=(x_3,y_3)$ can be computed as follows.*

If $P_3=P_1+P_2$,

$$y_3=-\frac{dy_1^2y_2^2-y_1^2-y_2^2+1}{y_0(dy_1^2y_2^2-dy_1^2-dy_2^2+1)} \tag{3}$$

If $P_3=2P_1$,

$$y_3=\frac{-dy_1^4-2y_1^2-1}{dy_1^4-2dy_1^2+1} \tag{4}$$

Proof. $P_3=P_1+P_2, P_4=P_2-P_1$, then

$$y_3=\frac{y_1y_2-x_1x_2}{1-dx_1x_2y_1y_2}, y_4=\frac{y_1y_2+x_1x_2}{1+dx_1x_2y_1y_2}$$

Multiplying the above two equations,

$$y_3y_4=\frac{y_1^2y_2^2-x_1^2x_2^2}{1-d^2x_1^2x_2^2y_1^2y_2^2},$$

Replace $x_1^2=\frac{1-y_1^2}{1-dy_1^2}$, $x_2^2=\frac{1-y_2^2}{1-dy_2^2}$,

$$y_3=\frac{d^2y_1^4y_2^4-dy_1^4y_2^2-dy_1^2y_2^4-1+y_1^2+y_2^2}{y_0(-d^2y_1^4y_2^4+d^2y_1^4y_2^2+d^2y_1^2y_2^4-1+dy_1^2+dy_2^2)},$$

By $x^2+y^2=1+dx^2y^2$, thus $-1+y_1^2+y_2^2=dy_1^2y_2^2$ and $1-dy_1^2-dy_2^2=-dy_1^2y_2^2$, then

$$y_3=\frac{dy_1^2y_2^2-y_1^2-y_2^2+1}{-y_0(dy_1^2y_2^2-dy_1^2-dy_2^2+1)},$$

If $P_4=2P_1$, then

$$y_4=\frac{y_1^2-x_1^2}{1-dx_1^2y_1^2},$$

since $x_1^2=\frac{1-y_1^2}{1-dy_1^2}$,

$$y_4=\frac{-dy_1^4+2y_1^2-1}{dy_1^4-2dy_1^2+1}$$

□

Lemma 2. *Let$P_0=(x_0,y_0)$, $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$, $\in E_{a,d}(F_p)$, and $P_0=P_2-P_1$. $y_i(i=1,2)$ are undiscovered. Then the x-coordinate of P_1 can be recovered by*

$$x_1=\frac{y_1y_0-y_2}{x_0-dx_0y_0y_1y_2} \tag{5}$$

Proof. As $P_2=P_4+P_1$, then

$$y_2(1-dx_1x_0y_1y_0)=y_1y_0-x_1x_0,$$

so,

$$x_1(x_0-dx_0y_0y_1y_2)=y_1y_0-y_2,$$

Thus,

$$x_1=\frac{y_1y_0-y_2}{x_0-dx_0y_0y_1y_2}$$

□

4 Co_Z Montgomery Ladder Algorithm Using Only-y-Coordinate

4.1 Montgomery Ladder Algorithm Based on MADC

Algorithm 1 is applied to the Montgomery ladder algorithm ((P, 2P)→(3P, 4P)→(7P, 8P)→(14P, 15P)→29P), and "point addition-point double" operation is performed in each loop, that is, point addition and point doubling operation is computed at the same time in each state, which can resist simple energy attack, as shown in the Algorithm 4.

In Algorithm 4, $Maddy$ and $Mdoubley$ are the point addition and point doubling formulas in (7) by Lemma 3. Among, line $3-4$ of Algorithm 4 represents the output value calculated by Algorithm 1, Lines $5-10$ represents the initial value set, Lines $11-22$ represent the calculation process, and Lines $23-27$ represents the last decision step.

Lemma 3. *Let $P_0=(x_0,y_0)$, $P_i=(X_i,Y_i,Z_i)\in E_{a,d}(F_p)$, $i=1,2,3,4$ where $P_0=P_2-P_1$. $Y_3,Z_3,X_i(i=1,2,3)$ are undiscovered. If $P_3=P_1+P_2, P_4=2P_1$, from Equation (3) the explicit projective formulas are given by*

$$\begin{aligned}\frac{Y_3}{Z_3}&=\frac{(dY_1^2Y_2^2-Y_1^2Z_2^2-Y_2^2Z_1^2+Z_1^2Z_2^2)}{-y_0(dY_1^2Y_2^2-dY_1^2Z_2^2-dY_2^2Z_1^2+Z_1^2Z_2^2)}\\ \frac{Y_4}{Z_4}&=-\frac{-dY_1^4+2Y_1^2Z_1^2-Z_1^4}{dY_1^4-2dY_1^2Z_1^2+Z_1^4}\end{aligned} \tag{6}$$

Algorithm 4 Montgomery Ladder Algorithm Based on MADC

Input: $P=(x,y)$, $k=MADC(k)$
Output: $Q=kP\in E(F_p)$
$MADC(k)=\{\{a_1,a_2\},\{a_3,a_4\},\cdots\{a_i,a_{i+1}\}\}$
$e_1,e_2,e_3,\cdots e_{max}$
set $Q_1\leftarrow a_iP$
if $k \bmod 2\neq 0$ **then**
  $Q_2\leftarrow a_{i+1}P, n\leftarrow\{Q_1,Q_2\}$
else
  $Q_2\leftarrow 2Q_1=2P, n\leftarrow\{Q_1,Q_1\}$
end if
for $j=1$ to $max, j++$ **do**
  $e_j=e_{max+1-j}$
  if $e_j=0$ **then**
    $Q_{2j+1}\leftarrow Maddy(Q_{2j},Q_{2j-1})$
    $Q_{2j+2}\leftarrow Mdoubley(Q_{2j})$
  else
    $Q_{2j+2}\leftarrow Maddy(Q_{2j},Q_{2j-1})$
    $Q_{2j+1}\leftarrow Mdoubley(Q_{2j-1})$
  end if
  $n_i\leftarrow\{Q_{2j+1},Q_{2j+2}\}$
  $n\leftarrow\{n,n_j\}$
end for
if $k \bmod 2\neq 0$ **then**
  $Q\leftarrow Q_{2j+1}+Q_{2j+2}$
else
  $Q\leftarrow 2Q_{2j+1}$
end if
Return Q
End

From Equation (6), the cost of projective y-coordinates addition and doubling formulas is $6M+6S+3D$. If we set $Z_1=1$, then the mixed projective y-coordinates addition and doubling formulas have the total cost $3M+4S+3D$ as follows:

$$\begin{aligned} &A_1=(Y_1^2+Z_1^2), B_1=(Y_1^2-Z_1^2), A_2=(Y_2^2+Z_2^2)\\ &B_2=(Y_2^2-Z_2^2), C=B_1B_2, D=Y_1^4-B_1^2\\ &E=B_1^2-Z_1^4, F=C-Z_1^4\\ &Y_3^{'}=(dY_1^2Y_2^2-Y_1^2Y_2^2+C), Z_3^{'}=dY_1^4+D\\ &Y_4^{'}=-y_0(dF+Z_1^4), Z_4^{'}=dE+Z_1^4 \end{aligned} \tag{7}$$

From Equation (7), the costs of addition and doubling formulas are $2M+5S+2D$ and $1M+2S+2D$, respectively. And, the total cost of the mixed addition and doubling formulas is $3M+7S+4D$. In addition, the cost of following mixed addition and doubling formulas is $3M+5S+4D$. If $P_3=P_1+P_2, P_4=2P_2$, the calculation process and the cost of addition and doubling formulas are the same as Equation (6) and Equation (7).

4.2 Co_Z Montgomery Algorithm Using Only-y-Coordinate

In this section, we use standard projective coordinates to improve Algorithm 3. A point in the affine coordinate (x,y) is written in the form of projection coordinates (X,Y,Z) , where $x=\frac{X}{Z}, y=\frac{Y}{Z}$.

In projective coordinate, the calculation cost of point addition and point doubling is $5M+4S+2D$ and $1M+4S+2D$, and the calculation cost of restoring X-coordinate is $I+7M+S+1D$. After unifying the Z coordinate, the denominator of the formula of point addition and point doubling is unified to common points, and the output Z coordinate is unified. The calculation cost of the optimized formula of Co_Z point addition-point doubling is $7M+5S+2D$, and the calculation cost of restoring the X coordinate is $I+6M+S+1D$.

Our Co_Z Montgomery Ladder Algorithm using only-y-coordinate on Edwards Curves is shown as Algorithm 5. In Algorithm 5, the functions $Mad1(Y_1,Y_2,Z)$, $Mad2(Y_1,Y_2,Z)$ and $MX_4Y_4(Y_1,Y_2,Z)$ are represented as (8), (9) and (10), respectively, by Lemma 4 and 5.

Algorithm 5 Co_Z Montgomery Algorithm Using Only-y-Coordinate

Input: P, $k=(k_{n-1},k_{n-2},...,k_1,k_0)_2\in N$
Output: $Q=kP\in E(F_p)$
set $k\leftarrow(k_{n-1},k_{n-2},...,k_1,k_0)_2$
set $Y_1\leftarrow y(1-dx^2y^2), Y_2\leftarrow y^2-x^2, Z\leftarrow 1-dx^2y^2$
for $i=n-2$ to 0 **do**
  if $k_i=0$ **then**
    $(Y_3,Y_4,Z_3)\leftarrow Mad1(Y_1,Y_2,Z)$
    $Y_1\leftarrow Y_4, Y_2\leftarrow Y_3, Z\leftarrow Z_3$
  else
    $(Y_3,Y_4,Z_3)\leftarrow Mad2(Y_1,Y_2,Z)$
    $Y_2\leftarrow Y_4, Y_1\leftarrow Y_3, Z\leftarrow Z_3$
  end if
end for
$X_1=MX_4Y_4(Y_1,Y_2,Z)$
Return $(Q=(\frac{X_1}{Z},\frac{Y_1}{Z}))$
End

The following lemma gives the method for calculating the Y and Z coordinates. In each cycle of the Montgomery algorithm, the input and output Z coordinates are the same. We use the Co_Z technique to unify the Z coordinates of point addition operation, and the cost of operation is reduced.

Lemma 4. *Let* $P_0=(x_0,y_0)$, $P_i=(X_i,Y_i,Z_i)\in E_{a,d}(F_p)$, $i=1,2,3$ *where* $Z_1=Z_2=Z, P_0=P_2-P_1$. $Y_3,Z_3,X_i(i=1,2,3)$ *are undiscovered. Then Y-coordinate and Z-coordinate of* P_3,P_4 *can be calculated as follows:*

If $P_3 = P_1 + P_2, P_4 = 2P_1$, *then*

$$\begin{cases} Y_3 = (dY_1^2Y_2^2 - Y_1^2Z^2 - Y_2^2Z^2 + Z^4) \\ \quad (dY_1^4 - 2dY_1^2Z^2 + Z^4) \\ Y_4 = -y_0(dY_1^2Y_2^2 - dY_1^2Z^2 - dY_2^2Z^2 + Z^4) \\ \quad (-dY_1^4 + 2Y_1^2Z^2 - Z^4) \\ Z_3 = -y_0(dY_1^2Y_2^2 - dY_1^2Z^2 - dY_2^2Z^2 + Z^4) \\ \quad (dY_1^4 - 2dY_1^2Z^2 + Z^4) \end{cases} \tag{8}$$

If $P_3 = P_1 + P_2, P_4 = 2P_2$, *then*

$$\begin{cases} Y_3 = (dY_1^2Y_2^2 - Y_1^2Z^2 - Y_2^2Z^2 + Z^4) \\ \quad (dY_2^4 - 2dY_2^2Z^2 + Z^4) \\ Y_4 = -y_0(dY_1^2Y_2^2 - dY_1^2Z^2 - dY_2^2Z^2 + Z^4) \\ \quad (-dY_2^4 + 2Y_2^2Z^2 - Z^4) \\ Z_3 = -y_0(dY_1^2Y_2^2 - dY_1^2Z^2 - dY_2^2Z^2 + Z^4) \\ \quad (dY_2^4 - 2dY_2^2Z^2 + Z^4) \end{cases} \tag{9}$$

Proof. Representation in standard projective coordinates with only-Y-coordinates:
If $P_3 = P_1 + P_2, P_4 = 2P_1$, the formula is as follows:

$$\frac{Y_3}{Z_3} = \frac{(dY_1^2Y_2^2 - Y_1^2Z_2^2 - Y_2^2Z_1^2 + Z_1^2Z_2^2)}{-y_0(dY_1^2Y_2^2 - dY_1^2Z_2^2 - dY_2^2Z_1^2 + Z_1^2Z_2^2)}$$
$$\frac{Y_4}{Z_4} = -\frac{-dY_1^4 + 2Y_1^2Z_1^2 - Z_1^4}{dY_1^4 - 2dY_1^2Z_1^2 + Z_1^4}$$

The coordinate points are represented when $Z_1 = Z_2 = Z$ as follows:

$$Y_3 = (dY_1^2Y_2^2 - Y_1^2Z^2 - Y_2^2Z^2 + Z^4)$$
$$Z_3 = -y_0(dY_1^2Y_2^2 - dY_1^2Z^2 - dY_2^2Z^2 + Z^4)$$
$$Y_4 = -dY_1^4 + 2Y_1^2Z^2 - Z^4$$
$$Z_4 = dY_1^4 - 2dY_1^2Z^2 + Z^4$$

The Z coordinate is unified, and the new coordinate changes as follows:

$$\begin{cases} Y_3 = (dY_1^2Y_2^2 - Y_1^2Z^2 - Y_2^2Z^2 + Z^4) \\ \quad (dY_1^4 - 2dY_1^2Z^2 + Z^4) \\ Y_4 = -y_0(dY_1^2Y_2^2 - dY_1^2Z^2 - dY_2^2Z^2 + Z^4) \\ \quad (-dY_1^4 + 2Y_1^2Z^2 - Z^4) \\ Z_3 = -y_0(dY_1^2Y_2^2 - dY_1^2Z^2 - dY_2^2Z^2 + Z^4) \\ \quad (dY_1^4 - 2dY_1^2Z^2 + Z^4) \end{cases}$$

which corresponds to the 'if $k_i = 0$ ' case in Line 6 of Algorithm 5. The addition and doubling formulas are computed cost of (8) is $7M + 5S + 2D$. When $Z_1 = 1$, the addition and doubling formulas are computed cost of (8) is $6M + 4S + 2D$.

If $P_3 = P_1 + P_2, P_4 = 2P_2$, the formula is as follows:

$$\frac{Y_3}{Z_3} = \frac{(dY_1^2Y_2^2 - Y_1^2Z_2^2 - Y_2^2Z_1^2 + Z_1^2Z_2^2)}{-y_0(dY_1^2Y_2^2 - dY_1^2Z_2^2 - dY_2^2Z_1^2 + Z_1^2Z_2^2)}$$
$$\frac{Y_4}{Z_4} = -\frac{-dY_2^4 + 2Y_2^2Z_2^2 - Z_2^4}{dY_2^4 - 2dY_2^2Z_2^2 + Z_2^4}$$

The coordinate points are represented when $Z_1 = Z_2 = Z$ as follows:

$$Y_3 = (dY_1^2Y_2^2 - Y_1^2Z^2 - Y_2^2Z^2 + Z^4)$$
$$Z_3 = -y_0(dY_1^2Y_2^2 - dY_1^2Z^2 - dY_2^2Z^2 + Z^4)$$
$$Y_4 = -dY_2^4 + 2Y_2^2Z^2 - Z^4$$
$$Z_4 = dY_2^4 - 2dY_2^2Z^2 + Z^4$$

To simplify the formula that the Z coordinate is unified. And the new coordinate changes as follows:

$$\begin{cases} Y_3 = (dY_1^2Y_2^2 - Y_1^2Z^2 - Y_2^2Z^2 + Z^4) \\ \quad (dY_2^4 - 2dY_2^2Z^2 + Z^4) \\ Y_4 = -y_0(dY_1^2Y_2^2 - dY_1^2Z^2 - dY_2^2Z^2 + Z^4) \\ \quad (-dY_2^4 + 2Y_2^2Z^2 - Z^4) \\ Z_3 = -y_0(dY_1^2Y_2^2 - dY_1^2Z^2 - dY_2^2Z^2 + Z^4) \\ \quad (dY_2^4 - 2dY_2^2Z^2 + Z^4) \end{cases}$$

which corresponds to the 'else' case in Line 9 of Algorithm 5. The addition and doubling formulas are computed cost of (9) is $7M + 5S + 2D$. When $Z_1 = 1$, the addition and doubling formulas are computed cost of (9) is $6M + 4S + 2D$. □

Lemma 5. *Let* $P_0 = (x_0, y_0)$, $P_i = (X_i, Y_i, Z_i) \in E_{a,d}(F_p)$, $i = 1, 2$ *where* $Z_1 = Z_2 = Z$, $P_0 = P_2 - P_1$. $X_i (i = 1, 2)$ *are undiscovered. Then the* X*-coordinate of* P_1 *can be recovered by:*

$$X_1 = \frac{y_0Y_1Z^2 - Y_2Z^2}{x_0Z^2 - dx_0y_0Y_1Y_2} \tag{10}$$

Proof. From lemma 2,

$$\frac{X_1}{Z_1} = \frac{y_0\frac{Y_1}{Z_1} - \frac{Y_2}{Z_2}}{x_0 - dx_0y_0\frac{Y_1}{Z_1}\frac{Y_2}{Z_2}}$$

Then,

$$X_1 = \frac{y_0Y_1Z^2 - Y_2Z^2}{x_0Z^2 - dx_0y_0Y_1Y_2}$$

which corresponds in Line 14 of Algorithm 5. The calculation cost of (10) is $1I + 6M + S + 1D$. When $Z_1 = 1$, the addition and doubling formulas are computed cost of (10) is $1I + 5M + 1D$. □

5 Efficiency and Analysis

To simplify the cost of scalar multiplication algorithms, we usually ignore the computational cost of addition and subtraction and do not calculate constant multiplication/division. The cost of scalar multiplication can be expressed in terms of field inversions (I), field multiplications (M), and field squaring (S). The usual assumption is that $S = 0.8 \sim 1M$, $I = 8 \sim 10M$. We chose the Edwards curve over prime fields [1] and used standard projection coordinates to eliminate inversion. To simplify

Table 1: Theoretical cost of Algorithms 3, 4 and 5

Algorithms	**Per cycle**	**Recovery and other cost**	**Total cost**
Algorithm 2 [20]	4I+6M+3S	—	l(4I+6M+3S)
Algorithm 3	2I+2M+4S+4D	I+3M+D	l(2I+2M+4S+4D)+(I+3M+1D)
Algorithm 4	3M+7S+4D	I+7M+S+D	$\frac{2l-1}{2}$(2M+5S+2D)+l(1M+2S+2D)
Co_Z Algorithm 5	7M+5S+2D	I+6M+S+D	l(7M+5S+2D)+(I+6M+S+D)

Table 2: Comparison cost of different algorithm at l=160bits

Algorithms	**Total cost**	**l=160bits**
Jacobian-3 [8]	l[(3M+5S)+(11M+5S)]	3584M
ExtJQuartic [8]	l[(2M+5S)+(7M+4S)]	2649.6M
Yu W Algorithm 3 [29]	l(10M+5S)+(2I+12M+S)	2304.2M
Co_ZAlgorithm 5	l(7M+5S)+(I+6M+S)	1808.5M
Algorithm 4	$\frac{2l-1}{2}$(2M+5S)+l(1M+2S)	1417.7M

Table 3: The cost comparison of different bits(S/M=0.84)

Algorithms	**Total cost**	**l=160bits**	**l=192bits**	**l=256bits**	**l=512bits**
Yu W Algorithm 3	l(10M+5S)+(2I+12M+S)	2304.2M	2758.6M	3667.4M	7302.6M
Edwards-formulas (10) [9]	l(4M+6S)	1446.4M	1747.2M	2329.6M	4659.2M
Co_Z Algorithm 5	l(7M+5S)+(I+6M+S)	1808.5M	2166.9M	2883.7M	5750.9M
Algorithm 4	$\frac{2l-1}{2}$(2M+5S)+l(1M+2S)	1417.7M	1701.8M	2270.2M	4543.4M

the comparison over prime fields, we assume $I \approx 9.7M$, $S \approx 0.84M$ [29].

Table 1 shows the theoretical costs of the three algorithms, where l is the binary length of the scalar. The total cost of scalar multiplication is approximately equal to the cost of the binary length of the scalar and the per cycle. Algorithm 3 is expressed in affine coordinates. Algorithm 4 is expressed in projective coordinates, and the integer k is expressed in the Markov chain. Algorithm 5 is expressed in projective coordinates, and the Z-coordinate is unified for the point addition and point doubling.

Table 2 has shown that the total cost of the different algorithms compared at l=160 bits. It can be seen from Table 2 that Co_Z Algorithm 5 is more efficient than Yu W Algorithm 3 when the Z coordinate is unified. In the absence of a unified Z-coordinate, Algorithm 4 is more efficient than Edwards-formulas (10). When S/M=0.84, the cost of Co_Z Algorithm 5 is 21.5% faster than that of Yu W Algorithm 3, the cost of Algorithm 4 is 38.5% faster of Yu W Algorithm 3, and the cost of Algorithm 4 is 2% faster of Edwards-formulas (10).

Both Table 3 and Figure 1 present the cost comparison of Yu W Algorithm 3, Edwards-formulas (10), Co_Z Algorithm 5, and Algorithm 4 on the 160, 192, 256, 512 bits prime fields. The results show that the cost of Co_Z Algorithm 5, and Algorithm 4 is always lower than that of Yu W Algorithm 3. Algorithm 4 is lower than that of Edwards-formulas (10).

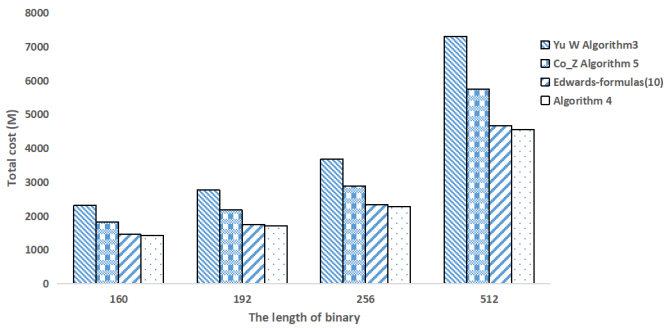


Figure 1: The cost of comparison different length of binary

Figure 1 shows the cost of comparative algorithms with different bits. It can be seen that the cost of Co_Z Algorithm 5, and Algorithm 4 is always lower than that of Yu W Algorithm 3, and the longer the binary length, the greater the difference. All we are using are binary bits at

the security level supported by NUMS.

When the cost of additions/subtractions in a prime field and the recovery of vertical coordinate are considered, the practical cost is slightly different from the theoretical cost. On the 256-bits prime field, the cost of Co_Z Algorithm 5 is 21.4% faster than Yu W Algorithm 3, and Algorithm 4 is 35.9% faster than Yu W Algorithm 3. Algorithm 4 is 2.5% faster than Edwards-formulas (10). In other words, Co_Z Algorithm 5 and Algorithm 4 has a lower cost.

6 Conclusions

This paper studies the Montgomery algorithm for Edwards elliptic curves in a 160, 192, 256, 512-bits field, and compares the cost of the Co_Z Montgomery algorithm with the different elliptic curves scalar multiplication algorithms in the prime number field. In each cycle, our algorithm eliminates the calculation of the x-coordinate, avoids inversions of all fields, and uses a unified Z coordinate in point addition and point doubling. Experiments on the prime field show that the cost of Co_Z Algorithm 5 is 21.4% faster than Yu W Algorithm 3, the cost of Algorithm 4 is 2.5% faster than Edwards-formulas (10). And the cost of Algorithm 4 is 21.3% faster than Co_Z Algorithm 5. The experimental results show that the cost of the Montgomery algorithm is lower on the Edwards curves. In the elliptic curve scalar multiplication algorithm, integer k expansion with Markov chain is faster than binary expansion.

Edwards performs on average 60% faster and uses less area than the Double-and-Add, and that the performance of the SPA resistant strongly unified version of the Edwards, far exceeded its Double-and-Add-Always equivalent. Using one or four ALUs operating in parallel, unified Edwards execution time exceeds the Double-and-Add for an equivalent number of ALUs [2]. The algorithm of unified Z coordinate can be used in the Elliptic curve processor based on FPGA, a lower cost of anti-SPA can be obtained [13].

In the future, if an appropriate integer k expansion can be selected to combine with Formula (7), efficiency will be improved even more, and scalar multiplication will be more suitable for specific hardware environments.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant (NO.62102311), and in part by the Key Research and Development Program of Shaanxi under Grant (NO.2021NY-211). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] R. Abarzua, S. Martinez, V. Mendoza, and N. Theriault, "Same Value Analysis on Edwards Curves," *Journal of Cryptographic Engineering*, vol. 10, pp. 27–48, 2020.

[2] B. Baldwin, R. Moloney, A. Byrne, G. Mcguire, and W. P. Marnane, "A Hardware Analysis of Twisted Edwards Curves for an Elliptic Curve Cryptosystem," *Springer Berlin Heidelberg*, vol. 5453, pp.355–361, 2009.

[3] D. J. Bernstein, P. Birkner, M. Joye, T. Lange and C. Peters, "Twisted Edwards Curves," *Lecture Notes in Computer Science*, vol. 5023, pp. 389–405, 2008.

[4] D. J. Bernstein and T. Lange,"Inverted Edwards Coordinates," in *International Conference on Applied Algebra*, vol.4851, pp. 20-27, 2007.

[5] D. J. Bernstein and T. Lange, "Faster Addition and Doubling on Elliptic Curves," pp. 29–50, 2007.

[6] E. Brier and M. Joye, "Weierstrass elliptic curves and side-channel attacks," *Public Key Cryptography, PKC 2002*, pp. 335–345, 2002.

[7] V. S. Dimitrov, L. Imbert and P. K. Mishra, "The Double-Base Number System and Its Application to Elliptic Curve Cryptography," *Mathematics of Computation*, vol. 77, no. 262, pp. 1075–1104, 2008.

[8] Y. Dou, J. Weng, C. Ma and F. Wei, "Secure and efficient ECC speeding up algorithms for wireless sensor networks," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 21, no. 9, pp. 5665-5673, 2017.

[9] R. R. Farashahi and S. G. Hosseini, "Differential Addition on Twisted Edwards Curves," *Australasian Conference on Information Security Privacy*, vol.10343, pp.366-378, May, 2017.

[10] D. Hankerson, A. J. Menezes and S. A. Vanstone, "Guide to Elliptic Curve Cryptography," *Mathematics of Computation*, 2004.

[11] N. Koblitz, "Elliptic Curve Cryptosystems," in *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[12] K. Kodera, C. M. Cheng, A. Miyaji. "Efficient algorithm to compute odd-degree isogenies between Montgomery curves for CSIDH," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E104.A, pp. 1245-1254, Jan, 2021. DOI:10.1587/transfun.2020DMP0024.

[13] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Energy/Area-Efficient Scalar Multiplication with Binary Edwards Curves for the IoT," *Sensors*, vol. 19, no. 3, pp. 720, Jan, 2019.

[14] A. K. Lenstra and E. R. Verheul, "Selecting Cryptographic Key Sizes," *Journal of Cryptology*, vol. 14, no. 4, pp. 255–293, 2001.

[15] S. G. Liu, R. R. Wang, Y. Q. Li and C. L. Zhai, "An Improved Ternary Montgomery Ladder Algorithm on Elliptic Curves over GF(3^m)," *International Journal of Network Security*, vol. 21, no. 3, pp. 384–391, 2019.

[16] S. G. Liu, H. T. Yao, and F. G. Li, "SPA Resistant Scalar Multiplication on Edwards Curve," *Computer Engineering and Applications*, vol. 53, no. 3, p. 103, 2017. DOI: 10.3778/j.issn.1002-8331.1504-0013.

[17] P. Longa and A. Miri, "Fast and Flexible Elliptic Curve Point Arithmetic over Prime Fields," *IEEE Computer Society*, vol. 57, pp. 289–302, March, 2008.

[18] J. Lopez and R. Dahab, "Fast Multiplication on Elliptic Curves Over GF(2^m) Without Precomputation," pp. 724–724, DOI: 10.1007/3-540-48059-5-27, Aug, 1999.

[19] V. S. Miller, "Use of Elliptic Curves in Cryptography," in *Lecture notes in computer sciences 218 on Advances in cryptology—CRYPTO 85.*, pp.417–426, June. 1986.

[20] P. L. Montgomery, "Speeding the Pollard and Elliptic Curve Methods of Factorisation," *Math. Comp*, vol. 48, no. 177, pp. 243–243, 1987.

[21] S. A. Musa and G. Xu, "Fast Scalar Multiplication for Elliptic Curves over Binary Fields by Efficiently Computable Formulas," *International Conference in Cryptology in India*, vol. 10698, pp. 206–226, 2017.

[22] T. Oliveira, J. Lopez, D. F. Aranha, and F. Rodriguez-Henriquez, "Two is The Fastest Prime: Lambda Coordinates for Binary Elliptic Curves," *Journal of cryptographic engineering*, vol. 4, no. 1, pp. 3–17, 2014.

[23] R. Raveen and Goundar *et al.* "Scalar Multiplication on Weierstrass Elliptic Curves from *Co_Z* Arithmetic," *Journal of Cryptographic Engineering*, vol. 1, pp. 161–176, 2011.

[24] M. Rivain, "Fast and Regular Algorithms for Scalar Multiplication over Elliptic Curves", *IACR Cryptology ePrint Archive*, vol. 2011, p. 338, Jan, 2011.

[25] R. Skuratovskii and V. Osadchyy, "Criterions of Supersinguliarity and Groups of Montgomery and Edwards Curves in Cryptography," *WSEAS TRANSACTIONS ON MATHEMATICS*, vol. 19, pp. 709-722, Mar, 2021.

[26] C. M. Tang, M. Z. Xu, and Y. F. Qi, "Cryptography on Twisted Edwards Curves over Local Fields," *Science China Information Sciences*, vol. 58, pp. 1–15, 2015.

[27] H. Wang, B. Li and W. Yu, "Montgomery Algorithm on Elliptic Curves over Finite Fields of Character Three," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 29, pp. 25-29+36, Oct, 2008. DOI: 10.11897/SP.J.1016.2017.01121.

[28] W. Yu, K. P. Wang, B. Li and S. Tian, "*Co_Z* Montgomery Algorithm on Elliptic Curves Over Finite Fields of Characteristic Three," *Chinese Journal of Computers,* vol. 40, no. 5, pp. 1121–1133, Jul, 2017.

[29] W. Yu, K. P. Wang, B. Li and S. Tian, "Montgomery Algorithm over a Prime Field," *Chinese Journal of Electronics,* vol. 28, pp. 39–44, Jan, 2019. DOI: 10.1049/cje.2018.11.006.

[30] H. L. Zhang, "Research and Design of Fast Algorithm for Scalar Multiplication of Elliptic Curve," M.S. thesis, Dept. Yangzhou Univ., May, 2011.

Biography

Shuang-Gen Liu was born in 1979 and received the PH.D. degree in cryptography form Xidian University in 2008. He is currently an associate professor with the school of cyber security, Xi'an University of Posts and Telecommunications, Xi'an, China. He is a member of the China Computer Federation and a member of the Chinese Association for Cryptologic Research. His recent research interests include cryptography and information security.

Rong Lu was born in 1996. She is a graduate student of Xi'an University of Posts and Telecommunications. She is mainly engaged in the research of the elliptic curve cryptosystem.

MIBFHE: Multi-identity Fully Homomorphic Encryption for Edge Data Sharing and Cooperative Computation

Jiawen Bao[1], Yan Zhang[1], Haiqing Liu[1], Yuancheng Li[1], and Rixuan Qiu[2]
(Corresponding author: Yuancheng Li)
School of Control and Computer Engineering, North China Electric Power University[1]
Beijing 102206, China
Email: ycli@ncepu.edu.cn
State Grid Jiangxi Information & Telecommunication Company[2]
Nanchang 330000, China
(Received Aug. 25, 2021; Revised and Accepted Jan. 12, 2022; First Online Feb. 26, 2022)

Abstract

Unlike a traditional centralized and producer-controlled power grid, the smart grid is a more complicated distributed power system consisting of many resources and applications. Furthermore, vast amounts of data generated by edge devices are collected by different parties in an intelligent grid. Therefore, it is important to enabling data sharing and cooperative computation among different parties to achieve high operation efficiency. This paper proposes MIBFHE, a multi-identity fully homomorphic encryption (MIBFHE) scheme for edge-enabled Smart Grid. In MIBFHE, a preimage sampling algorithm is adopted to ensure the security of multi-party computation (e.g., summation) in edge nodes. In addition, the consortium blockchain and smart contract are used to increase system security further and avoid the dependency on trusted third parties. Additionally, a two-round MPC protocol is proposed. The analysis shows that MIBFHE can meet the security and privacy requirements of the smart grid. The experimental evaluation results demonstrate that our scheme has a better performance than other popular schemes.

Keywords: Blockchain; Edge Computing; Privacy Preservation; Secure Multi-party Computation; Smart Grid

1 Introduction

Edge computing is a new decentralized paradigm that can improve the efficiency and quality of services [18,20]. However, edge computing introduces an intermediate edge layer, so edge devices will lose control of the original data and face some new security and privacy issues. Therefore, it is an important issue to achieve secure data sharing and computation while protecting the privacy of edge nodes.

Aiming at data security sharing and computing in edge computing, we combine blockchain technology and secure multi-party computing and propose a blockchain-based edge data sharing and secure multi-party computation (MPC) scheme. Our scheme can realize privacy protection data calculation in edge computing. The feature of computing tasks at the edge layer is that there are multiple distributed edge nodes. This scheme can directly process and securely calculate the ciphertext data under the premise of protecting the edge nodes' privacy and data security to ensure the edge data's security analyzes statistics.

Since blockchain is not suitable for large-scale data storage, analysis and statistics, edge data generated by edge devices are stored in a combination of on-chain and off-chain. Based on this storage model, we propose a secure computing model for edge data. In this scheme, blockchain technology is used to ensure the verifiability of calculations, and MPC technology is used to safeguard the confidentiality of input data during the calculation process. The contributions of this paper are as follows:

1) Propose a blockchain-based edge data sharing and secure multi-party computing model, and build a distributed edge server framework to realize parallel computing operations on ciphertext data.

2) Construct a multi-identity homomorphic encryption scheme, and expand this construction, construct a two-round MPC protocol to realize secure homomorphic computing while protecting the privacy of edge nodes.

3) Construct a novelty MPC agreement based smart contracts to achieve a joint secure computation between edge nodes.

The rest of the paper is organized as follows. Section 2 reviews related work. The proposed model algorithm formulation is provided in Section 3. In Section 4, we discuss the experiment results and analyze the security of the proposed scheme. Finally, Section 5 concludes our work.

2 Related Work

MPC is a multiparty computation algorithm that protects data security and privacy. It solves the problem of collaborative computing that protects privacy between a group of untrusted parties. MPC can ensure some features such as the independence of input, the correctness of computation and decentralization. At the same time, ensure that each input value is not disclosed to other members of the calculation [24].

MPC has always been a research hotspot in cryptography because it can solve the privacy problem of collaborative computation. In 1982, Turing Prize winner Yao Qizhi first proposed the "secure two-party computation" in the classic millionaire problem [22]. Goldreich *et al.* extended the two parties to multi-party computing [9, 10, 23]. [2, 3, 7, 19] studied MPC schemes based on secure sharing, but these schemes all focus on improving the performance of MPC. [6] studied a distributed secure multi-party computation technology for privacy protection data analysis.

MPC is not a single technology, and it can be seen as a combination of cryptography and computing problems in different application scenarios. It is composed of cryptographic primitive, including commonly used hash functions, zero-knowledge proofs, homomorphic encryption (HE) technology, etc. It also includes basic tools in MPC: secret sharing (SS), oblivious pseudorandom function (OPRF), oblivious transfer (OT), etc. Homomorphic encryption is an important tool for constructing secure multi-party computing protocols, and many scholars have carried out the research. López-Alt *et al.* [15] constructed the first multi-key fully homomorphic encryption (MKFHE) based on NTRU and used it to design an MPC protocol. However, the size of a public key was large, which led to high computing costs in the secure multi-party computing environment. The identity-based fully homomorphic encryption (IBFHE) scheme solved the above problems. Gentry, Sahai and Waters proposed the first (leveled) IBFHE scheme. However, their method only worked in a single-identity setting [8]. Based on the research of Gentry *et al.*, Clear *et al.* [5] proposed the first multi-identity setting scheme based on a well-established assumption such as standard learning with errors (LWE), but the key structure of these schemes was complicated. Carpov *et al.* [4] constructed a general MPC protocol with only two rounds of interaction in the common random string (CRS) model. In this paper, the author introduces MKFHE based on LWE to allow one-round distributed decryption of multi-key ciphertext. The key structure of the above schemes is relatively complex, and some schemes rely heavily on bootstrap conversion technology, which requires bit-by-bit encryption when publishing the private key in the public key, which is very cumbersome. The trapdoor function of the scheme proposed in the literature [13, 21] is not efficient enough. The computation of the trapdoor generation algorithm and preimage sampling algorithm is complex, and the generated key's scale is large. The efficiency of the IBFHE algorithm largely depends on the preimage sampling during trapdoor and key generation. So we choose the efficient MP12 trapdoor function [1] to construct the multi-identity fully homomorphic encryption (MIBFHE) scheme. Furthermore, blockchain technology is introduced to solve the data security problems in joint computing.

MPC is the interaction between participants under a specific agreement. The purpose is to obtain the calculation result while ensuring the confidentiality of the input. MPC emphasizes the secrecy of input data but cannot guarantee that the data is verifiable. Blockchain [16] technology can verify the correctness of calculations completed jointly to achieve unanimous recognition of results and prevent tampering with result records. The combination of blockchain technology and MPC can solve more data security and privacy issues. The advantages of blockchain are a digital signature, non-tampering, traceability, and decentralization. And the benefits of MPC are input privacy, calculation accuracy and decentralization. Combining the two techniques can realize decentralization, data protection, joint computing, which can solve more data security and privacy issues.

The application of blockchain technology in MPC is still in its infancy. ZCash added the function of protecting transaction privacy to Bitcoin through zero-knowledge proof. [11, 14] built a secure multi-party secure computing protocol in the Bitcoin system to ensure currency transactions' security and privacy. [12] proposed a smart contract framework based on MPC, standardized the execution of smart contracts based on SMPC and ensured the confidentiality of inputs' correctness calculations in executing the smart contract.

3 System Model and Algorithm Formulation

3.1 System Model

To ensure the privacy of edge nodes, we propose to build an edge data secure computing model based on blockchain. This model uses a multi-identity encryption algorithm to decentralize the management of keys, and perform homomorphic calculations on data using different keys. It can realize the homomorphic computation of data using different keys. Because of the blockchain characteristics, if a large amount of data is directly stored in the blockchain, it will waste a lot of computing power, so the proposed scheme uses a combination of on-chain and off-chain storage. Our solution stores the data encrypted

at the edge in the edge server, and at the same time, stores the summary information in the blockchain. This model allows edge devices to outsource their encrypted data to edge servers, and use secure multi-party computing methods to analyze and calculate edge data, so that ciphertexts can be directly used for data processing and secure computing. Edge nodes share data for collaborative calculations and send the calculation results to the inquirer. Our scheme uses a permissioned blockchain, which can eliminate the incentive mechanism's resource consumption mining process.

As shown in Figure 1, the model includes the edge device layer, blockchain network layer, and edge nodes layer.

1) The edge device layer comprises a large number of edge devices and is responsible for collecting data.

2) The blockchain network layer consists of the application layer, secure computing layer, contract layer, and consensus layer. The secure computing layer mainly implements MPC and secret sharing. The MPC algorithm is called when performing smart contract computation tasks. The contract layer encapsulates the smart contract and its scheduling algorithm. The smart contract interface is the bridge between the smart contract layer and the consensus layer. The application interface is the bridge between the application and the secure computing layer, which can connect the communication and interaction of all parties.

3) The edge nodes layer comprises edge servers with storage space and computing capabilities, and the edge servers can store data collected by edge nodes.

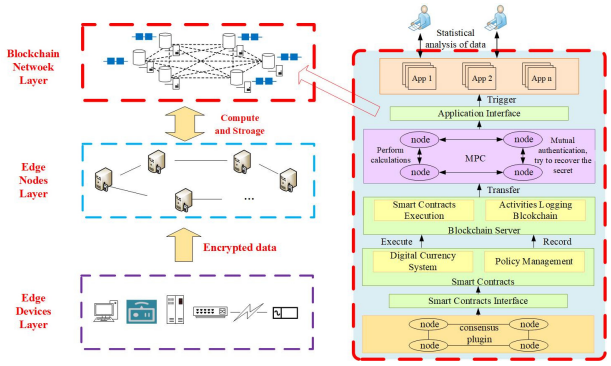


Figure 1: System model

To ensure the privacy of edge nodes, we construct a MIBFHE algorithm, which decentralizes the management of keys and can perform homomorphic calculations on data using different keys. Furthermore, we construct a two-round MPC protocol in the CRS model. Finally, to achieve a fair MPC protocol, we use smart contract technology to add an economic penalty mechanism and call the MPC protocol. To ensure data security during transmission and calculation, edge devices use multi-identity leveled FHE algorithms for data encryption on the device side. When the calculation requester sends a calculation request, the smart contract will automatically perform the following operations, implementing the call to the MPC algorithm. The blockchain uses a consensus mechanism to verify the participating nodes to ensure mutual trust between nodes. After verification, the participants share the encrypted ciphertext data. The participants use the two rounds of MPC protocol we constructed to calculate the ciphertext directly. The reconstructing party receives the received calculation result and reconstructs it to obtain the correct calculation result finally.

3.2 MIBFHE Scheme

This section presents a multi-identity leveled FHE scheme based on LWE. As shown in Figure 2, the edge devices encrypt the data on the device side and then sends it to the edge server to secure computing. This scheme is based on the improvement of multi-identity FHE [17]. We use the efficient MP12 trapdoor generation algorithm and preimage sampling algorithm to complete the Setup and KeyGen stages. Then, combine with the dual LWE encryption algorithm to complete the encryption and decryption operation. Besides, this scheme introduces a "masking system", which can prevent an attacker without a key from obtaining plaintext data through encrypted information $\mathcal{U} \in \{0,\ 1\}^*$. The "masking system" is composed of a pair of algorithms (UniEnc, Extend), which is a very important link in the fully homomorphic encryption scheme. It can hide plaintext (μ) by a pair of values ($\mathcal{U}$, $\mathcal{C}$). This scheme can perform multiple ciphertext homomorphic calculations on different ciphertexts corresponding to different identity id. We record the proposed multi-identity hierarchical fully homomorphic encryption scheme as ξ.

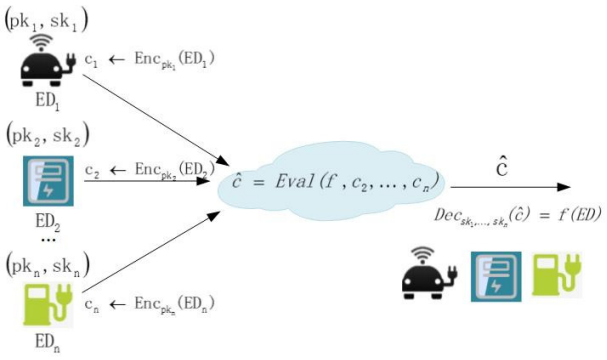


Figure 2: MIBFHE

Where, ED_i are different edge devices, (pk_i, sk_i) are the public and private keys corresponding to ED_i, c_i is the encrypted ciphertext, and $\hat{c}$ is the ciphertext after the homomorphic calculation of the f function.

Lemma 1. (**MP12 TrapGen**$(1^n, 1^m, q,\ H)$[16]). *Input parameters* $n,\ q,\ m$, *let* $n \geq 1, q \geq 2, m = O(n \log_2 q)$, *an invertible matrix* $H \in \mathbb{Z}_q^{n\times n}$). *Output a matrix* $A = \left[\bar{A}|HG - \bar{A}R\right] \in \mathbb{Z}_q^{n\times m}$ *and a trapdoor* $T_A \in \mathbb{Z}^{(m-nk)\times nk}$.

Lemma 2. (**MP12 Sample**$(T_A, A, H, u,\ \sigma)$). *Input* $A \in \mathbb{Z}_q^{n\times m}$, $T_A \in \mathbb{Z}^{(m-nk)\times nk}$, *an invertible matrix* $H \in \mathbb{Z}_q^{n\times n}$, *a random uniform vector* $u \in \mathbb{Z}_q^n$, *Gaussian parameter* $\sigma \geq s_1(T_A)\cdot\omega(\sqrt{\log_2 n})$, *extract.e* $\leftarrow D_{\mathbb{Z}^m,\ \sigma}$ *from negl(n) distribution, so that* $A \bullet e = u$.

Lemma 3. (**MP12 DelTrap**$^{\mathbf{O}}(A' = [A|A_1]\ ,\ H',\ s')$). *Input* $A' = [A|A_1] \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times nk}$, *an invertible matrix* $H \in \mathbb{Z}_q^{n\times n}$. s' *is the parameter of the oracle machine* $\mathcal{O}$ *discrete Gaussian sampling at* $\Lambda = \Lambda^{\perp}(A)$, $s' \geq \eta_{\epsilon}(\Lambda)$. *Output a trapdoor* $R' \in \mathbb{Z}_q^{n\times nk}$, *so that* $AR' = H'G - A_1$.

A MIBFHE scheme is a tuple of polynomial time algorithms, ξ = (*Setup*, *KeyGen*, *Encrypt*, *Expand*, *Eval*, *Decrypt*), The parameter setting is: a security parameter λ, $m = O(n \log q)$, $n = n(\lambda)$, $q = q(\lambda)$, $\bar{m} = m - nk$, LWE error distribution $\chi = \chi(\lambda)$, $\ell_q = \lfloor \log q \rfloor + 1$, matrix $A \in \mathbb{Z}_q^{n\times m}$, lattice $\Lambda^{\perp}(A) = \{x \in \mathbb{Z}^m : A \bullet x = 0\ mod\ q\}$, matrix $G = I_n \bigotimes g^T \in \mathbb{Z}^{n\times nk}$, while I_n is a unit matrix, $g^T = [1,\ 2,\ \ldots,\ 2^{k-1}] \in \mathbb{Z}_q^k$.

ξ. *Setup*$(1^{\lambda},\ 1^{\mathcal{L}},\ 1^{\mathcal{D}})$**:** Input a security parameter λ, a circuit depth to support $\mathcal{L}$ and the number of distinct identities $\mathcal{D}$ that can be tolerated in an evaluation. Output$(pp,\ msk) \leftarrow Setup(1^{\lambda},\ 1^{\mathcal{L}},\ 1^{\mathcal{D}})$.

1) Run *TrapGen* algorithm, generate a matrix $A_0 \in \mathbb{Z}_q^{n\times m}$ and a corresponding trapdoor $R_0 \in \mathbb{Z}^{(m-nk)\times nk}$.
2) Run again *TrapGen* algorithm, generate $\mathcal{D}$ matrices $(A_1, \ldots,\ A_{\mathcal{D}} \in \mathbb{Z}_q^{n\times m})$, choose a random uniform vector $u \in \mathbb{Z}_q^n$. Output: $pp = (A_0,\ A_1, \ldots,\ A_{\mathcal{D}}, G,\ u)$, $msk = R_0$.
3) Use the following multi trapdoor derive algorithm to obtain trapdoor matrix R_{ℓ} of the identity id_{ℓ}.
4) $H_{id_{\ell}}$ is a mapping matrix of the id_{ℓ} through FRD function. Let $A_{id_{\ell}} = [A_{id_{\ell-1}}||A_{\ell} + H_{id_{\ell}}G]$. Run $R_{\ell} \leftarrow DelTrap^{\mathcal{O}}(A' = [A_{id_{\ell-1}}||A_{\ell} + H_{id_{\ell}}G], H_l, s')$, generate $A_{id}R_{\ell} = H_{\ell}G - (A_{\ell} + H_{id_{\ell}}G)$.

ξ. *KeyGen*$(msk,\ id)$**:** Input a master secret key msk and an identity id, output a secret key sk_{id} for id.

1) Run Sample algorithm, generate e.
2) Let $\bar{s} \leftarrow (1, -e) \in \mathbb{Z}_q^{m+1}$, output $sk_{id} = Powersof2\,(\bar{s}) = (Powersof2_p\,(1), -\frac{p}{q}owersof2_q\,(e))$

ξ. *Encrypt*$(pp,\ id,\ \mu_{\mathcal{D}})$**:** Input data, public parameters pp and an identity id. Output $c_i \leftarrow Encrypt(\mu_{\mathcal{D}_i},\ pp,\ id_{j_i}), d \leq \mathcal{D}, j_1, \ldots, j_{\ell} \in [d], i \in [\ell]$.

Run the masking scheme UniEnc$(pp,\ id,\ \mu_{\mathcal{D}})$, output a pair of values$(\mathcal{U},\ \mathcal{C})$, where: $\mathcal{C} \in \mathbb{Z}_c^{[(d+1)n+1]\times m}, \mathcal{U} \in \{0,\ 1\}^*, c = \mathcal{U},\ \mathcal{C})$.

1) Select a random matrix $R \longleftarrow \{0,\ 1\}^{m\times m}$, Let A = pp, $\mathcal{C} = AR + \mu_D G$.
2) Encrypt each element of R to get m^2 ciphertext $V^{(i,\ j)}$, let $\mathcal{U} = \left(\mathbb{V}^{(1,1)}, \ldots, \mathbb{V}^{(m,m)}\right) \in \left(\mathbb{Z}_c^{[(d+1)n+1]}\right)^{m^2}$.

ξ. *Expand*$((id_1, \ldots, id_N)$**,** i**,** $c)$**:** Input identity set, a identity id_i and ciphertext$c$. Output expand ciphertext $\widehat{c_i} \leftarrow Expand((id_1, \ldots, id_N),\ id_i,\ c_i),\ i \in [\ell]$ for id.

1) For $\forall j \in \{id_1, \ldots, id_N\}\backslash\{i\}$, run $X_j \leftarrow Extend(pp, \mathcal{U}, id_i, id_j)$.
2) Then can get a matrix $\hat{C}$ composed of d^2 submatrix, sub-matrix $C_{a,\ b} \in \mathbb{Z}_c^{[(d+1)n+1]\times m}$ is defined as follow:

$$C_{a,\ b} = \begin{cases} c & , a = b \\ \mathbb{X}_j, a = i \neq j \quad and \quad b = j \\ 0^{[(d+1)n+1]\times m}, otherwise \end{cases}$$

3) Finally, output $\hat{c} = \hat{C}$.The structure of the expanded ciphertext $\widehat{C_i}$for id_i is shown as:

$$\widehat{C_i} = \begin{bmatrix} C & 0 & \cdots & 0 & 0 \\ 0 & C & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_1 & \cdots & C & \cdots & X_d \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & C \end{bmatrix}$$

ξ. *Eval*$(pp,\ C,\ (\widehat{c_1},\ \ldots,\ \widehat{c_{\ell}}))$**:** Input data, public parameters pp, boolean circuit C: $depth(C) \leq \mathcal{L}$ and ℓ expanded ciphertext $(\widehat{c_1},\ \ldots,\ \widehat{c_{\ell}})$. Output $\hat{c} = Eval(pp, C, (\widehat{c_1},\ \ldots,\ \widehat{c_{\ell}}))$. Where, ξ.Eval algorithm includes ξ.AddEval and ξ.MultEval.

ξ.AddEval: output $\hat{C}^{+} = \hat{C}_1 + \hat{C}_2$;

ξ.MultEval: output $\hat{C}^{\times} = \hat{C}_1 G^{-1}(\hat{C}_2)$.

ξ. *Decrpty*$((sk_{id_1}, \ldots,\ sk_{id_N}),\ c)$**:** For $sk_i \leftarrow KeyGen(msk,\ \mathrm{id}_i),\ \forall i \in [\ell],\ Decrypt((sk_{id_1}, \cdots, sk_{id_N}), \hat{c}) = C(\mu_{D_1}, \cdots, \mu_{D_{\ell}})$.

The threshold distributed multi-identity decryption scheme ξ. *Decrpty* consists of two algorithms: ξ. *PartDec* and ξ. *FinDec*. The specific described as follows.

1) ξ. *PartDec*$(\hat{c},\ (id_1, \ldots,\ id_N),\ k,\ sk_{id_k})$: Input an expanded ciphertext $\hat{c} = C$, which consisting of sub-matrices $\hat{C} = \left[\widehat{C_1} \quad \cdots \quad \widehat{C_N}\right]^T$. Define vector $\hat{w} = [0,\ \ldots,\ 0,\ \lceil q/2 \rceil]$. Compute

$\gamma_i = t_i\widehat{C_i}\hat{G}^{-1}(\hat{w}^T)$. Output partial decryption $ED_i = \gamma_i + e_i^{sm}$.

Where, t_i is the private key sk_{id_i} corresponding to identity id_i.$MatrixG \in \mathbb{Z}_q^{n\times m}$, $GG^{-1}(M) = M$, $\hat{G}$ is the expaned matrix of the matrix G. $e_i^{sm} \longleftarrow [-2^{d\lambda log\lambda}B_\chi, 2^{d\lambda log\lambda}B_\chi]$ is some small random noise.

2) ξ. $FinDec(ED_1, \ldots, ED_N)$: Input N partial decryption ciphertext ED_i, comput $p = \sum_{n=1}^{N} ED_i$, Output $\mu_D = \left|\left\lceil \frac{p}{q/2} \right\rfloor\right| \in \{0,1\}$

3.3 Two-round MPC Protocol

In this section, we present a two-round MPC protocol, using the threshold multi-identity fully homomorphic scheme.The scheme proposed in this section consists of five polynomial-time algorithms, includingξ.Setup (initialization algorithm),ξ.KenGen (key generation algorithm), ξ.Encrypt (encryption algorithm), ξ.Eval (ciphertext homomorphic algorithm)), ξ.Dec (decryption algorithm). The data provider uses the key to encrypt the data. After homomorphic calculation, the requester can obtain the ciphertext calculation result, and decrypt it with its secret key to obtain the final calculation result. The protocol, given in Figure 3.

Initialize: Define the compute function $\mathcal{F}: (\{0,1\}^{\ell_{in}})^N \to \{0,1\}^{\ell_{out}}$. Let $\mathcal{L}$ be the depth of the circuit, the number of distinct identities $\mathcal{D}$ that can be tolerated in an evaluation

Setup: Run Setup $\leftarrow \xi$. Setup$(1^\lambda, 1^\mathcal{L}, 1^\mathcal{D})$. All the EDs share the common Setup.

Input: Each ED ED_k has input $x_k \in \{0,1\}^{\ell_{in}}$.

The Protocol:

Round I. Each ED ED_k executes the following steps.

- Generate a secret key $sk_k \leftarrow \xi$. KeyGen(Setup) for identity id_k.
- Encrypt the data bit-by-bit:

$$\{c_{k,j} \leftarrow \xi. Encrypt(pp, id_k, x_k[j])\}_{j\in[\ell_{in}]}.$$

- Broadcast the ciphertexts $\{c_{k,j}\}_{j\in[\ell_{in}]}$.

Round II. Each ED ED_k on receiving value $\{c_{i,j}\}_{i\in[N]\setminus\{k\},j\in[\ell_{in}]}$ executes the following steps.

- First expand each $c_{i,j}$:

$$\{\widehat{c_{i,j}} \leftarrow \xi. \mathrm{Expand}((id_1, \ldots, id_N), i, c_{k,j})\}_{i\in[N],j\in[\ell_{in}]}$$

- Then run evaluation algorithm to generate the evaluated ciphertext:

$$\{\hat{c}_j \leftarrow \xi. \mathrm{Eval}(\mathcal{F}_j, (\hat{c}_{1,1}, \ldots, \hat{c}_{1,\ell_{in}}))\}_{j\in[\ell_{out}]}$$

Where: $\mathcal{F}_j$ is the Boolean function for j-th bit of the output of $\mathcal{F}$.

- Finally, all the EDs concurrently take party in one-round threshold decryption to obtain the output data bit-by-bit as follows:
  - Each ED_k computes the partial decryption for all $j \in [\ell_{out}]$:

$$\mathrm{ED}_k^{(j)} \leftarrow \xi. \mathrm{PartDec}(\hat{c}_j, (id_1, \ldots, id_N), k, sk_k)$$

  - ED_k broadcasts all the values $\{\mathrm{ED}_k^{(j)}\}_{j\in[\ell_{out}]}$

Output: On receiving all values $\{\mathrm{ED}_k^{(j)}\}_{i\in[N],j\in[\ell_{out}]}$ run the final decryption to obtain the j-th output bit: $\{y_i \leftarrow \xi. \mathrm{FinDec}(\mathrm{ED}_1^{(j)}, \ldots, \mathrm{ED}_N^{(j)})\}_{j\in[\ell_{out}]}$. Output $y = y_1 \ldots y_{\ell_{out}}$.

Figure 3: MPC protocol

3.4 Smart Contract

Our scheme implements secure collaborative computing through the invocation of smart contracts. The smart contracts are pre-programmed and deployed on the blockchain. The requester initiates a request, which will trigger the smart contract to perform corresponding operations automatically. The detailed smart contract pseudocode is shown in Algorithm 1, and the flowchart is shown in Figure 4. The entire smart contract execution process includes four steps: initialization, encrypted data sharing, homomorphic computing, verification, and decryption. All nodes participating in the calculation need to deliver a certain deposit in the initialization phase, and the smart contract verifies their identity. If the node is legitimate, proceed to the next step; otherwise, the deposit is deducted, and the algorithm execution is stopped. After the node identity verification is passed, the model invokes the MPC algorithm of the computing layer to perform the calculation task and sends the calculation result to the requester.

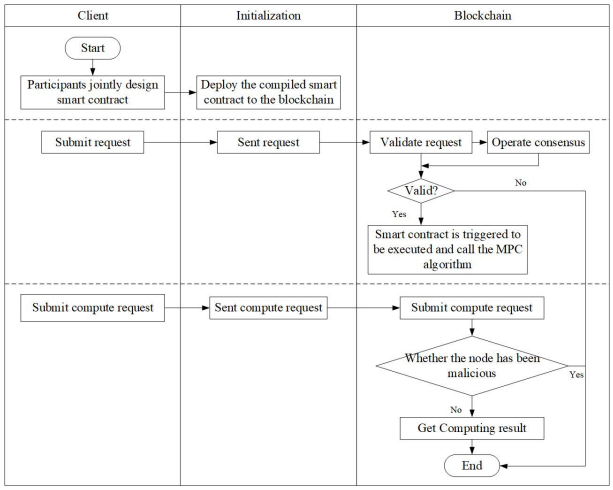


Figure 4: Program flow

4 Evaluations

4.1 Security Analysis

In this section, we analyze the security of our scheme. Firstly, we use Game-based security proof to prove the security of the proposed MIBFHE scheme. We further analyze the security of our architecture.

Definition 1. *(LWE). For a positive integer n and the related q=q(n), let χ be the error distribution on $\mathbb{Z}_q$, and s be a variable on $\mathbb{Z}_q$. Define the probability distribution corresponding to $\mathbb{Z}_q^n \times \mathbb{Z}_q$ according to the above conditions is $A_{s,\chi}$.$A_{s,\chi}$ needs to meet the condition of $(a, a^T \cdot s+x)$, where x is a value on χ and the variable a is uniformly distributed on $\mathbb{Z}_q^n$. All relevant operations are valid modulo q. The ultimate goal of the fault-tolerant learning problem*

Algorithm 1 MPC smart contract algorithm

Begin
Initialize :Define the compute function $\mathcal{F}$: $(\{0,\ 1\}^{\ell_{in}})^{N} \rightarrow \{0,\ 1\}^{\ell_{out}}$. Let $\mathcal{L}$ be the depth of the circuit, the number of distinct identities $\mathcal{D}$ that can be tolerated in an evaluation
The set of participants to the protocol $ED = \{ED_1, \ldots, ED_N\}$
EDs pay deposit to blockchain
The set of malicious parties M = $\emptyset$, the number of malicious parties m=0
Deposit array P
Validation array V
Input :Each $EDED_k$ has input $x_k \in \{0,\ 1\}^{\ell_{in}}$
Output :$y = y_1 \ldots y_{\ell_{out}}$
function Deposit( )
if P[1] & P[2] & ... & P[n] = 1, D = 1 **then**
  continue
else if D $\neq$ 1 **then**
  return deposit
  break
end if
end function
function MPC( )
Each ED ED_k executes the Protocol
Each ED ED_k broadcasts a partial decryption of the output $\{ED_k{}^j\}_{j\in[\ell_{out}]}$
if verified & the secret is recovered **then**
  $V_k\,[0] = 1$
  $V = V_1 \& V_2 \& \ldots \& V_n$
  if $V\,[0] = 1$ **then**
    output y
  else if $V\,[0]! = 1 \& \& V\,[i]! = 1$ **then**
    $ED_i \in M$
  end if
end if
end function
End

LWE$_{m,\ n,\ q,\ \chi}$ *is to find s with a large enough probability given m independent samples on* $A_{s,\ \chi}$. *The goal of the* D LWE$_{m,\ n,\ q,\ \chi}$ *problem is to determine which distribution these samples are taken from with a non-negligible probability.*

Theorem 1. *There exists a MIBFHE scheme is IND-sID-CPA (indistinguishable security under selective plaintext and selective id attacks) secure under the assumption that the hardness of LWE.*

Proof. The theorem is proved using a game-based proof approach. Assuming that there is a polynomial-time attacker $\mathcal{A}$, The advantage of $\mathcal{A}$ in Game is defined by $Adv_{Game}[\mathcal{A}]$.

Game 0: Game 0 is a game between an attacker and an IND-sID-CPA challenger built on an attack plan.

Game 1: The goal of challenger $\mathcal{C}$ is to generate the challenge ciphertext c^* from id^*. The difference between Game 1 and Game 0 is that Game 1 changes the generation method of $A_1, \ldots,\ A_{\mathcal{D}}$ in pp, and directly selects uniformly and randomly from the matrix $R_1^*, \ldots,\ R_{\mathcal{D}}^*$, let $A_i = [-H_{id_i^*} \cdot G - A_0 R_i^*]$. Attacker $\mathcal{A}$ is unable to distinguish between Game 0 and the modified Game 1. Thus

$$|Adv_{Game1}[\mathcal{A}] - Adv_{Game0}[\mathcal{A}]| = 0$$

Game 2: The difference between Game 2 and Game 1 is that Game 2 changed the key generation method. Game 2 uses the $TrapGen(1^n, 1^m, q,\ H)$ algorithm to generate the trapdoor matrix R_G. Challenger $\mathcal{C}$ run the Sample algorithm to output the private key through $trapdoor R_G$, completing the query response to the attacker $\mathcal{A}$'s identity private key. The probability of attacker $\mathcal{A}$ distinguishing Game 2 from Game 1 is less than its advantage in solving the DLWE$_{m,\ n,\ q,\ \chi}$ problem:

$$|Adv_{Game2}[\mathcal{A}] - Adv_{Game1}[\mathcal{A}]| \leq Adv_{D\text{LWE}_{m,\ n,\ q,\ \chi}}[\mathcal{A}]$$

Game 3: In Game 3, the challenge ciphertext c^* given by challenger $\mathcal{C}$ is no longer generated by the encryption algorithm, but directly selected from the ciphertext space $\mathbb{Z}_q \times \mathbb{Z}_q^{\ell \times nk+m}$ uniformly and randomly. Under the premise that the hardness of solving the DLWE problem, the next step is to prove that Game 3 is statistically indistinguishable from Game 2 for probabilistic polynomial-time (PPT) adversaries. Challenger $\mathcal{C}$ selects a series of samples $(t_i,\ u_i),\ i = 0,\ 1,\ \ldots, \bar{m}$, attacker $\mathcal{A}$ submits message $b^* \in 0,\ 1$. Let $t^* = [t1 \ldots tm]T$. Hide the plaintext message by the following method, $c_0^* = t_0 + b^* \cdot \lfloor q/2 \rfloor, c_1^* = \begin{bmatrix} t^* \\ (-\bar{R}_k^*)^T t^* \end{bmatrix}, \bar{R}_k^* = [R_1^* || \ldots || R_k^*]$. *At this time,* $A_{id}^* = [\bar{A} || -\bar{A}R],\ t^* = \bar{A}^T s + y, therefore, c_1^* = (A_{id}^*)^T s + \begin{bmatrix} y \\ -\bar{R}_k^{*T} y \end{bmatrix}$, the right side actually is the randomly selected portion of c_1 from $\mathbb{Z}_q \times \mathbb{Z}_q^m$ space. Since $t_0 = u_0{}^T s + x$, then $c_0^* = u_0{}^T s + x + b^* \cdot \lfloor q/2 \rfloor$, actually is the randomly selected portion of c_0 from $\mathbb{Z}_q \times \mathbb{Z}_q^m$. Therefore, the challenge ciphertext has the same distribution as Game 3. Thus, the advantage of attacker $\mathcal{A}$ in Game 3 is 0, that is, $Adv_{Game\ 3}[\mathcal{A}] = 0$.

Therefore, under the assumptions of DLWE$_{m,\ n,\ q,\ \chi}$, the advantage of Adv[$\mathcal{A}$] can be ignored. Theorem 1 is proved. □

It has been proven the MIBFHE scheme is secure. In the following, we further analyze the security of our architecture, and how our architecture defends against some attacks.

Security: The consortium blockchain guarantees the security of our architecture. The communication security and identity authentication of the consortium

blockchain reduces the risk of remote hacking attacks. The blockchain can ensure that the computation results are verifiable. Blockchain is tamper-proof and traceable. The information written on the blockchain cannot be tampered with, which also guarantees security.

Privacy: The attacker may try to discover the true identity of the node by correlating the different data from anonymous nodes, thus compromising the privacy of the node. Our scheme uses a multi-identity fully homomorphic encryption scheme and consortium blockchain to ensure the privacy of nodes. Only nodes authorized by Hyperledger fabric can join the network and share data between trusted nodes.

The possible attacks on communication involving important data in the network mainly include Man-in-the-Middle Attack (MITM), Forgery attack and Distributed Denial of Service (DDoS). The analysis of resistance to these attacks is as follows:

1) DDoS: DDoS attacks are relatively common attacks. Attackers may try to eavesdrop on the identity verification results, sign them with false signatures and pass the verification information to the blockchain. Even if the authentication result is valid, the tainted message will be regarded as an unreliable message, and the transaction will be discarded. Only when many nodes are destroyed and controlled, DDoS attacks can be realized, which is difficult.

2) Forgery attack: Attackers may try to forge the identity of the terminal device or edge node to obtain data information. Assuming that the attacker steals the signature of a message, it is computationally difficult for the attacker to forge a legitimate signature. On the other hand, if the attacker submits a request with a false signature, it will be rejected. Mutual authentication between nodes in our scheme can effectively prevent forgery attacks.

3) MITM: MITM attacks generally occur during data transmission between terminal devices and edge nodes. Attackers may try to eavesdrop, intercept, or manipulate data information sent by terminal devices and edge nodes. The attacker needs to forge the identity of the terminal device or the edge node. If the above forgery attack is blocked, a valid MITM attack cannot be carried out.

4.2 Performance Evaluation

Table 1 shows the comparison between our scheme and several schemes. Compared with GSW13, the GSW13 scheme can only support the ciphertext homomorphic computation in the single-identity environment. Our scheme solves this problem in GSW13. Our scheme extending single-identity and single-key to multi-identity and multi-key, it can support ciphertext homomorphic computation under multi-identity. However, compared with GSW13, our scheme adds some parameters, such as a maximum degree of identity $\mathcal{D}$, the private key sk_{id} corresponding to the identity id, etc., so the efficiency of our scheme is slightly low compared with the GSW13 scheme.

Compared with the Clear scheme [17]. The scheme we propose can perform multiple ciphertext homomorphic computations on different ciphertexts corresponding to different identities. Let d be the maximum number of different identities of the participating edge devices, the expanded ciphertext is β-noisy ciphertext, and the noise of the Clear scheme after the homomorphic computation is $\beta(d\,(m+1)\,\ell_q+1)$, the noisy expansion is $(d\,(m+1)\,\ell_q+1)$. The noise after homomorphic computation of our scheme is $\beta(dm+1)$, and the noisy expansion is $(dm+1)$. Therefore, the noisy expansion rate of our scheme is smaller than that of the Clear scheme.

4.3 Numerical Results

To evaluate the performance of our proposed scheme, we conducted experiments. Our scheme is mainly written in GO language, and Truffle is used to compile and deploy smart contracts. The experimental operating system is Ubuntu16.04, the blockchain platform is Hyperledger Fabric1.3. The hardware environment includes an NVIDIA GeForce GTX 1050 graphics card, a CPU with 2.3 GHz, a Quad-core i5-8300H Intel Core, a memory with 8G dual-channel and an SSD with 128G.

Figure 5 shows the time cost of the four stages $\xi.\ KeyGen$, $\xi.\ Encrypt, \xi.\ AddEval$, $\xi.\ MultEval$ in MIBFHE. Where, we set the value of the security parameter λ 100, and set the ciphertext dimension n the power of 2, respectively, 2048, 4096, ..., 14366. From the figure, we can observe that the time cost of these four stages increases linearly with the growth of n. However, if the value of the ciphertext dimension n is small, the execution time of $\xi.\ KeyGen$, $\xi.$ Encrypt, $\xi.\ AddEval$, $\xi.\ MultEval$ will be very small, and the total time will be very small. Therefore, our scheme is feasible in practice.

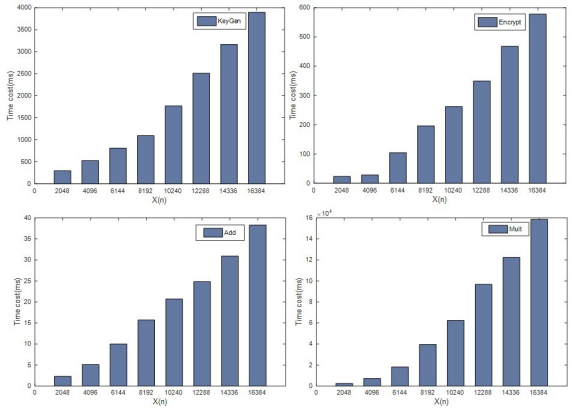


Figure 5: Time cost of MIBFHE

Table 1: Performance evaluation

Case	Multi-key	Multi-identity	Leveled	Homomorphic type
GSW13	No	No	Yes	Many-times
Clear scheme	Yes	Yes	No	Single-time
Our scheme	Yes	Yes	Yes	Many-times

Then, we conducted a comparative experiment to compare our proposed protocol (Protocol II in the figure) with the protocol in [11] (Protocol I in the figure). The two experiments' environment and parameters are the same, and only the data set size or the threshold size is changed to observe the time cost of the protocol execution.

Figure 6 shows the average time cost (i.e., computation latency) between Protocol II Protocol I by varying the data sizes. We can observe from the figure that the two protocols' computation latency increases linearly with the increase of the data sizes. The time cost of Protocol II is lower than that of Protocol I. The reason is that our scheme is based on a distributed architecture. The homomorphic computing operations between servers are parallel, and the computing time cost is only determined by a single server.

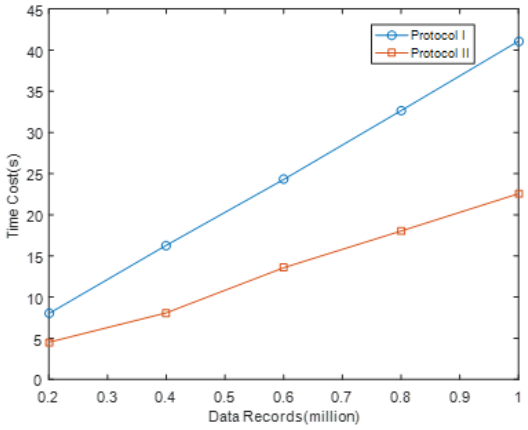


Figure 6: Computation latency with various data records

In Figure 7, we compare the computation latency by varying the Protocol I and Protocol II threshold values. The figure shows that increasing the threshold can effectively increase the computation latency. The computation latency of Protocol II is lower than that of Protocol I. The reason is that our proposed protocol can utilize the computing resources of all distributed edge servers to execute in parallel operation. Therefore, the edge server's computation load in our protocol is low, which results in a smaller computation latency of our protocol. This further illustrates the superiority of the MPC protocol based on the distributed architecture that we constructed.

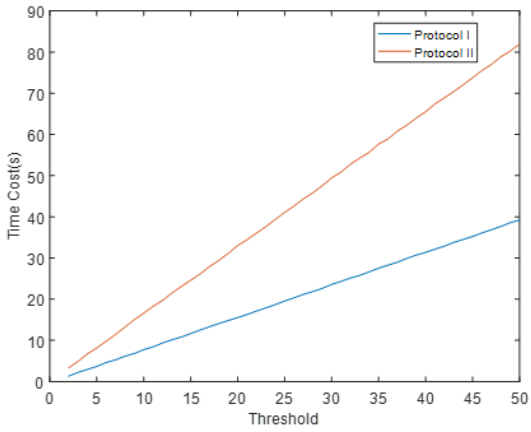


Figure 7: Computation latency with various threshold

5 Conclusions

In this paper, we propose a blockchain-based edge data sharing and secure computation solution, which can solve the data security and user privacy issues in data sharing. Our solution enables edge devices to obtain autonomous control of data while ensuring the fairness and security of computing. Blockchain technology can ensure the verifiability of calculations, and secure multi-party computing technology can ensure the confidentiality of input data during the calculation process. Based on LWE, a multi-identity hierarchical fully homomorphic encryption scheme is constructed to solve the existing edge computing data privacy protection problem. Besides, the blockchain is introduced into secure multi-party computing, and economic penalty mechanisms are added to verify the identity of nodes to build fair MPC. We further conduct security analysis and performance analysis. Our protocol can perform homomorphic computing operations in parallel on distributed edge servers. The result analysis shows that our solution achieves a smaller communication overhead and computing delay. The fully homomorphic encryption algorithm has the problem of low efficiency. Future work should consider further optimizing the efficiency of the homomorphic encryption algorithm.

Acknowledgments

This work was supported in part by the State Grid Jiangxi Information & Telecommunication Company

Project "Research on de-boundary security protection technology based on zero trust framework" under Grant 52183520007V. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] P. Bert, G. Eberhart, L. Prabel, A. Roux-Langlois, and M. Sabt, "Implementation of lattice trapdoors on modules and applications," in *International Conference on Post-Quantum Cryptography*. Springer, 2021, pp. 195–214.

[2] D. Bogdanov, S. Laur, and J. Willemson, "Sharemind: A framework for fast privacy-preserving computations," in *European Symposium on Research in Computer Security*. Springer, 2008, pp. 192–206.

[3] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[4] S. Carpov, K. Deforth, N. Gama, M. Georgieva, D. Jetchev, J. Katz, I. Leontiadis, M. Mohammadi, A. Sae-Tang, and M. Vuille, "Manticore: Efficient framework for scalable secure multiparty computation protocols." *IACR Cryptol. ePrint Arch.*, vol. 2021, p. 200, 2021.

[5] M. Clear and C. McGoldrick, "Multi-identity and multi-key leveled fhe from learning with errors," in *Annual Cryptology Conference*. Springer, 2015, pp. 630–656.

[6] L. Feng, Y. Jie, L. Zhibin, and Q. Jiayin, "A secure multi-party computation protocol for universal data privacy protection based on blockchain," *Journal of Computer Research and Development*, vol. 58, no. 2, p. 281, 2021.

[7] T. Feng, H. Pei, P. Xie, and X. Feng, "Blockchain data sharing scheme based on searchable agent re-encryption," *International Journal of Network Security*, vol. 23, no. 3, pp. 535–544, 2021.

[8] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Annual Cryptology Conference*. Springer, 2013, pp. 75–92.

[9] O. Goldreich, *Foundations of Cryptography, Volume 2*. Cambridge university press Cambridge, 2004.

[10] S. Goldwasser, "Multi party computations: past and present," in *Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing*, 1997, pp. 1–6.

[11] V. Goyal, E. Masserova, B. Parno, and Y. Song, "Blockchains enable non-interactive mpc," in *Theory of Cryptography Conference*. Springer, 2021, pp. 162–193.

[12] L. Guo, X. Li, and J. Gao, "Multi-party fair exchange protocol with smart contract on bitcoin." *Int. J. Netw. Secur.*, vol. 21, no. 1, pp. 71–82, 2019.

[13] X.-T. Lin, R.-P. Chen, H.-N. Wu, F.-Y. Meng, Q.-W. Liu, and D. Su, "A composite function model for predicting the ground reaction curve on a trapdoor," *Computers and Geotechnics*, vol. 141, p. 104496, 2022.

[14] J. Liu, X. He, R. Sun, X. Du, and M. Guizani, "Privacy-preserving data sharing scheme with fl via mpc in financial permissioned blockchain," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[15] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 2012, pp. 1219–1234.

[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[17] T. Pal and R. Dutta, "Chosen-ciphertext secure multi-identity and multi-attribute pure fhe," in *International Conference on Cryptology and Network Security*. Springer, 2020, pp. 387–408.

[18] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116 974–117 017, 2020.

[19] H. Pu, Z. Cui, and T. Liu, "An electronic voting scheme using secure multi-party computation based on secret sharing," *International Journal of Network Security*, vol. 23, no. 6, pp. 997–1004, 2021.

[20] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5g mobile edge computing: Architectures, applications, and technical aspects," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.

[21] F.-Y. Yang, "Improvement on a trapdoor hash function." *Int. J. Netw. Secur.*, vol. 9, no. 1, pp. 17–21, 2009.

[22] A. C. Yao, "Protocols for secure computations," in *23rd annual symposium on foundations of computer science (sfcs 1982)*. IEEE, 1982, pp. 160–164.

[23] A. C.-C. Yao, "How to generate and exchange secrets," in *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*. IEEE, 1986, pp. 162–167.

[24] Y. Yao and F. Yu, "Privacy-preserving similarity sorting in multi-party model." *Int. J. Netw. Secur.*, vol. 19, no. 5, pp. 851–857, 2017.

Biography

Jiawen Bao received B.S. degree in computer science and technology from Nanchang Hangkong University,

Nanchang, in 2020. Since 2020, He is a M.S. candidate in information security at North China Electric Power University. His research interests include machine learning and artificial intelligence, information security, and blockchain.

Zhang Yan is a M.S. candidate fellow in School of control and Computer Engineering of North China Electric Power University. She received B.S. degree in computer science and technology from Shanghai University of Electric Power. Her current research is in machine learning and artificial intelligence, blockchain, electric vehicle in smart grid.

Liu Haiqing earned an engineering PHD degree in Computer Application Technology from Wuhan University. She went to WPI (Worcester Polytechnics Institute) in the USA as a visiting scholar for 1 year. She was employed as an associate professor and master's supervisor by Wuhan University. She then transferred to the Control and Computer Engineering Academy in North China Electric Power University. Her current research is Software Engineering, Database and Artificial Intelligence.

Yuancheng Li received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2003.From 2004 to 2005, he was a Postdoctoral Research Fellow with the Digital Media Lab, Beihang University, Beijing, China. Since 2005, he has been with the North China Electric Power University, Beijing, where he is a Professor and the Dean with the Institute of Smart Grid and Information Security. From 2009 to 2010, he was a Postdoctoral Research Fellow with the Cyber Security Laboratory, College of Information Science and Technology, Pennsylvania State University, State College, PA, USA.

Rixuan Qiu received B.S. degree in Information and Computing Science from North China Electric Power University, Beijing, in 2015. Since 2015, he is a M.S. candidate in information security at North China Electric Power University. His research interests include network and information security, industrial control network security.

Formal Security Evaluation and Research of Automotive CAN Protocol Based on CPN

Tao Feng, Lu Zheng, and Peng-Shou Xie
(Corresponding author: Lu Zheng)
Faculty of Computer and Communication, Lanzhou University of Technology, China
Email: zhengl0518@163.com
(Received Aug. 31, 2021; Revised and Accepted Jan. 13, 2022; First Online Feb. 26, 2022)

Abstract

Controller Area Network (CAN) bus is the most representative in-vehicle bus technology in Intra-Vehicular Networks (IVNs) for its high reliability. However, the continuous increment of complex electronic systems makes the IVNs vulnerable to various malicious attacks. This work proposes a protocol model detection method based on a combination of Colored Petri Nets (CPN) and the Dolev-Yao attack model to a CAN2.0B-based IVN protocol for formal security evaluation. The results show that this protocol is vulnerable to two types of man-in-the-middle attacks: replay and spoofing. To address this, we use the asymmetric cryptosystem, and digital signature combined with the HASH function for reinforcement and again use the protocol model detection method to evaluate the security of the new protocol. The results show that the new protocol can effectively improve security.

Keywords: CAN Protocol; CPN Tools; Dolev-Yao; Formal Analysis; Security Evaluation

1 Introduction

With the fusion of vehicles and information technology, more systems and functions of ordinary vehicles are transformed from mechanical systems to electronic systems. Vehicle sensors, engine control and Anti-lock Brake System (ABS) and Advanced Driver-Assistance System (ADAS) [3], etc. have been put under the control of Electronic Control Units (ECUs) [7]. Manufacturers have transitioned from wire-heavy, point-to-point schemes to the bus, such as LIN, CAN, FlexRay, and MOST [22].

CAN is the international standard for communication inside the vehicle due to its high reliability, high fault tolerance, real-time, flexibility, etc. [9]. However, the security attributes of the CAN bus are mainly designed to ensure reliable communication. There are no security attributes such as encryption, authentication, integrity, and confidentiality. In recent years, the continuous improvement of functional requirements has caused a large increase in ECUs. [19] At the same time, the continuous increase of open interfaces makes IVN faces huge risks. Attackers can launch direct or indirect, long-distance or short-distance attacks on cars through OBD ports, WiFi, Bluetooth, cellular networks, etc. [18], and steal private data and cause serious harm. However, simply adding firewall protection to the vehicle gateway cannot fundamentally prevent malicious attacks. Therefore, the core focus of in-vehicle security is to protect the information security of the CAN protocol in the vehicle.

The rest of the paper is organized as follows. Section 2 discusses the security issues described in related work. Section 3 presents the CAN frame and security requirements. Section 4 models the CAN2.0B-based IVN protocol [17] and presents consistency verification based on CPN. Section 5 introduces the attacker model and describes the formal security analysis for the CAN2.0B-based IVN Protocol. Section 6 designs and models the new protocol. Section 7 presents the formal security analysis based on CPN. Section 8 compares our model detection method and the security attributes of our protocol with other work. Section 9 concludes the work.

2 Related Work

To build a safe in-vehicle CAN communication environment, in the past ten years, the European Union and other organizations have funded and conducted many projects to deal with in-vehicle network security, such as E-safety Vehicle Intrusion Protected Applications (EVITA) [8], Secure Vehicle Communication (SEVECOM), Open Vehicular Secure Platform (OVERSEE), etc. EVITA designs and verifies the architecture of the vehicle network, and develops a hardware security module (HSM) [8]. SEVECOM studies the protection of vehicle sensor data based on vehicle security middleware from the perspective of threat analysis and security architecture. OVERSEE provides the design of a vehicle application and communication platform, and builds a standardized in-vehicle environment [5]. However, this type of project did not study the specific security mechanism for the vehicle network, and did not propose the corresponding security protocol.

Security protocols in [10, 15, 23] were designed for the limited data load of the CAN data frames. However, these protocols do not support real-time data processing, and there is no corresponding formal analysis and security evaluation for these protocols. In [14], Radu and Garcia proposed a protocol to generate session keys, but the protocol lacked entity authentication and the specific formal analysis of the protocol. Woo *et al.* [17] proposed a set of CAN2.0B-based generic IVN protocols to solve the problem of providing a safe and fast key distribution mechanism for IVN. Basker *et al.* [13] used Tamarin Prover for formal security verification for this protocol, but it lacks intuitiveness.

To sum up, most of the existing research work on vehicle network security focuses on adding security mechanisms to achieve corresponding security functions, and the security evaluation and consistency verification of the protocol is still in its infancy. This paper focuses on the existing CAN2.0B-based IVN protocol [17], based on the theory of combining the colored Petri net (CPN) and the Dolev-Yao attack model in [1], conducts the security evaluation of the protocol and discovers its potential loopholes, fundamentally improves the protocol and prevents malicious attacks.

Compared with the existing research results, the main contributions of this paper include three aspects:

1) We propose a protocol model detection method that combines a colored Petri net (CPN) and the Dolev-Yao attack model;

2) We use the CPN Tools to perform CPN-based formal modeling of the CAN2.0B-based IVN protocol based on its specification, and introduce the Dolev-Yao attacker model to conduct security evaluation and perform consistency verification of the protocol model;

3) For the various vulnerabilities exposed after security evaluation of the protocol, we propose a new set of lightweight security enhancement protocols, and perform formal modeling of the new protocol. We verify functional and security by introducing the Dolev-Yao attacker model;

3 CAN Frames and CAN Security Requirements

3.1 CAN Frames

CAN bus was developed by German BOSCH company [4] in 1986, it has become the world's mainstream bus generally recognized by automobile manufacturers in various countries and has been set as an international standard. Figure 1 shows the CAN2.0B data frame format.

The CAN bus protocol supports two message formats, which can be divided into 11-bit CAN2.0A standard frame format and 29-bit CAN2.0B extended frame format according to the difference of ID fields [23]. The CAN2.0B

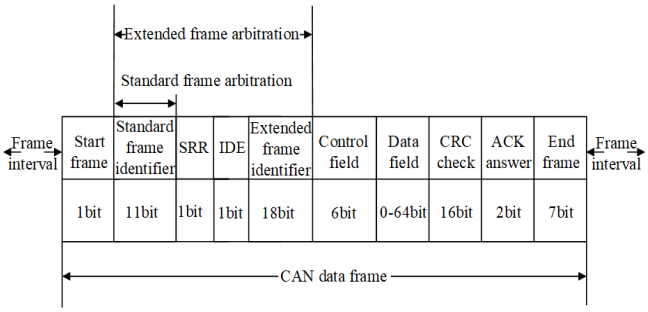


Figure 1: CAN2.0B data frame

protocol is compatible with the CAN2.0A, and is compatible with data messages in standard frame format and extended frame format at the same time. Since the data field defined by the CAN protocol can contain 0-8 bytes [11], the time consumed during data transmission is relatively short, thereby the real-time nature of data transmission is guaranteed. CAN protocol uses CRC to provide error checking in data transmission. However, only using CRC can only ensure the accuracy of the transmitted data, but cannot ensure reliable transmission [12].

3.2 CAN Security Requirements

A large number of researches on in-vehicle network security point out that the lack of data encryption and node authentication is the most serious vulnerability of CAN. However, due to the limitations of real-time and the limited data payload of CAN data frames and the limited memory in the ECU [11], achieving data authentication on CAN is a huge challenge [7]. In addition, malicious nodes can easily launch tampering attacks and replay attacks by stealing the ID of the bus node. The CRC on the CAN bus cannot ensure safe and accurate data transmission. Therefore, security protocols need to encrypt and authenticate data frames to prevent tampering attacks and replay attacks.

For the defects that CAN cannot encrypt data, cannot provide identity authentication for nodes, and cannot ensure the correctness and integrity of data, combined with the design characteristics of CAN bus, the security functions added on the original CAN protocol need to meet the following points of security demand:

1) Confidentiality. Each data frame in the CAN bus should be encrypted;

2) The authenticity of identity. The two parties who need to communicate in the CAN bus should confirm the identity of the other party before communication to prevent attackers from pretending to be a normal node to steal private data on the bus;

3) Correctness. The communication receiver in the CAN bus should ensure that the message comes from the correct sender;

4) Completeness. The communication receiver in the CAN bus should ensure that the message has not been tampered with by the attackers during the transmission;

5) Real-time. Due to the limited data payload of the CAN data frame, the freshness of the data transmitted on the CAN bus needs to be ensured to prevent replay attacks.

3.3 CPN Tools

The formal analysis method uses mathematics or logical structure to describe the system model, and verifies whether the system meets the requirements of consistency and completeness through a certain form of reasoning. Early methods used for formal analysis of protocols, such as BAN logic, string space, state machines, etc. The formal verification of these methods focuses on the theorem proving, without specific analysis of formal semantics. In recent years, powerful analysis tools [20] such as ProVerif, Scyther, Tamarin Prover, and CPN Tools [16] have emerged, which can perform formal security verification and semantic analysis for protocols.

This paper uses CPN Tools to edit, simulate, and analyze colored Petri nets, perform state space analysis and performance analysis, and model, analyze, and verify protocols. The programming language of CPN is based on the standard Markup Language (ML) [2] language, CPN Tools is similar to a state machine, and can simulate and analyze concurrent systems. Because of its powerful state space analysis [21] capabilities and easy-to-understand expressions, it has become one of the mainstream formal modeling and analysis tools for security protocols, and is widely used in many fields.

4 CAN2.0B-based IVN Protocol

The existing CAN2.0B-based IVN protocol [17] uses the security mechanism of HMAC combined with the KDF key derivation function. The communication process of the protocol is divided into three protocols, including the initial session key distribution phase (ISDP), the session key update phase (SKUP), and the data frame phase (DFP).

Inside the protocol, K_i and GK are symmetric keys shared by ECUi and GECU. EK_k is the encryption key for the kth session. AK_k is the authentication key for the kth session. KEK_k is the encryption key for the kth session in the key update phase. KGK_k is the generated key of the kth session in the key update phase. CTR_{GECU} is the data frame counter of GECU. M is the plaintext of the data frame. C is the ciphertext of the data frame. $KDF_x()$ is the key derivation function. $H_x()$ is the HASH function.

4.1 Modeling CAN2.0B-based IVN Protocol Based on CPN

When building large-scale models of protocol, using a traditional CPN single-page model is not only very complicated, but also not intuitive. To solve this problem, this paper adopts the idea of building hierarchical models, and replaces a module with substitution transitions of the CPN Tools in the high-level model, and builds multilayer system models with substitution transitions.

In this paper, the double-layer rectangles represent substitution transitions. The ovals represent the message places, which are used to store messages in the communication process. The arrows represent message transmission, which is used to transmit messages in the communication process.

We modeled the protocol into three levels: top, middle and bottom. The top-level CPN model of the protocol is shown in Figure 2. It intuitively simulates the process of the entire protocol.

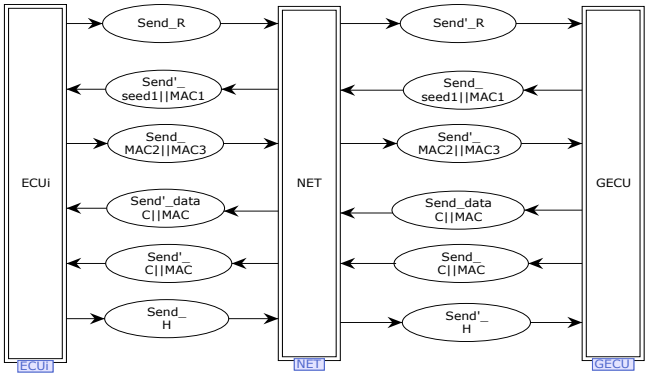


Figure 2: Top-level CPN model

The internal CPN model of the substitution transition NET is shown in Figure 3. The arrows represent the direction of data transmission.

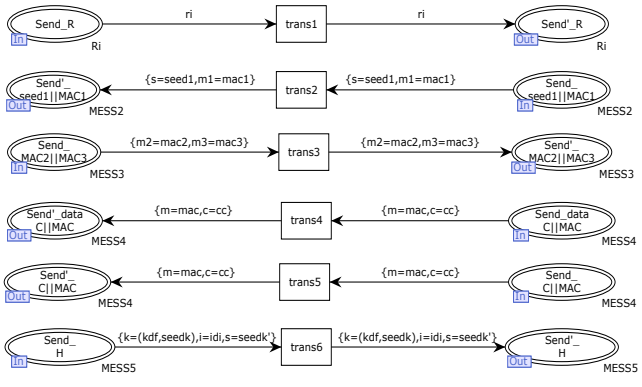


Figure 3: CPN model of substitution transition NET

The middle-level CPN model of the protocol is shown in Figure 4. It consists of 1 transition, 8 substitution tran-

sitions, and 14 places. Transition start means that ECUi selects the random number R_i and sends it to GECU. Substitution transitions Initialize and Initialize' represent the process of generating the initial session key. Substitution transitions date and date' represent the process of data frame transmission. Substitution transitions update and update' represent the process of updating the session key. Substitution transition cycle is responsible for storing and integrating the updated key seeds in the SKUP phase, and real-time updating of the keys used in the DFP phase.

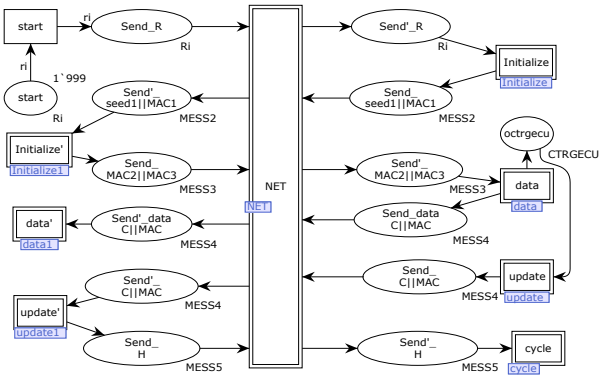


Figure 4: Mid-level CPN model

The bottom-level CPN model of the protocol consists of 8 parts. The ISDP phase, DFP phase, and SKUP phase are described in turn according to the protocol process. The internal CPN model of the substitution transition Initialize is shown in Figure 5. It simulates the process of GECU selecting the random number $Seed_1$, using K_i and HMAC algorithm to generate MAC_1 from ID_{GECU}, ID_i, R_i and $Seed_1$, and sending it to ECUi with $Seed_1$.

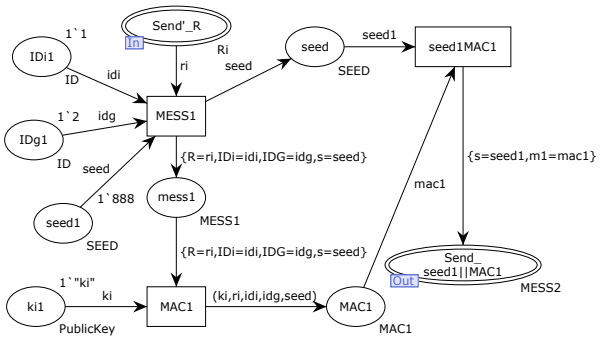


Figure 5: CPN model of substitution transition Initialize

The internal CPN model of the substitution transition Initialize' is shown in Figure 6. It simulates ECUi verifying MAC_1 and using $Seed_1$, GK and KDF to generate the initial session keys EK_1, AK_1, KEK_1 and KGK_1, and using K_i and HMAC algorithm to generate MAC_2 from ID_i and $Seed_1$, then using AK_1 and HMAC algorithm to generate MAC_3 from ID_i, EK_1, AK_1, KEK_1 and KGK_1, then sending MAC_2 and MAC_3 to GECU.

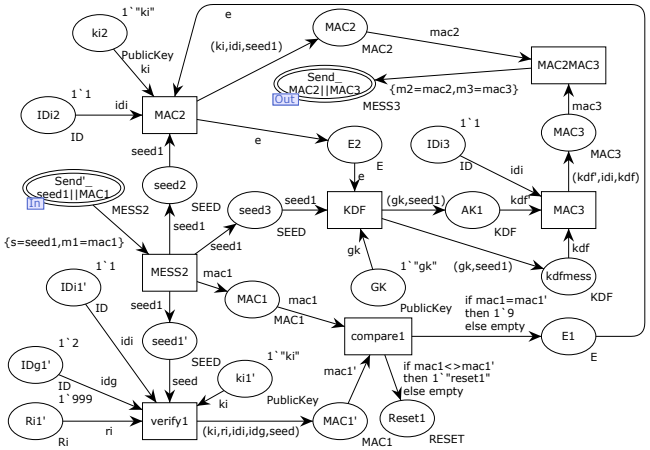


Figure 6: CPN model of substitution transition Initialize'

The internal CPN model of the substitution transition data is shown in Figure 7. It simulates GECU verifying MAC_2 and using KDF to calculate the initial session keys, and verifying MAC_3, and using EK_k and AES-128 to generate C from CTR_{GECU} and M, and using AK_k to generate MAC from ID_{GECU}, C and CTR_{GECU}, and sending it to ECUi with C, then increasing CTR_{GECU}.

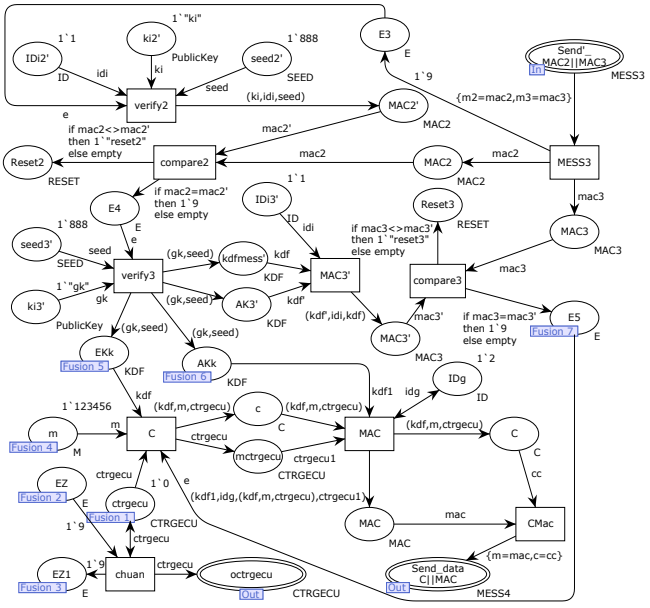


Figure 7: CPN model of substitution transition data

The internal CPN model of the substitution transition data' is shown in Figure 8. It simulates ECUi verifying MAC, decrypting to obtain M, increasing CTR_{GECU}.

The internal CPN model of the substitution transition update is shown in Figure 9. It simulates GECU selecting random number $Seed_{k+1}$, using KEK_k to generate C from CTR_{GECU} and $Seed_{k+1}$, using AK_k to generate MAC from ID_{GECU}, C, CTR_{GECU}, composing C and MAC into a key request message, sending it to ECUi.

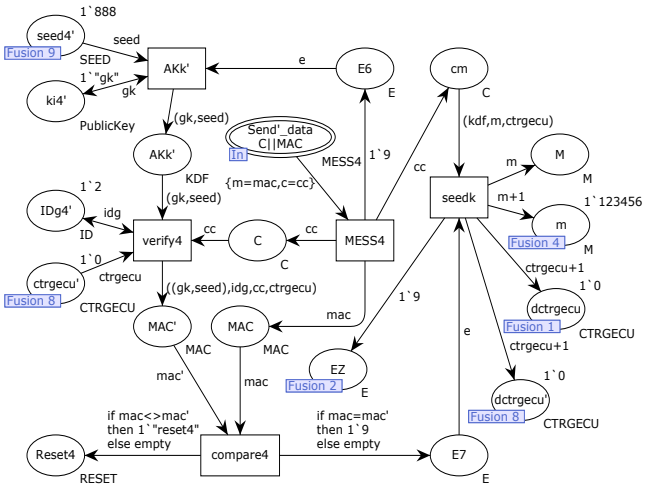


Figure 8: CPN model of substitution transition data'

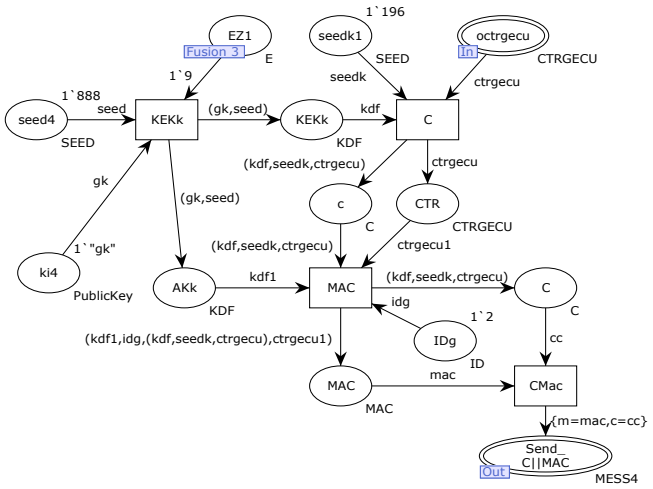


Figure 9: CPN model of substitution transition update

The internal CPN model of the substitution transition update' is shown in Figure 10. It simulates ECUi verifying MAC and decrypting C, and using KDF with KGK_k to derive the session key that will be used in the k+1 session, and initializing data frame counter to zero, then using AK_{k+1} and HMAC algorithm to generate a key response message from ID_i and $Seed_{k+1}$, and sending it to GECU.

4.2 Consistency Verification for the CPN Model

The consistency between the original CPN model and the protocol specification determines the accuracy of the security evaluation of the subsequent protocol model. We uses the state space analysis tool in CPN Tools to verify the consistency of the CPN model and the function of the protocol. The state space analysis report generated by the original CPN model is shown in Table 1.

In the state space report shown in Table 1, the number

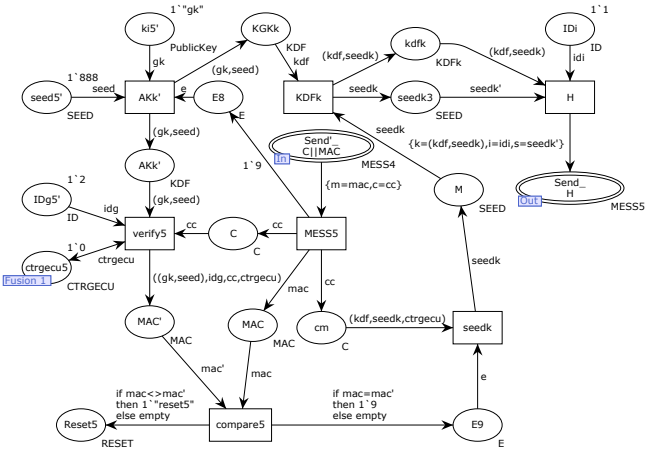


Figure 10: CPN model of substitution transition update'

Table 1: State Space report of the original model of CAN2.0B-based IVN protocol

Type	Number
State Space Node	56
State Space Arc	55
SCC Graph Node	56
SCC Graph Arc	55
MainState Space Node	0
Live Transition Instances	0
Dead Transition Instances	0
Dead Marking	1

of nodes in the state space is the same as the number of strongly connected nodes, and the number of directed arcs in the state space is the same as the number of strongly connected arcs, it indicates that all state nodes of the original CPN model are reachable, and there is no state infinite loop behavior in the model. In the case of no attackers, the model will successfully perform initial key distribution, data frame transmission, trigger the key update and reply according to the protocol process. All interactive processes will not trigger the 5 reset places established in the model. The number of dead transitions is 0, which indicates that there is no situation where the transitions cannot be triggered. The number of dead markings is 1, which indicates that this model interacts following the protocol process, and there is 1 termination state, which is consistent with the expected result.

5 Security Evaluation of the Attacker Model

Dolev and Yao proposed an attacker model to verify the cryptographic protocols. So far, most of the attacker models introduced for the research of security protocols

are based on the Dolev-Yao model. The Dolev-Yao attacker model points out that on the assumption that the cryptographic system is "absolutely secure", it does not study the security of the specific cryptographic algorithms of the protocol, but takes the inherent security properties of the protocol as the research goal.

The NET subpage in the original model of the protocol describes the data transmission path between ECUi and GECU in detail, it is equivalent to the network channel that the protocol information must pass through. Therefore, this paper introduces the Dolev-Yao attacker model through the NET subpage in the original model.

5.1 Introducing the Attacker Model

According to the powerful capabilities of the attacker in the Dolev-Yao attacker model, this paper launches three types of man-in-the-middle attacks, including replay, tampering, and spoofing on the network channel. The attacker model of the CPN model of the original protocol is shown in Figure 11.

The blue type of places and transitions represent the replay attack launched by Transition Attack1 and Attack2. The replay attack on the trans5 path is adopted an attacker model based on message splitting and combination, which can ensure the attacker's ability and effectively reduce the state space. The type of the place dis is DB, which stores the split and the unsplit message. Places P31, P32, P33, P34 store all kinds of atomic information. Transition TF uses the attacker's transition rules to store undecipherable messages in place P5. Transition TE uses the attacker's synthesis rules to synthesize and store atomic messages in place P5. The type of place P5 is CB, which stores information that cannot be decrypted and the information after the atomic message is synthesized. The red type of places and transitions represent the tampering attack launched by Transition Attack3. The place ASEED1 tampered with the key seed in the original message. The purple type of places and transitions represent the spoofing attack launched by all transitions trans1, trans2, trans3, trans4, trans5, and trans6 on the network transmission path.

5.2 Security Analysis of the Model

The state space reports of the original model and the attacker model are shown in Table 2. All state nodes of the attacker model are reachable, and there is no state infinite loop behavior in the model. The number of state space nodes and directed arcs of the attacker model has not increased significantly compared with the original model, it indicates that the introduction of the attacker model did not cause the state space to be too large to explode, it further shows that the introduced attacker model is effective.

The number of dead transitions of the attacker model is 0, which indicates that there is no unpredictable final state of the protocol process due to the man-in-the-middle

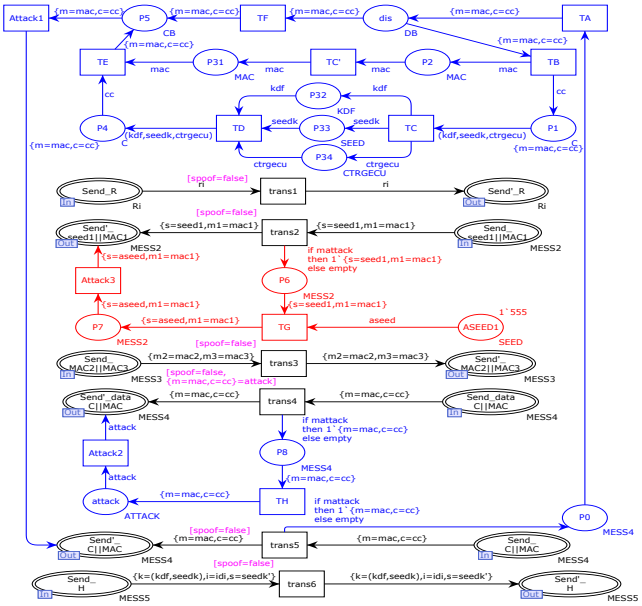


Figure 11: Formal description based on the attacker model

attack. The number of dead markings is 10, it indicates that after the introduction of the attacker model, there are 10 termination states where the data transmission is completed.

In the attacker model, the serial numbers of 10 dead markings can be obtained by writing the ML program and ListDeadMarking() function, all transitions and places corresponding to the serial number of each dead marking can be queried by NodeDescriptor() function. Through further analysis of the state of the places and transitions, we find that the 6 dead markings are caused by the replay attacks, including 3 dead markings are due to the lack of backward confidentiality caused by SKUP using the previous session key to update key, the remaining 3 dead markings are due to the lack of identity authentication caused by DFP only uses message verification to transmit data frames. The 4 dead markings are due to the attacker pretending to be a legal identity for message authentication caused by the spoofing attacks.

6 The New Protocol

6.1 The Message Flow Model of the New Protocol

In our new protocol, we introduce the asymmetric cryptosystem and digital signature for authentication to fix the vulnerability of the original protocol. we ensure the backward confidentiality of the session key. We use the HASH algorithm instead of HMAC to reduce the costs of computing and storage. The message flow model of the new protocol is shown in Figure 12.

Inside the model, PU_a and PR_a are the public and

Table 2: State Space reports of the original model and the attack model of CAN2.0B-Based IVN protocol

Type	Original Model	Original Attack Model
State Space Node	56	1288
State Space Arc	55	3343
SCC Graph Node	56	1288
SCC Graph Arc	55	3343
MainState Space Node	0	0
Live Transition Instances	0	0
Dead Transition Instances	0	0
Dead Marking	1	10

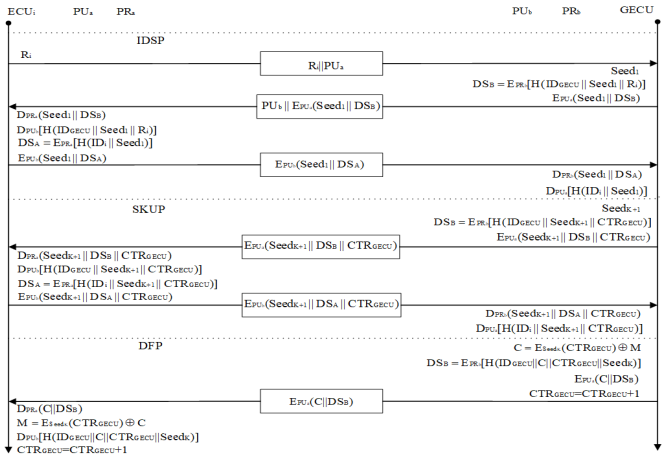


Figure 12: Message flow model of the new protocol

private keys generated by ECUi respectively, PU_b and PR_b are the public and private keys generated by GECU respectively, the pair of public and private keys are distributed with the aid of the KPI architecture. DSB is the digital signature generated by GECU. DSA is the digital signature generated by ECUi. M is CAN data frame. CTR_{GECU} is the data frame counter of GECU. $H_x()$ is the HASH function. The process of ISDP is shown as follows:

1) ECUi selects a random number R_i and sends it to GECU together with PU_a;

2) GECU selects a random number $Seed_1$, generates the $hash1$ from ID_GECU and $Seed_1$, uses PR_b to generate the DSB from $hash1$, and uses PU_a to encrypt Seed1 and DSB, and transmits them to ECUi together with the PU_b;

3) ECUi uses PR_a to decrypt to obtain $Seed_1$ and DSB, uses PU_b to decrypt to obtain the $hash1$. ECUi verifies $hash1$;

4) ECUi generates the $hash2$ from ID_i and $Seed_1$, uses PR_a to generate the DSA from $hash2$, uses PU_b to encrypt $Seed_1$ and DSA, and transmits them to GECU;

5) GECU uses PR_b to decrypt to obtain $Seed_1$ and DSA, uses PU_a to decrypt to obtain $hash2$, GECU verifies $hash2$;

The process of SKUP is shown as follows:

1) GECU selects a random number $Seed_{k+1}$, generates $hash3$ from ID_{GECU}, $Seed_{k+1}$ and CTR_{GECU}, uses PR_b to generate the DSB from $hash3$, and uses PU_a to encrypt $Seed_{k+1}$, CTR_{GECU} and DSB, and sends them to ECUi;

2) ECUi uses PR_a to decrypt to obtain $Seed_{k+1}$, DSB, CTR_{GECU}, uses PU_b to decrypt to obtain $hash3$. ECUi verifies $hash3$;

3) ECUi generates $hash4$ from ID_i, $Seed_{k+1}$ and CTR_{GECU}, uses PR_a to generate DSA from $hash4$, uses PU_b to encrypt $Seed_{k+1}$, CTR_{GECU} and DSA, and sends them to GECU;

4) GECU uses PR_b to decrypt to obtain $Seed_{k+1}$, CTR_{GECU} and DSA, uses PU_a to decrypt to obtain $hash4$. GECU verifies $hash4$;

The process of DFP is shown as follows:

1) GECU uses Seedk as the key to transmits M, uses the AES-128 to generate C from CTRGECU and M based on

$$C = E_{Seed_k}(CTR_{GECU}) \oplus M$$

GECU generates the $hash5$ from ID_GECU, C, $Seed_k$ and CTR_{GECU}, uses PR_b to generate the DSB from $hash5$, uses PU_a to encrypt C and DSB, and sends them to ECUi. GECU increases the CAN data frame counter (CTR_{GECU});

2) ECUi uses PR_a to decrypt to obtain C and DSB, decrypts to obtain the plaintext M based on

$$M = E_{Seed_k}(CTR_{GECU}) \oplus C$$

ECUi uses PU_b to decrypt to obtain the $hash5$, ECUi verifies $hash5$. ECUi increases CTR_{GECU};

6.2 Modeling the New Protocol Based on CPN

We model the new protocol based on CPN to verify whether its safety mechanism meets the safety requirements. The middle-level CPN model of the new protocol is shown in Figure 13. Transition start represents the process of ECUi and GECU generating public and private keys respectively. Substitution transitions Initialize and Initialize' represent the process of completing the initial session key distribution and identity authentication. Substitution transition date and date' represent the process of completing the secure transmission of the CAN data frames. Substitution transition update and update' represent the process of completing the session key update.

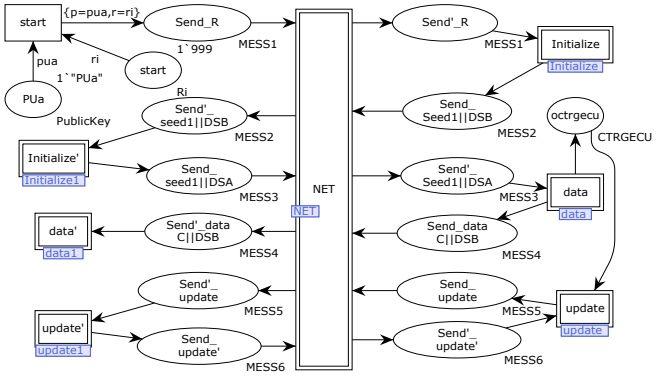


Figure 13: Middle-level CPN model of the new protocol

The internal CPN model of the substitution transition Initialize is shown in Figure 14. Transition HASH1 combines ID_{GECU}, R_i, and $Seed_1$ into $hash1$. Transition DSB1 combines $hash1$ and PR_b into DSB_1. Transition seed1DSB1 combines PU_a, $Seed_1$, DSB_1, and PU_b, and sends them to ECUi through the place Send_Seed1||DSB.

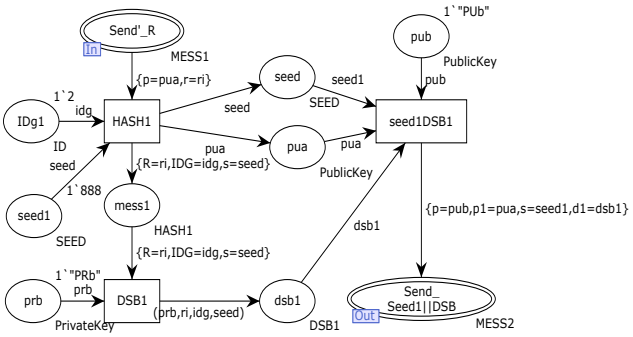


Figure 14: CPN model of new protocol's substitution transition Initialize

The internal CPN model of the substitution transition Initialize' is shown in Figure 15. Transition compare1 verifies hash1 in the DSB_1. If the verification succeeds, transition HASH2 combines $Seed_1$ and ID_i into $hash2$, else place Reset1 is triggered. Transition DSA1 combines $hash2$ and PR_a into DSA_1. Transition seed1DSA combines PU_b, $Seed_1$, and DSA_1, and sends them to GECU through the place Send_seed1||DSA.

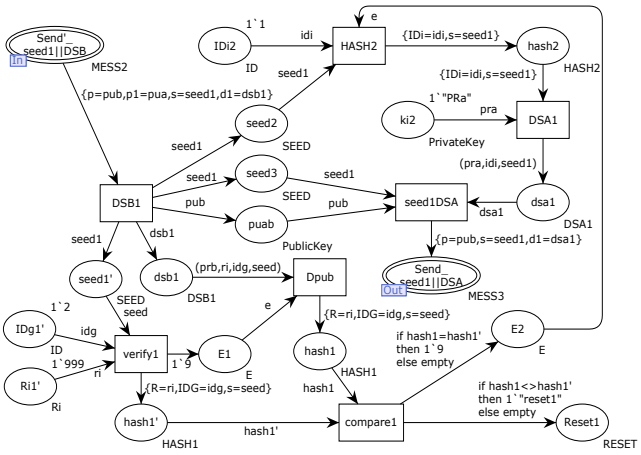


Figure 15: CPN model of new protocol's substitution transition Initialize'

The internal CPN model of the substitution transition data is shown in Figure 16. Transition compare2 verifies $hash2$ in the DSA_1. If the verification succeeds, transition C combines m, $ctrgecu$ and $Seed_k$ into C, else place Reset2 is triggered. Transition HASH3 combines ID_{GECU}, C, $ctrgecu$, and $Seed_k$ into $hash3$. Transition DSB2 combines $hash3$ and PR_b into DSB_2. Transition CDSB combines PU_a, C, and DSB_2, and sends them to ECUi through the place Send_dataC||DSB. Transition chuan is responsible for updating the data frame counter in time when a data frame is transmitted.

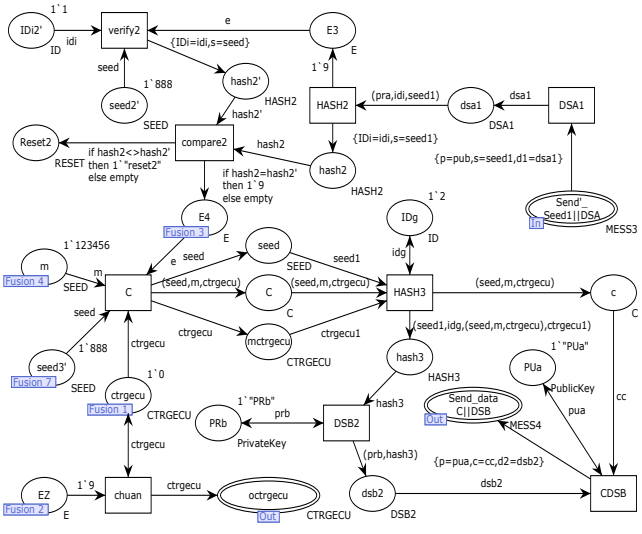


Figure 16: CPN model of new protocol's substitution transition data

The internal CPN model of the substitution transition

data' is shown in Figure 17. Transition compare3 verifies $hash3$ in the DSB_2. If the verification succeeds, the transition M decrypts C so that ECUi obtains M, else place Reset3 is triggered. At this time, a data frame has been transmitted, the data frame counter stored in the place is increased by 1.

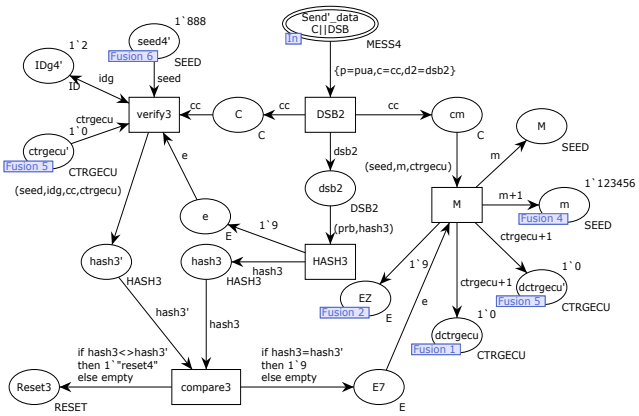


Figure 17: CPN model of new protocol's substitution transition data'

The internal CPN model of the substitution transition update is shown in Figure 18. Transition HASH4 combines ID_{GECU}, $ctrgecu$, and $Seed_k$ into $hash4$. Transition DSB3 combines $hash4$ and PR_b into DSB_3. Transition update combines PU_a, $Seed_k$, and DSB_3, and sends them to ECUi through the place Send_update. Transition DSA2 verifies $hash5$ in the DSA_2. If the verification succeeds, the SKUP is completed and place E4 reaches the subpage of substitution transition data to trigger the next data frame transmission, else place Reset5 is triggered.

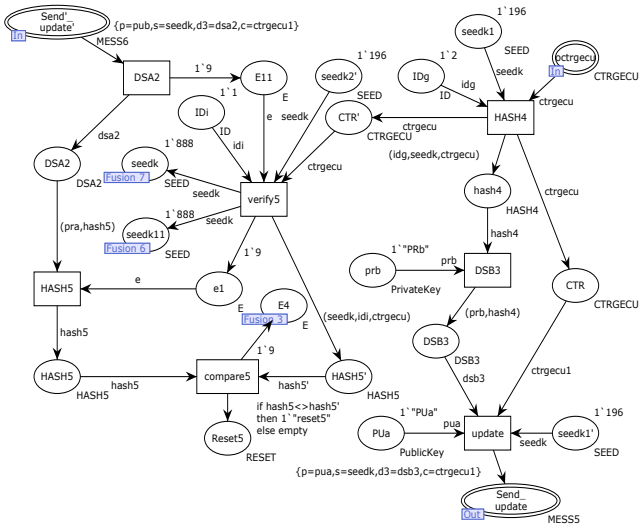


Figure 18: CPN model of new protocol's substitution transition update

The internal CPN model of the substitution transition update' is shown in Figure 19. Transition compare4 verifies $hash4$ in the DSB_3. If the verification succeeds, transition HASH5 combines ID_i, $ctrgecu$, and $Seed_k$ into $hash5$, else place Reset4 is triggered. Transition DSA2 combines $hash5$ and PR_a into DSA_2. Transition H combines PU_b, $ctrgecu$, $Seed_k$, and DSA_2, and sends them to GECU through the place Send_update'.

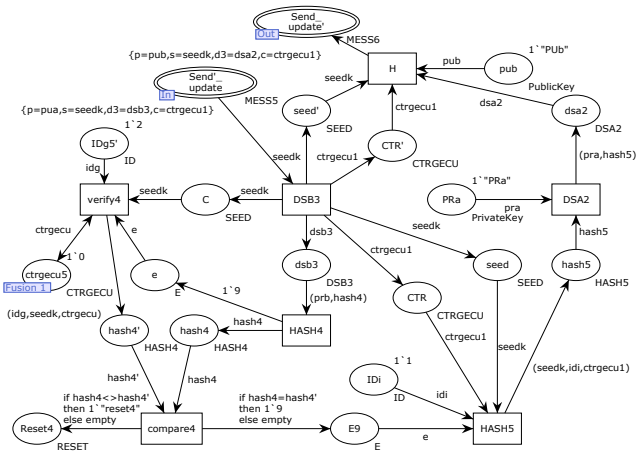


Figure 19: CPN model of new protocol's substitution transition update'

7 Formal Security Evaluation of the New Protocol

7.1 Introducing the Attacker Model

In the new protocol, the same method is used to introduce the Dolev-Yao attacker model. We launch man-in-the-middle attacks of replay, tampering, and spoofing to the network channel. The attacker model of the CPN model of the new protocol is shown in Figure 20. Inside the model, the blue type of places and transitions represent the replay attack, the red type of places and transitions represent the tampering attack, the purple type of places and transitions represent the spoofing attack.

7.2 Security Analysis of the New Protocol

The state space reports of the attacker model of the new protocol and the attacker model of the original protocol are shown in Table 3. There was no explosion of the state space indicates that the attacker model was effectively introduced. The number of dead markings in the attacker model of the new protocol is 1, which indicates that there is only 1 termination state after the data transmission is completed. It further shows that the new protocol has no other attack status and can resist man-in-the-middle attacks of replay, tampering, and spoofing.

Table 3: State Space reports of the attacker model of original protocol and new protocol

Type	Original Attack Model	New Attack Model
State Space Node	1288	2035
State Space Arc	3343	4974
SCC Graph Node	1288	2035
SCC Graph Arc	3343	4974
MainState Space Node	0	0
Live Transition Instances	0	0
Dead Transition Instances	0	0
Dead Marking	10	1

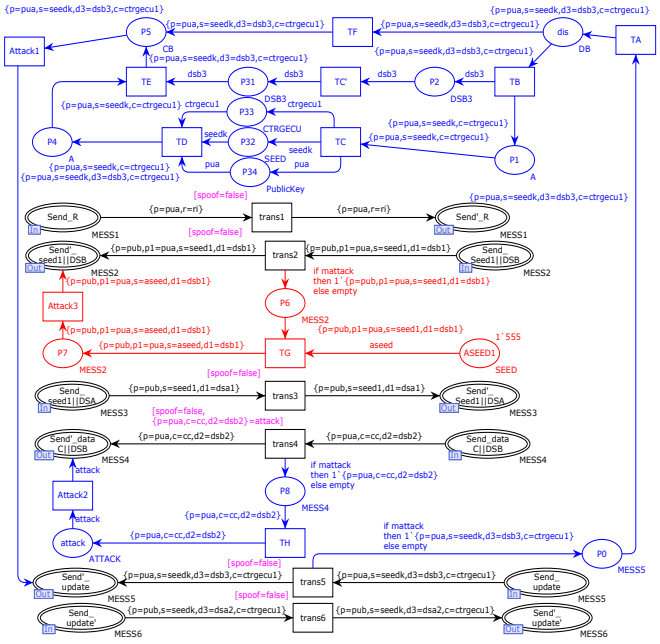


Figure 20: Formal description of new protocol based on the attacker model

By writing the ML program to further analyze, it is found that the session key of the new protocol ensures freshness and backward confidentiality, so it can resist replay attacks. The new protocol can resist spoofing attacks by authentication of digital signatures. The new protocol can resist tampering attacks by the HASH function to verify the correctness of the message during data transmission.

8 Comparison of Analysis Methods and Performance Analysis of the New Protocol

We compare our proposed protocol with other CAN2.0B-based IVN protocols with high security. The comparison of security attributes is shown in Table 4.

Our scheme introduces the asymmetric cryptosystem, improves the defect that the original protocol completely relies on the secure channel to distribute symmetric keys, eliminates the potential risks brought by traditional symmetric cryptography, and reduces the difficulty of key distribution and the complexity of key management. The new protocol cancels the HMAC and uses the HASH algorithm for message verification, which reduces the cost of computing while also ensuring security. For the interactive process, the state space of the new protocol increased but did not explode. It indicates that the new protocol increases computing time within an acceptable range to improve security.

To verify the effectiveness of the formal analysis in our proposed model detection method used in this paper, we compare it with the other effective model detection methods of IVN protocols. The results are shown in Table 5.

Woo *et al.* can only achieve verification of the functional correctness of the protocol. Basker *et al.* [13] use the Tamarin tool to formally analyze the CAN2.0B-based IVN protocol, which can achieve anomaly detection and verification of the functional correctness of the protocol. Joe *et al.* [6] verify the proposed IVN protocol through the AVISPA tool, which can verify the functional correctness of the protocol and has intuitiveness. Our proposed protocol model detection method can achieve anomaly detection and show the attack types of the man-in-the-middle attacks. The intuitive graphics can accurately describe the steps of the protocol. The state space generated by formal analysis is intuitive and can detect the security attributes of the protocol and verify the functional correctness of the protocol. This model detection method can be used for security analysis and research of other IVN protocols.

9 Conclusion

This paper proposes a model detection method based on a combination of CPN and Dolev-Yao attack models for formally security evaluation of the Woo *et al.* CAN2.0B-based IVN protocol, and shows that it cannot resist the two types of man-in-the-middle attacks: replay and spoofing. To solve them, we propose a new protocol to ensure

Table 4: Comparison of protocol security attributes

Security attributes	Ref [17]	Ref [13]	Ref [6]	Ref [14]	Ref [11]	Our Scheme
Session key security	No	Yes	Yes	Yes	Yes	Yes
Entity authentication	No	No	No	No	Yes	Yes
Resistance to reply attack	No	Yes	No	No	No	Yes
Resistance to tampering attack	Yes	Yes	Yes	Yes	Yes	Yes
Resistance to spoofing attack	No	No	No	No	Yes	Yes
Provable security	No	Yes	Yes	Yes	No	Yes
Formal verification	No	Yes	No	No	Yes	Yes

Table 5: Comparison of protocol analysis methods

Scheme	Anomaly detection	Attack type	Intuitive graphic	Verify functional correctness	State space
Ref [17]	No	No	No	Yes	No
Ref [13]	Yes	No	No	Yes	No
Ref [6]	No	No	Yes	Yes	No
Ref [14]	Yes	No	No	Yes	No
Ref [11]	Yes	No	No	Yes	No
Our Scheme	Yes	Yes	Yes	Yes	Yes

session key security attributes: backward confidentiality and entity authentication. Finally, we give a comparison of our protocol with Woo *et al.* protocol by the state space reports of CPN Tools and writing ML language. The results show that our protocol does not affect the computing time and storage cost, and provides robust security than Woo *et al.* protocol, and can resist the three types of man-in-the-middle attacks: replay, spoofing and tampering.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61762060).

References

[1] R. Amoah, S. Suriadi, S. Camtepe, and E. Foo, "Security analysis of the non-aggressive challenge response of the dnp3 protocol using a cpn model," in *ICC 2014 - 2014 IEEE International Conference on Communications*, 2014.

[2] I. Artamonov and A. Sukhodolov, "Cpn tools-based software solution for reliability analysis of processes in microservice environments," *International Journal of Simulation: Systems*, vol. 19, no. 6, 2019.

[3] L. L. Bello, R. Mariani, S. Mubeen, and S. Saponara, "Recent advances and trends in on-board embedded and networked automotive systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1038–1051, 2019.

[4] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, "Evaluation of can bus security challenges," *Sensors*, vol. 20, no. 2364, pp. 16–17, 2020.

[5] B. Groza and S. Murvay, "Efficient protocols for secure broadcast in controller area networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2034–2042, 2013.

[6] J. Halabi and H. Artail, "A lightweight synchronous cryptographic hash chain solution to securing the vehicle can bus," in *IEEE International Multidisciplinary Conference on Engineering Technology*, 2018.

[7] M. Inam, Z. Li, A. Ali, and A. Zahoor, "A novel protocol for vehicle cluster formation and vehicle head selection in vehicular ad-hoc networks," *International Journal of Electronics and Information Engineering*, vol. 10, no. 2, pp. 103–119, 2019.

[8] R. Islam and R. Refat, "Improving can bus security by assigning dynamic arbitration ids," *Journal of Transportation Security*, vol. 13, 2020.

[9] S. Jadhav and D. Kshirsagar, "A survey on security in automotive networks," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018.

[10] J. Lu, X. He, Y. Yang, D. Wang, and B. Meng, "Automatic verification of security of identity federation security protocol based on saml2.0 with proverif in the symbolic model," *International Journal of Network Security*, vol. 22, no. 1, pp. 80–92, 2020.

[11] S. Murvay and B. Groza, "Security shortcomings and countermeasures for the sae j1939 commercial vehi-

cle bus protocol," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2018.

[12] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1–1, 2015.

[13] B. Palaniswamy, S. Camtepe, E. Foo, and J. Pieprzyk, "An efficient authentication scheme for intra-vehicular controller area network," *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 1–1, 2020.

[14] A. I. Radu and F. D. Garcia, "Leia: A lightweight authentication protocol for can," *Springer International Publishing*, 2016.

[15] Ivan Edmar Carvajal Roca, Jian Wang, Jun Du, and Shuangqing Wei, "A semi-centralized security framework for in-vehicle networks," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 2020.

[16] M. Simon, D. Moldt, D. Schmitz, and M. Haustermann, "Tools for curry-coloured petri nets," *International Conference on Applications & Theory of Petri Nets & Concurrency*, 2019.

[17] S. Woo, H. J. Jo, and H. L. Dong, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 1–14, 2014.

[18] S. Woo, H. J. Jo, I. S. Kim, and D. H. Lee, "A practical security architecture for in-vehicle can-fd," *IEEE Transactions on Intelligent Transportation Systems*, pp. 2248–2261, 2016.

[19] L. Xiao, X Lu, T. Xu, W. Zhuang, and H. Dai, "Reinforcement learning-based physical-layer authentication for controller area networks," *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 1–1, 2021.

[20] L. Yao, J. Liu, D. Wang, J. Li, and B. Meng, "Formal analysis of sdn authentication protocol with mechanized protocol verifier in the symbolic model," *International Journal of Network Security*, vol. 20, no. 6, pp. 1125–1136, 2018.

[21] A Yg, A Jl, A Xl, A Yc, and J. B. Yan, "State space model identification of multirate processes with time-delay using the expectation maximization," *Journal of the Franklin Institute*, vol. 356, no. 3, pp. 1623–1639, 2019.

[22] C. Young, J. Zambreno, H. Olufowobi, and G. Bloom, "Survey of automotive controller area network intrusion detection systems," *IEEE Design and Test*, pp. 1–1, 2019.

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

Biography

Tao Feng received the D.E from Xidian Univerisity, China, in 2008. He is a member of the China Computer Federation and China Cryptography Federation. He is currently a professor and doctoral supervisor at the Faculty of Computer and Communication, Lanzhou University of Technology, China. His research interests are network and information security, industrial internet with focus on technologies for provable theory of security protocols, design and implementation of security middleware and cyberspace application system, privacy protection.

Lu Zheng received the B.E. degrees from Hefei University of Technology, China, in 2019. She is currently enrolled as M.E. student at the Faculty of Computer and Communication in the same institution. Her research interests are information security and provable theory of security protocols.

Peng-shou Xie is currently a professor and master supervisor at the Faculty of Computer and Communication, Lanzhou University of Technology, China. His research interest is security on internet of vehicles and industrial internet.

Improving the Efficiency of Point Arithmetic on Elliptic Curves Using ARM Processors and NEON

Pham Van Luc[1], Hoang Dang Hai[2], and Leu Duc Tan[3]
(Corresponding author: Pham Van Luc)

Faculty of Electronic Posts and Telecommunications Institute of Technology[1]
Km10, Nguyen Trai , Ha Dong District, Hanoi, Vietnam
Email: pvluc@bcy.gov.vn
Posts and Telecommunications Institute of Technology Ha Noi, Viet Nam[2]
Leu Duc Tan Academy of Cryptography Techniques Ha Noi, Viet Nam[3]
(Received Aug. 31, 2021; Revised and Accepted Jan. 13, 2022; First Online Feb. 14, 2022)

Abstract

Point arithmetic operations, especially scalar point multiplication, are important for cryptosystems on elliptic curves. These operations have a large computational overhead, which significantly affects the efficiency and speed of the cryptosystems. Several studies proposed methods to reduce the number of operations and the computational cost. However, few studies investigated hardware characteristics to improve the efficiency of point arithmetic operations by reducing the number of intemediate calculations. In this paper, we propose combining the Karatsuba algorithm with dual multiplications, which are performed in parallel on ARM processors with the NEON component. We propose some improved algorithms for point arithmetic operations by grouping pairs of multiplications or pairs of squarings to reduce intermediate calculations. Experimental results shown an efficiency increase of from 20% to 30% for point arithmetic operations (point addition, doubling and scalar point multiplication) and from 10% to 20% for the cryptographic primitive operations in ECDH and ECDSA protocols on ARMv7 and ARMv8 embedded microprocessors.

Keywords: ARM Processor; Elliptic Curve Cryptography; NEON Component; Point Addition; Point Doubling; Scalar Point Multiplication

1 Introduction

Arithmetic operations in the finite fields, especially scalar multiplication, play an important role in the elliptic curve cryptography (ECC) systems. Several standards have emerged, such as digital signature standard (DSS) [22], IEEE 1363 [27], and NIST [23], which recommend using finite fields for the digital signature algorithm on the elliptic curves. The most important ECC operation is the scalar point multiplication kP, where k is an integer, P is a point on the elliptic curve. To perform the multiplication, we can use the point addition and the point doubling multiplication. Point multiplications are more complex, thus, they take up the most computation time. The efficiency of the ECC cryptosystems mainly depends on the complexity and the speed of these operations [7, 14, 29]. The solution for these two opposing objectives is challenging because the efficiency depends further on the computing power of the deployment platform, especially for constrained hardware platforms with low-performance processors.

For improving the efficiency of the arithmetic operations on elliptic curves, theoretical studies often focused on: 1) reducing the number of the point addition and point doubling operations required in scalar multiplications [14]; 2) increasing the efficiency of the point addition and doubling formulas by exploiting the methods of point representation on elliptic curves [13]; 3) reducing the computational cost by improving the efficiency of the arithmetic operations [11, 20]. In fact, these methods can be combined together for obtaining a better performance on some hardware platform, such as in [4, 9, 16]. The Karatsuba (KA) [13] algorithm was the first to efficiently perform the multiplication of integers with a low computational complexity. However, for deploying on hardware platforms with limited processing capacity or on new hardware platforms, the algorithm needs to be improved for taking the full advantage as shown in some studies, such as [12, 16, 20, 24, 28].

In recent years, there is an increasing number of embedded microprocessors that provide single instruction multiple data (SIMD) capabilities to support parallel processing on dedicated modules. The commonly used ARM processor family is the Cortex-A architecture [1, 9, 17]. Most

Cortex-A architectures include the NEON component, which provides SIMD vector instructions. The NEON component allows parallel processing of SIMD instructions, resulting in an increased computation speed. A number of studies attempted to exploit the NEON instructions on ARM processors for increasing the computation speed of the implemented cryptographic algorithms, such as in [3, 5, 9, 26]. Some recent studies, such as in [11, 12, 15, 20], shown certain results by applying the SIMD architecture with NEON for implementing arithmetic operations on ECC. The NEON instructions, in particular the decomposition of instructions for performing parallel tasks in the SIMD architecture, enabled considerable advantages. However, several remaining issues are the cost of intermediate products, the challenge of handling propagation carries in prime fields, the problem of redundant calculations, etc. A large number of redundant intermediate calculations can cause a considerable degradation of the computation performance.

According to our survey, we see that the problem of reducing redundant intermediate calculations has not been fully investigated. There is a need for exploiting the specific features of the new hardware platform with NEON, such as ARM Cortex-A's, to reduce redundant instructions. This possibility can help to increase the computation speed, and therewith improves the efficiency of the arithmetic operations on elliptic curves. In particular, the combination of the well-known Karatsuba algorithm with ARM's advanced hardware platform can lead to a higher performance, but has not received much attention. The use of much read and write instructions between the memory and the NEON component can cause a considerable computational overhead, which can result in the performance degradation of the ECC cryptosystems.

This paper proposes a method to improve the efficiency of the point arithmetic operations (including the point addition, the point doubling, and scalar point multiplication) on elliptic curves. Our method focuses on gaining the SIMD hardware features of ARM processors with the integrated NEON component to speed up the point arithmetic operations. The key features of the method are: 1) combining the ordinary multiplication (the operand-scanning method) with the Karatsuba's multiplication algorithm for long operands; 2) implementing the parallel multiplications (the dual multiplications) on the prime fields in the combination with pairing to reduce the overhead of reading and/or writing data between the internal memory and the NEON component; 3) using an available large number library RELIC to speed up the computation. The proposed method has been fully integrated into the calculations of ECDH (Elliptic Curve Diffie Hellman) and ECDSA (Elliptic Curve Digital Signature Algorithm) protocols on GF(p) fields with sizes of 256 bits, 384 bits, and 521 bits.

The rest of the paper is structured as follows. Section 2 presents related studies. Section 3 gives a brief overview of elliptic curves over finite fields. Section 4 presents our methods to improve the efficiency of the point arithmetic. Experimental results are given in Section 5. Finally, Section 6 is the conclusion.

2 Related Work

In this section, we present some typical related studies, which apply the SIMD architecture and the NEON component in order to increase the speed of the arithmetic operations such as multiplications, squarings, and modulo operations in the finite fields on elliptic curves.

The authors in [3] suggested using NEON in the Cortex-A8 processor to speed up the computation of the shared secret key. A simplified radix representation method was used to perform NEON-based multiplications to speed up arithmetic operations on Curve25519 and Ed25519 curves. The NEON vector was used for two independent multiplications: a point multiplication as well as a single multiplication. The paper in [9] presented an algorithm to improve the efficiency of calculating the scalar multiplications on ECC with the focus on side-channel protection. Scalar multiplications were based on the proposed GLV (Gallant-Lambert-Vanstone) method, and used interleaved ARM-NEON instructions to perform 128-bit independent multiplications in parallel. The study in [26] implemented an attribute-based encryption scheme using Cortex-A9 and NEON. The authors proposed a method to exploit the ability to compute bilinear pairings on the ellictic curves. The authors in [5] presented a polynomial multiplier using NEON in the ARM Cortex-A8, A9, and A15 processors. The NEON vector was used to speed up the computation in the binary fields on the elliptic curves. The polynomial multiplier used two 8-bit vectors to form 128-bit products. These basic multipliers can be used in point multiplications. Some studies focused on the topic of implementing cryptographic primitive functions on the SIMD structure by using the NEON vector to improve the computational efficiency on ECC, such as [4,17]. In [4], a parallel version of the Montgomery interleaved multiplication algorithm was proposed using the extended vector instructions of SIMD and NEON. The Montgomery multiplication was splitted into two parallel operations. In [17], the authors proposed to use the interleaved ARM and NEON instructions to speed up the multiplications on $\mathbb{F}_{p^2}$, thereby they could speed up four-dimensional scalar multiplications on the Four$\mathbb{Q}$ twisted Edwards curves [17].

As we can be, using NEON to perform parallel operations can provide good efficiency. However, some typical problems are still remaining as follows. Most of the studies followed the approach of decomposing the required algorithm into steps for parallel execution using NEON in the SIMD architecture. This method has the disadvantage as most of the SIMD architectures (including NEON) do not support the propagation carries between data elements that are processed in parallel, especially when the operands are represented in a non-redundant form (with sufficient radix).Due to this limitation, the paper in [8]

provided a reduced radix representation method (the redundant form) to facilitate the handling of the carry propagations. However, as indicated in [25], this approach can lead to more intermediate products than the number of required operations. The reduced radix representation (the redundant representation) requires more multiplications than the canonical representation (the non-redundant representation). Several methods were proposed in [4, 25] to improve the performance by parallelizing parts in an operation with operands in the full radix form.

Another trend in applying NEON is to perform two operations in parallel on the finite fields. It is not necessary to handle the extra carry propagation during data parallelization. In [20], the authors presented the application of the NEON component in the dual multiplication to improve the efficiency of the Montgomery multiplication. However, the method in [20] used a lot of instructions to read and write data between the internal memory and NEON. The computational overhead for reading and writing data between the memory and the NEON component is often quite large. This problem has not been carefully considered in [20].

The studies in [4, 20, 25] focused on improving the Montgomery multiplication. This multiplication is said to be best for multiplying large numbers in a finite field. The multiplication of large numbers is more suitable for exponentiation operations (e.g, in RSA cryptosystems), but it is less useful for simple multiplications in ECC cryptosystems. The reason is that the Montgomery multiplication requires radix conversions. Radix conversions are often computationally expensive. Studies in [19] and [18] presented the possibility of improving the multiplication by decomposing the algorithm into parallel steps in a multiplication. The algorithm combined the Karatsuba multiplication with the NEON component in the SIMD architecture. However, the presented method only focused on the multiplication in the binary fields $GF(2)$, and did not deal with the carry propagation.

3 Elliptic Curves on Finite Fields

For the sake of clarity, this section presents a brief overview of the elliptic curves over finite fields. There are three basic coordinate forms that are commonly used. They are: 1) relative coordinates (Affine coordinates), 2) projective coordinates, and 3) compression coordinates. However, the relative coordinates and the projective coordinates are more commonly used [10, 14].

3.1 Affine Coordinates

Let $\mathbb{F}_p$ be a finite field with a prime number p, (E) be the elliptic curve over $\mathbb{F}_p$. A finite point P on (E) is defined by two elements, x and y, in $GF(p)$ that satisfy the equation of the curve:

$$(E):\ y^2 = x^3 + ax + b;\ a,\ b \in \mathbb{F}_q, 4a^3 + 27b^2 \neq 0\ (mod\ p).$$

x and y are called the relative coordinates (the Affine coordinate) of the point P. The point at infinity ∞ has no Affine coordinates. For the purpose of calculations, ∞ is often represented by a pair of coefficients (x, y) that do not belong to (E).

The grouping rule is often used to express the relationship between the points in a finite field, and is defined as follows.

Grouping Rule: Let E be an elliptic curve defined on the field $\mathbb{F}_p$ by the equation $y^2 = x^3 + Ax + B$ with $A, B \in \mathbb{F}_p$ and $4A^3 + 27B^2 \neq 0\ mod\ p$. Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be points on E with $P_1, P_2 \neq \infty$. The definition of $P_3 = (x_3, y_3)$ with $P_1 + P_2 = P_3$ is as follows:

1) If $x_1 \neq x_2$ then

$$x_3 = m^2 - x_1 - x_2, y_3 = m(x_1 - x_3) - y_1,$$

where $m = (y_2 - y_1)/(x_2 - x_1)$.

2) If $x_1 = x_2$ but $y_1 \neq y_2$, then

$$P_1 + P_2 = \infty$$

3) If $P_1 = P_2$ and $y_1 \neq 0$, then

$$x_3 = m^2 - 2x_1, y_3 = m(x_1 - x_3) - y_1,$$

where $m = (3x_1^2 + A)/(2y_1)$

4) If $P_1 = P_2$ and $y_1 = 0$, then

$$P_1 + P_2 = \infty$$

3.2 Projective Coordinates

To avoid the division in the fields, we suggest to represent the points in projective coordinates (coordinates in the fraction form). There are two basic types of projective coordinates, namely, the standard projective coordinates and the Jacobian projective coordinates.

In the standard projective coordinates, a point is represented as (X, Y, Z) with $Z \neq 0$, which is equivalent to the Affine coordinate of $(X/Z, Y/Z)$. The standard projective elliptic curve equation will have the following form:

$$Y^2Z = X^3 + aXZ^2 + bZ^3$$

In the Jacobian coordinates, a point (X, Y, Z) is equivalent to the point $(X/Z^2, Y/Z^3)$ in the Affine coordinates. The equation of curve (E) has the following form:

$$Y^2 = X^3 + aXZ^4 + bZ^6$$

A point with the Jacobian projective coordinates can be converted to the Affine coordinates according to the formula as follows:

$$(X, Y, Z) \rightarrow (x = X/Z^2, y = Y/Z^3).$$

Conversely, we can convert a point in the Affine coordinates to the Jacobian projective coordinates according to the formula as follows:

$$(x, y) \rightarrow (X = x, Y = y, Z = 1).$$

For additions and doublings of points in the projective coordinates, there is no need to use the inversions (divisions). The formula for the addition and doubling of two points in the projective coordinates can be obtained by converting the points to the Affine coordinates. After that, we apply the corresponding formula in the Affine coordinates, and finally, we remove the denominator of the formula.

3.3 Point Addition

Given 2 points $P(x_1 : y_1 : z_1)$, $Q(x_2 : y_2 : z_2)$ on the curve E with $P, Q \neq \infty$, $P \neq \pm Q$, we define the point $P + Q = (x_3 : y_3 : z_3)$ as follows:

Let

$$a = x_1 z_2^2, \quad b = x_2 z_1^2, \quad c = y_1 z_2^3$$
$$d = y_2 z_1^3, \quad e = b - a, \quad f = d - c$$

Then,

$$\begin{cases} x_3 = -e^3 - 2a \cdot e + f^2 \\ y_3 = c \cdot e^3 + f(a \cdot e^2 - x_3) \\ z_3 = z_1 \cdot z_2 \cdot e \end{cases}$$

3.4 Point doubling

Given a point $P(x_1 : y_1 : z_1)$ on the curve E with $P \neq \infty$, we define the point $[2]P = (x_3 : y_3 : z_3)$ as follows: Let

$$w_1 = 4x_1 \cdot y_1^2, w_2 = 3x_1^2 + A \cdot z_1^4$$

We have

$$\begin{cases} x_3 = -2w_1 + w_2^2 \\ y_3 = -8y_1^4 + w_2(w_1 - x_3) \\ z_3 = 2y_1 z_1 \end{cases}$$

The number of operations performed to add and double points in the coordinates is shown in Table 1 as follows (where I is the cost of the inverse operation and M is the cost of the multiplication).

Table 1: Number of operations performed to add and double points

Coordinates	Common addition	Doubling
Affine	1I,2M	1I,2M
Standard projective coordinates $(X/Z, Y/Z)$	$13M$	$7M$
Jacobian projective coordinates $(X/Z^2, Y/Z^3)$	$14M$	$5M$

3.5 Scalar Multiplications on Elliptic Curve

Scalar multiplications can be performed with several algorithms including:

1) right-to-left binary algorithm,
2) NAF (Non-Adjacent Form) algorithm,
3) NAF algorithm with sliding window.

4 A Method for Improving the Efficiency of Point Arithmetic Operations

In this section, we propose a method for improving the efficiency of the point additions and doublings on the Elliptic curve $E(\mathbb{F}_p)$ using dual multiplications in the prime field $\mathbb{F}_p$. The proposed method focuses on leveraging the SIMD hardware features of ARM processors, especifically of ARMv7 and ARMv8 [1] with the integrated NEON component to speed up arithmetic operations.

4.1 The Implementation Model

The model to perform multiplications is depicted in Figure 1. The basis of the model comprises: 1) the use of an available large number library, and 2) the implementation of parallel instructions.

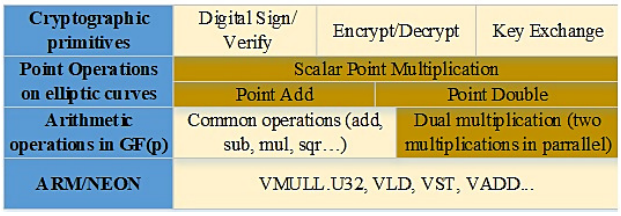


Figure 1: The implementation model

Most available large number libraries (e.g., RELIC, OpenSSL, MIRACL, GMP) represent large numbers in a non-redundant form with the radix 2^{32} or 2^{64} to match the basic multiplication support of the processors. The execution of two parallel multiplications has the advantage that we do not need to deal with the additional carry propagations in the case of integrating the proposed algorithms into the existing library. At the $GF(p)$ layer of the arithmetic operations, we propose to construct three dual multiplications that have the input operand of 256-bit, 384-bit, and 521-bit, respectively. At the next step, we integrate the dual multiplications into the point additions and point doublings by reorganizing the steps of the algorithm to build the pairs of the multiplications and squarings. Finally, we apply the NAF-based scalar multiplication algorithm to perform the scalar multiplications.

4.2 The Multiplication of Large Integers on $GF(p)$

Two basic methods are commonly used for the multiplication of large integers on $GF(p)$: 1) the operand scan-

ning method, and 2) the product scanning method. These methods differ in the way they handle the operands, and in the number of instructions for loading and storing the data needed in the calculation. For describing the mentioned methods, we use the following notations. Let A and B be two large integers of m-bit length stored in the array

$$A = (A[s-1], ..., A[2], A[1], A[0])$$
$$\text{and } B = (B[s-1], ..., B[2], B[1], B[0]).$$

We denote with w the number of bits of the word, which is usually chosen according to the processor types (e.g., 8, 16, 32, 64 bit). We denote with $s = \lceil m/w \rceil$ the number of words for representing the integers A and B. The result of the multiplication, i.e., $C = A * B$, is represented using the array

$$C = (C[2s-1], ..., C[2], C[1], C[0]).$$

4.2.1 Operand Scanning Method

The operand scanning method is the simplest method (also known as the common multiplication [21]) for performing multiplications of large numbers that have two operands A and B with s words. The method is implemented through two nested loops: the outer loop (the i loop) is for loading the values $A[i]$, while the inner loop (the j loop) is for loading the values B[j] and multiplying by $A[i]$, where $j = 0, ..., s-1$. The partial products are accumulated into the intermediate result column $C[i+j]$ along with the carry of the previous column. The formula for calculating the product of the components is as follows:

$$Carry, C[i+j]) = C[i+j] + A[i] * B[j] + Carry.$$

Figure 2 illustrates how to calculate the component products. The method uses a row-wise view, where the flow of computation is in the direction of the arrows.

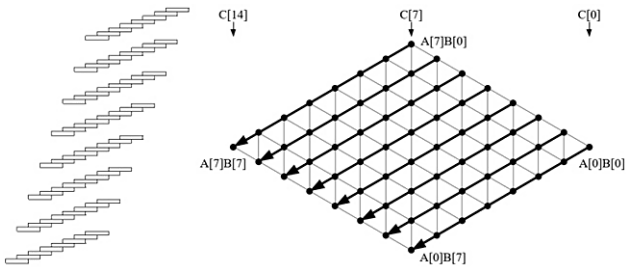


Figure 2: Multiplication of large numbers using the operand scanning method

The number of instructions for loading and saving data of this method is determined as follows:

- In each row: The number of load instructions in each inner loop is $2s$ (for loading data $B[j], C[i+j]$). The number of save instructions in each inner loop is s (to store $C[i+j]$). Thus, the total number of instructions in each row is $3s$.
- The outer loop needs s operands $A[i]$ for loading, and contains s values for carry, i.e., $C[i+s] = carry$. Thus, the total number of instructions is $2s$.

Therefore, the total cost of the operand scanning method is $2s^2+s$ for loading data, and is s^2+s for storing data, respectively. In total, this method takes $3s^2 + 2s$ instructions for loading and storing data. It is difficult to implement the parallelization of the algorithm, because the data depends on each other in a row-wise manner.

4.2.2 Product Scanning Method

The product scanning method implements the multiplications of large integers based on column-wise manner [6]. This method has the advantage of reducing the number of memory accesses. In cryptography, the number of columns to multiply does not exceed 2^w. We have: $s < 2^{3w}/2^{2w} = 2^w$, where w denotes the bit length of a word. Since the accumulator has a size of 3 words, it can contain the sum of all component multiplications of a column without having to contain intermediate results.

Figure 3 describes how to perform the product scanning multiplication. First, all operands of each column are multiplied by each other and their products are cumulatively added (i.e., using a cumulative multiplication method). After processing a column, the first word of the accumulator stored in the memory is a part of the final result. Therefore, no intermediate results are need for saving or loading in the algorithm. Furthermore, handling the carry propagations is quite easy, because the carries will be added to the result of the next columns. In addition, only five registers are required to perform the multiplication: two registers for storing the input operands, three registers for serving as the accumulators. This method is very suitable for the devices with limited resources. The formula for calculating products in the product scanning method is as follows:

$$C = A * B = \sum_{t=0}^{2s-2} (\sum_{i+j=t, 0 \le i,j \le s-1} A_i * B_j) W^t$$

where $W = 2^w$ is called the radix.

The rhombus in Figure 3 represents the process of calculating component products in the column-wise manner instead of the row-wise manner in the operand scanning method. In the product scanning method, only one storage operation is needed for storing the word of the final result. The cost of the entire multiplication is as follows:

- Because the outer loop has a size of $2s$ and the inner loop changes from 0 to s, the number of instructions for loading data (i.e., for loading $A[i]$, and $B[j]$ in each loop) is $2s^2$.
- The number of instructions for saving data is $2s$ (each step of the outer loop only needs to save one value).

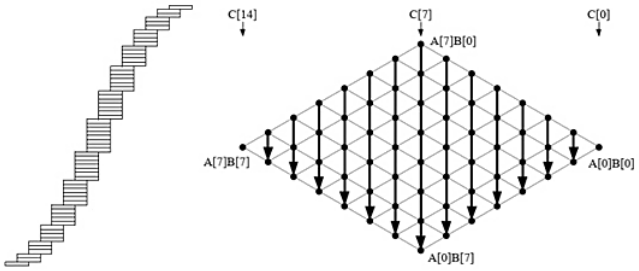


Figure 3: Multiplication of large numbers using the product scanning method

- The total number of instructions for loading and saving data is $2s^2 + 2s$.

The following table summarizes the cost of each method.

Table 2: Comparison of computational costs between multiplication methods

Methods	Number of instructions for loading data	Number of instructions for saving data	Total number of instructions
Operand scanning	$2s^2 + s$	$s^2 + s$	$3s^2 + 2s$
Product scanning	$2s^2$	$2s$	$2s^2 + 2s$

4.3 Parallelization of Two Multiplications on the Field GF(p)

The NEON component has sixteen 128-bit registers on the ARMv7 processors. Thus, we can directly implement the operand scanning method for multiplications (i.e., the common multiplications) with sizes of 256-bit and 384-bit. For 521-bit multiplications, we suggest using the conventional Karatsuba algorithm for the operations based on the combination of implementations in C and in NEON, since there are not enough registers to directly implement on NEON. On the ARMv8 processors, the NEON component has more registers (i.e., thirty-two 128-bit registers). However, we are able to use the same algorithm for both platforms, ARMv7 and ARMv8.

The modulo calculation algorithm uses the primitive algorithm as presented in [10]. The SIMD architecture has the feature of supporting two 32-bit multiplications using a single instruction. We apply this operation in Algorithm 1 as follow.

Figure 4 shows the parallel execution of two multiplications. The NEON instructions execute inside the loop of Step 4 in Algorithm 1 as shown in Figure 5.

Algorithm 1 Parallel multiplication of two multiplications on the field $GF(p)$

Input: $A = (A_{s-1}, ..., A_1, A_0)$, $B = (B_{s-1}, ..., B_1, B_0)$ and $C = (C_{s-1}, ..., C_1, C_0)$, $D = (D_{s-1}, ..., D_2, D_1, D_0)$.
Output: $M = (M_{2s-1}, \ldots, M_1, M_0) = A \cdot B$ and $N = (N_{2s-1}, .., N_1, N_0) = C \cdot D$.

1: $M = 0$, $N = 0$
2: **for** i=0 to s-1 **do**
3: $T_1 = 0$, $T_2 = 0$
4: **for** j=0 to s-1 **do**
5: $(T_1, S_1) = M_{i+j} + A_i \cdot B_j + T_1$, $(T_2, S_2) = M_{i+j} + C_i \cdot D_j + T_2$
6: $M_{i+j} = S_1$, $N_{i+j} = S_2$
7: **end for**
8: $M_{i+s} = T_1, N_{i+s} = T_2$
9: **end for**
10: **return** (M, N)
11: End

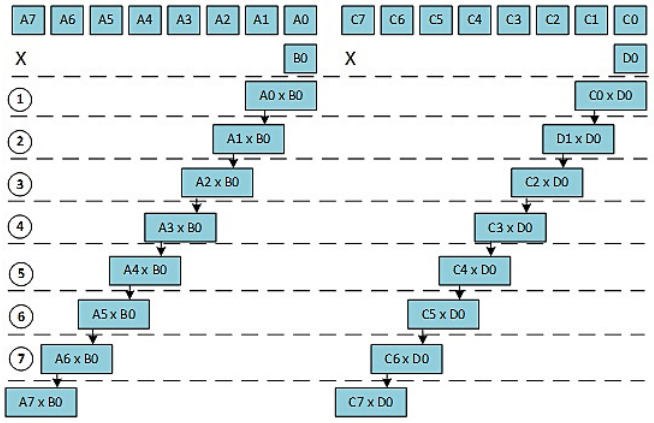


Figure 4: Parallel multiplication of two multiplications

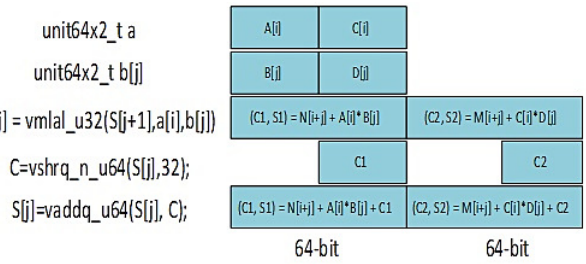


Figure 5: Calculations in the j-th loop of Algorithm 1 using NEON

As indicated in [21], the values (T_1, S_1) and (T_2, S_2) are represented within two words (for this case the radix is 2^{32}). Because it is able to perform two multiplications simultaneously, the NEON instructions are useful to speed up the computations of the arithmetic algorithms. However, it is expensive for loading and storing data between the NEON registers and the internal memory. Loading

and storing instructions are used much in the existing studies, as we pointed out in the section 2 of this paper. Such instructions can result in a high computational overhead. For example, data variables are continuously loaded into the NEON registers and removed from memory in two loops as shown in Algorithm 3 in the paper [20]. The mentioned method cannot achieve the Pipeline mechanism, and the performance of the parallel multiplication in NEON is significantly slowed down.

Therefore, this paper proposes to minimize the number of instructions for loading and storing data for leveraging the advantages of parallel computations in NEON. In this manner, our method differs from existing methods, e.g. the method presented in [20]. The method for restricting the number of instruction is described in detail as below.

4.3.1 Instruction Restriction for 256-bit and 384-bit Multiplications

Since it takes a long time for reading and writing data between the memory and the NEON registers (using vld and vst instructions), we read all inputs from the memory, and then apply Algorithm 1 for the computation. The result is written back from the NEON registers to the memory. This manner can minimize the time of reading and writing data between the memory and the NEON registers.

- For 256-bit multiplications: The 256-bit input operands are stored in an 8-word memory array, each word consists of 32 bits. Eleven 128-bit registers are required. Among them, 9 registers are for storing intermediate products and 2 registers are for storing temporary variables (e.g. carries, variables for getting the lower part of the data). In addition, 10 registers of 64-bit are needed, where 9 registers are for storing input terms and one register is for storing temporary variables.

- For 384-bit multiplications: The 384-bit input operands are stored in a 12-word memory array, each word consists of 32 bits. Fifteen 128-bit registers are required, where 13 registers are for storing intermediate products and 2 registers are for storing temporary variables. In addition, fourteen 64-bit registers are needed, where 13 registers are for storing the input terms and one register is for storing temporary variables.

4.3.2 Evaluating the Number of Instructions for Loading and Storing Data

Algorithm 3 in the study [20] presented the instructions for loading and storing data using the *neon_dual_mac*2 function. Because the dual multiplication is performed, the number of instructions is twice as many as the one in the operand scanning multiplication method. Thus, the total number of instructions for accessing the memory is required as $2 \times (3s^2 + 2s)$.

In our algorithm, i.e. Algorithm 1, the loop i and j only require to load the operands A, B, C, D, and then to write the final result (M, N) from the NEON registers to the memory. Thus, it takes 4s instructions for loading data and $2 \times 2s$ instructions for storing data.

The following table compares the number of data loading and saving instructions between our new proposed algorithm and Algorithm 3 presented in the study [20].

Table 3: Comparison of the number of instructions for data loading and saving between two algorithms

Methods	Number of instructions for loading data	Number of instructions for saving data	Total number of instructions for accessing memory
Algorithm 3 [20]	$2(2s^2 + 2)$	$2(s^2 + s)$	$6s^2 + 4s$
Algorithm 1 (256-bit and 384-bit multiplications)	$4s$	$4s$	$8s$

4.3.3 Instruction Restriction for 521-bit Multiplications

In the case of 521-bit multiplications, the 521-bit input operand is stored in a memory array of 17 words, each word consists of 32 bits. Twenty 128-bit registers are needed. However, the ARMv7's NEON component only has sixteen 128-bit registers, i.e., it has not enough registers for implementing the parallel multiplications. Therefore, we propose a combination mechanism that applies Karatsuba algorithm to perform 521-bit multiplications with the C/NEON programming language as follows.

Step 1. Implementing two additional multipliers, i.e., a dual multiplier for two 288-bit (9 words) operands and a dual multiplier for two 320-bit (10 words) operands. The procedure is similar to that for the dual multiplications of 256-bit operands as presented above.

Step 2. Applying the conventional Karatsuba algorithm with one level. The 1-level Karatsuba method is based on the paper [26]. We split the 521-bit operand (stored in seventeen 32-bit words) into 2 parts: The first part has 8 words and the second part has 9 words. Two common approaches can be used for implementing the Karatsuba algorithm, namely the additive and the subtractive algorithms. Suppose that, we want to multiply two pairs of operands, $M = A \cdot B$ and $N = C \cdot D$, where $A = A_H \cdot 2^{256} + A_L$, $B = B_H \cdot 2^{256} + B_L$ and $C = C_H \cdot 2^{256} + C_L$, $D = D_H \cdot 2^{256} + D_L$. The dual multiplications

$M = A \cdot B$ and $N = C \cdot D$ are calculated according to the following addition formula:

$$A_H \cdot B_H \cdot 2^{521} + [(A_H + A_L)(B_H + B_L) - A_H \cdot B_H - A_L \cdot B_L] \cdot 2^{256} + A_L \cdot B_L$$

$$C_H \cdot D_H \cdot 2^{521} + [(C_H + D_L)(C_H + D_L) - C_H \cdot D_H - C_L \cdot D_L] \cdot 2^{256} + C_L \cdot D_L.$$

The operands A_L, B_L, C_L, D_L have a length of 256-bits (8 words). Therefore, the pairs of multiplications $A_L \cdot B_L$ and $C_L \cdot D_L$ are performed using the 8-word dual multiplication algorithm that is inherited from the dual multiplier with 256-bit operand length. The operands A_H, B_H, C_H, D_H have a length of 288-bits (9 words). Thus,the pairs of multiplications $A_H \cdot B_H$ and $C_H \cdot D_H$ are performed using the 9-word dual multiplication algorithm by a dual multiplier with 288-bit operand length. The operands $(A_H + A_L)$, $(B_H + B_L)$, $(C_H + C_L)$, $(D_H + D_L)$ are 320-bits (10 words) long. Thus, the pairs the multiplications $(A_H + A_L)(B_H + B_L)$ and $(C_H + C_L)(D_H + D_L)$ are performed using the 10-word dual multiplication algorithm that uses a dual multiplier with 320-bit operand length.

Algorithm 2 Dual Karatsuba multiplications for 521-bit

Input: Four 17-word operands $A = A_H||A_L, B = B_H||B_L$ and $C = C_H||C_L, D = D_H||D_L$ (each word consists of 32 bits).

Output: $M = A \cdot B$ and $N = C \cdot D$ with the length of 34 words (each word consists of 32 bits).

1: $M_L = A_L \cdot B_L$ and $N_L = C_L \cdot D_L$ {NEON, 256bit}
2: $M_H = A_H \cdot B_H$ and $N_H = C_H \cdot D_H$ {NEON, 288bit}
3: $A_{HL} = (A_H + A_L)$, $B_{HL} = (B_H + B_L)$ {C, 320bit}
4: $C_{HL} = (C_H + C_L)$, $D_{HL} = (D_H + D_L)$ {C, 320bit}
5: $M_M = A_{HL} \cdot B_{HL}$ and $N_M = C_{HL} \cdot D_{HL}$ {NEON, 320bit}
6: $M = M_H \cdot 2^{521} + (M_M - M_H - M_L) \cdot 2^{256} + M_L$ {C}
7: $N = N_H \cdot 2^{521} + (N_M - N_H - N_L) \cdot 2^{256} + N_L$ {C}

4.4 Point Arithmetic Algorithms on Elliptic Curves

As presented in the study [2], the algorithms for adding and doubling points on the Jacobi coordinates use the multiplication and squaring operations. As we analyzed in the previous section, we can use the dual multiplication algorithms to perform the pairs of multiplications in parallel for the pairs of multiplications on F_p, whose data do not depend on each other... This principle can also be applied to the squarings. To simplify the implementation process, we do not implement the dual squarings (i.e., parallelizing two squarings). Instead, we use the dual multiplication to perform two squarings in parallel. In the following section, the paper presents several improvements for the point addition and doubling algorithms that originally presented in the paper [2].

4.4.1 Point Addition Algorithm

Algorithm 3 presented in the paper [2] is a point addition algorithm using the "add-2007-bl" formula with the sequential multiplication.

Algorithm 3 Point adding using the sequential multiplication algorithm [2]

Input: $P_1 = (X_1, Y_1, Z_1)$, $P_2 = (X_2, Y_2, Z_2)$ represented in Jacobi coordinates.

Output: $P_3 = P_1 + P_2 = (X_3, Y_3, Z_3)$.

1: $T_1 = Z_1^2$
2: $T_2 = Z_2^2$
3: $U_1 = X_1 \cdot T_1$
4: $U_2 = X_2 \cdot T_2$
5: $S_1 = Y_1 \cdot Z_2 \cdot T_2$
6: $S_2 = Y_2 \cdot Z_1 \cdot T_1$
7: $H = U_2 - U_1$
8: $I = (2 \cdot H)^2$
9: $J = H \cdot I$
10: $R = 2 \cdot (S_2 - S_1)$
11: $V = U_1 \cdot I$
12: $X_3 = R^2 - J - 2 \cdot V$
13: $Y_3 = R \cdot (V - X_3) - 2 \cdot S_1 \cdot J$
14: $Z_3 = ((Z_1 + Z_2)^2 - T_1 - T_2) \cdot H$

Since the point addition algorithm on the Jacobi coordinates can be decomposed into independent operations, we can apply the dual multiplication algorithm for this point addition algorithm. Thus, we propose to improve Algorithm 3 of the paper [2] by our Algorithm 4 using the dual multiplication that allows the parallel execution as follow. In Algorithm 4, the multiplications and squarings are organized into pairs of multiplications and pairs of squarings whose data are independent of each other. We can see that Algorithm 4 is equivalent to Algorithm 3, except that Algorithm 4 leverages the parallel execution.

4.4.2 Point Doubling Algorithm

Algorithm 5 described below is a point doubling algorithm using the formula "dbl-2001-b" as presented in the paper [2], which use the sequential multiplication.

Similar to point addition, the point doubling on the Jacobi coordinates can be decomposed into independent operations. Therefore, we can apply the dual multiplication algorithm for the point doubling algorithm. We propose to improve Algorithm 5 of the paper [2] by our Algorithm 6 using the double multiplication that allows the parallel execution as follow.

From the implementation of the steps, we can see that our Algorithm 6 is equivalent to Algorithm 5 of [2]. However, our Algorithm 6 differs from Algorithm 5 by leveraging the parallel execution.

Algorithm 4 Adding two points using SIMD-based dual multiplication algorithm

Input: $P_1 = (X_1, Y_1, Z_1)$, $P_2 = (X_2, Y_2, Z_2)$ represented in Jacobi coordinates.
Output: $P_3 = P_1 + P_2 = (X_3, Y_3, Z_3)$.

1:	$T_1 = Z_1^2$, $T_2 = Z_2^2$	{NEON}
2:	$U_1 = X_1 \cdot T_1$, $U_2 = X_2 \cdot T_2$	{NEON}
3:	$S_1 = Z_2 \cdot T_2$, $S_2 = Z_1 \cot T_1$	{NEON}
4:	$S_1 = Y_1 \cdot S_1, S_2 = Y_2 \cdot S_2$	{NEON}
5:	$H = U_2 - U_1$	{C}
6:	$I = 2 \cdot H$	{C}
7:	$I = I^2, T_3 = R^2$	{NEON}
8:	$J = H \cdot I$, $V = U_1 \cdot I$	{NEON}
9:	$R = 2 \cdot (S_2 - S_1)$	{C}
10:	$X_3 = T_3 - J - 2 \cdot V$	{C}
11:	$T_3 = V - X_3$	{C}
12:	$T_3 = R \cdot T_3$, $T_4 = S_1 \cdot J$	{NEON}
13:	$Y_3 = T_3 - 2 \cdot T_4$	{C}
14:	$Z_3 = ((Z_1 + Z_2)^2 - T_1 - T_2) \cdot H$	{C}

Algorithm 5 Point doubling using the sequential multiplication algorithm [2]

Input: $P_1 = (X_1, Y_1, Z_1)$ represented in Jacobi coordinates.
Output: $P_3 = 2 \cdot P_1 = (X_3, Y_3, Z_3)$.

1:	$T_0 = delta = Z_1^2$
2:	$T_1 = gamma = Y_1^2$
3:	$T_2 = beta = X_1 \cdot T_1$
4:	$T_3 = X_1 - T_0$
5:	$T_4 = X_1 + T_0$
6:	$T_3 = alpha = 3 \cdot T_3 \cdot T_4$
7:	$X_3 = T_3^2 - 8 \cdot T_2$
8:	$Z_3 = (Y_1 + Z_1)^2 - T_1 - T_0$
9:	$Y_3 = T_3 \cdot (4 \cdot T_2 - X_3) - 8 \cdot T_1^2$

Algorithm 6 Point doubling using the sequential multiplication algorithm [2]

Input: $P_1 = (X_1, Y_1, Z_1)$ represented in Jacobi coordinates.
Output: $P_3 = 2 \cdot P_1 = (X_3, Y_3, Z_3)$.

1:	$T_0 = delta = Z_1^2, T_1 = gamma = Y_1^2$	{NEON}
2:	$T_3 = X_1 - T_0$	{C}
3:	$T_4 = X_1 + T_0$	{C}
4:	$T_2 = beta = X_1 \cdot T_1$, $T_3 = T_3 \cdot T_4$	{NEON}
5:	$T_3 = alpha = 3 \cdot T_3$	{C}
6:	$T_5 = Y_1 + Z_1$	{C}
7:	$T_4 = T_3^2$; $T_5 = T_5^2$	{NEON}
8:	$X_3 = T_4 - 8 \cdot T_2$	{C}
9:	$Z_3 = T_5 - T_1 - T_0$	{C}
10:	$T_4 = 4 \cdot T_2 - X_3$	{C}
11:	$T_4 = T_3 \cdot T_4$, $T_5 = T_1^2$	{NEON}
12:	$Y_3 = T_4 - 8 \cdot T_5$	{C}

4.4.3 Comparison of the proposed algorithms with the original algorithms

For the comparison of our parallel algorithm with the sequential algorithm [2], we use the following notations. We denote M, S as the cost of the conventional multiplication and squaring, and Mt, St as the cost of the dual multiplication according to the proposed algorithms.

The cost of the point addition according to the sequential model [2] (i.e., Algorithm 3) is $11M + 5S$, while the cost of our proposed addition algorithm (i.e., Algorithm 4) is $5Mt + 2St + 1M + 1S$. Since $Mt < 2M$ and $Mt < 2S$, we can see that the cost of the proposed algorithm (i.e., Algorithm 4) is more efficient than the cost of the point addition algorithm presented in [2] (i.e, the Algorithm 3).

Table 4: Comparison of the costs of the algorithms

Algorithm	Cost of the point operation	
	Cost of the point addition	Cost of the point doubling
Sequential algorithm [2]	$11M + 5S$	$3M + 5S$
Our proposed parallel algorithm	$5Mt + 2St + 1M + 1S$	$4Mt$

For the point doubling algorithms, the cost of the sequential point doubling [2] (i.e., Algorithm 5) is 3M+5S, while the cost of the proposed point doubling (i.e., Algorithm 6) is 4Mt. Thus, the cost of the proposed algorithm (i.e., Algorithm 6) is more efficient than the cost of the point doubling algorithm presented in [2] (i.e., Algorithm 5).

Table 4 shows the comparison of the costs of the mentioned algorithms.

4.4.4 Scalar Point Multiplication

For the scalar point multiplication, we use the sliding window NAF algorithm to multiply a positive integer by a point. In this scalar point multiplication, our algorithm uses the point addition and the point doubling according to Algorithms 4 and 6 based on the SIMD dual multiplication.

5 Experiments and Evaluations

5.1 Experiment Settings

The following tests are performed on ARMv7 and ARMv8 processors with the integrated NEON component. The NEON [1] instructions are used to perform two multiplications in parallel including:

- *vmlal_u32* (for multiplying and accumulating unsigned integers in the vector form):
  For $Q0 = vmlal\_u32(Q0, D2, D3[0])$, we calculate

$$D1 = D1 + D2[1] \times D3[0] \text{ and}$$

$$D0 = D0 + D2[0] \times D3[0].$$

- $vshrq\_n\_u64$ (to right shift the vector by n bits)):
  For $Q0 = vshrq\_n\_u64(Q0, 32)$, we calculate

$$D0 = D0 \gg 32 \text{ and } D1 = D1 \gg 32.$$

- $vaddq\_u64$ (to perform addition in vector form):
  For $Q0 = vaddq\_u64(Q0, Q1)$, we calculate

$$D0 = D0 + D2 \text{ and } D1 = D1 + D3.$$

- $vandq\_u64$ (to perform AND operation in vector form):
  For $Q0 = vandq\_u64(Q0, Q1)$, we calculate

$$D0 = D0\&D2 \text{ and } D1 = D1\&D3.$$

- $vmovn\_u64$ (to transfer vector data in narrow mode)
  For $D0 = vmovn\_u64(Q1)$, we calculate

$$D0[0] = D2[0] \text{ and } D0[1] = D3[0].$$

- $vld1\_u32$ (to load data from memory into registers in NEON)
  For $D0 = vld1\_u32([N])$, we calculate

$$D0[0] = [N] \text{ and } D0[1] = [N + 1].$$

- $vget\_lane\_u32$ (to copy data from NEON registers to memory)
  For $R1 = vget\_lane\_u32(D0, 0)$, we calculate

$$R1 = D0[0].$$

  For $R2 = vget\_lane\_u32(D0, 1)$, we calculate

$$R2 = D0[1].$$

Based on the above instructions, we implement the dual multiplication algorithm, the point addition algorithm, and the point doubling algorithm as described in the section 4 based on the RELIC cryptographic library version 0.5.0. The RELIC is a cryptographic library that is considered to have a fairly fast execution speed among other open source cryptographic libraries such as OpenSSL and GMP. Next, we performed experiments and evaluated the results between the library that integrates the proposed improvements and the original library that uses the default algorithm. To measure the time of an operation, we take the average number of 100 executions of that operation.

5.2 Experiments on ARMv7

The following experiment results are performed on the hardware platform: a Xilinx Zynq Kit with the ARMv7 1.3 GHz processor running an embedded Linux operating system. The development tool used for compiling the program is arm-xilinx-linux-gnueabi-gcc version 4.8.3. We performed tests for the proposed multiplication algorithm on three curves: Curve NIST-P256, Curve NIST-P384, and Curve NIST-P521.

The experiment results with ARMv7 are presented in Tables 5, 6, and 7. Table 5 shows the results of performing the arithmetic operations on F_p. Table 6 presents the results for performing the point arithmetic operations on the curve $E(F_p)$. Table 7 depicts the results with the cryptographic primitive functions based on the $E(F_p)$ curves.

Table 5 shows the comparative evaluation test between the time for performing a dual multiplication and the time for performing two multiplications (Mul) or two squarings (Sqr) sequentially (i.e., the default operation using the RELIC library). The last column indicates the ratio (i.e., the efficiency). As shown in Table 5, the proposed algorithm is faster than the default algorithms using the RELIC library, namely 30% and 20% faster for the multiplications and squarings, repectively.

Table 6 presents the comparative evaluation test between the execution time of the point addition (Add), the point doubling (Dbl) and the scalar point multiplication ($k \cdot P$) of the proposed algorithm with that of the default operations using the RELIC library. The results in the last column show that, the proposed algorithm (i.e., using the NEON component) provides results of 20% to 30% faster than the default algorithm using the RELIC library.

Table 7 shows the performance evaluation for two cryptographic protocols (ECDH, ECDSA) using our proposed algorithm (i.e., the algorithm based on NEON) and the algorithm using the original RELIC library. On the ARMv7 platform, the performance of the proposed algorithms for ECDH and ECDSA protocols increases by 10% to 20% compared to the original algorithms.

5.3 Experiments on ARMv8

The following experiment results are performed on the hardware platform: a NXP IMX8M Kit with the ARMv8 1.5 GHz processor running an Android 10 operating system. We use the Android development tool NDK (Standalone toolchains) to compile the programs with GCC compiler version 6.3. We performed tests for the proposed multiplication algorithm on three curves: Curve NIST-P256, Curve NIST-P384, and Curve NIST-P521.

The experiment results with ARMv8 are presented in Tables 8, 9, and 10. Table 8 shows the results for performing the arithmetic operations on F_p. Table 9 presents the results for performing the point arithmetic operations on the curve $E(F_p)$. Table 10 indicates the results with the cryptographic primitive functions based on the $E(F_p)$ curves.

Table 8 shows the comparative evaluation test between the time for performing a dual multiplication and the time for performing two multiplications (Mul) or two squarings (Sqr) sequentially (i.e., the default operation using the RELIC library). The results in the last column show

Table 5: Arithmetic operations on F_p with ARMv7

Bit Length	Arithmetic operations on F_p	Time (10^3 nanoseconds)		Ratio
		Dual multiplication	Sequential multiplication	
256	Mul	4.9	6.9	0.71
	Sqr	4.9	6.3	0.78
384	Mul	9.2	13.5	0.68
	Sqr	9.2	13.0	0.71
521	Mul	19.6	28.6	0.69
	Sqr	19.6	28	0.70

Table 6: Point arithmetic operations on the curve $E(F_p)$ with ARMv7

Bit Length	Arithmetic operations on F_p	Time (10^3 nanoseconds)		Ratio
		Dual multiplication	Sequential multiplication	
256	Add	43	57.2	0.75
	Dbl	21.8	30.3	0.72
	$k \cdot P$	8742	10770	0.81
384	Add	81.5	107.8	0.76
	Dbl	40.6	57.3	0.71
	$k \cdot P$	23672	29476	0.80
521	Add	176.5	238.5	0.74
	Dbl	87.8	125.2	0.70
	$k \cdot P$	68243	85689	0.80

Table 7: $E(F_p)$-based cryptographic primitives with ARMv7

Bit Length	Protocols based on $E(F_p)$	Time (10^3 nanoseconds)		Ratio
		SIMD	Sequential multiplication	
256	ECDH	9.0	11.2	0.8
	ECDSA Sig	4.9	5.3	0.92
	ECDSA Ver	12.6	14.7	0.86
384	ECDH	23.4	29.3	0.80
	ECDSA Sig	12.6	13.6	0.93
	ECDSA Ver	32.5	37.3	0.87
521	ECDH	69.4	86.9	0.80
	ECDSA Sig	36.1	39.3	0.92
	ECDSA Ver	74.8	85.0	0.88

Table 8: Arithmetic operations on F_p with ARMv8

Bit Length	Arithmetic operations on F_p	Time (10^3 nanoseconds)		Ratio
		Dual multiplication	Sequential multiplication	
256	Mul	1.8	2.9	0.62
	Sqr	1.8	2.4	0.75
384	Mul	3.6	6.1	0.60
	Sqr	3.6	5.1	0.71
521	Mul	8.4	13.8	0.61
	Sqr	8.4	11.0	0.76

that, the proposed algorithm is faster than the default algorithm using the RELIC library, namely 40% and 25% faster for the multiplications and squarings, repectively.

Table 5 presents the comparative evaluation test between the execution time of the point addition (Add), the point doubling (Dbl) and the scalar point multiplication ($k \cdot P$) of the proposed algorithm with that of the default operations using the RELIC library. The results in the last column show that, the proposed algorithm (i.e. the algorithm using the NEON component) is about 20% to 30% faster than the default algorithm using the RELIC library.

Table 9: Point arithmetic operations on the curve $E(F_p)$ with ARMv8

Bit Length	Arithmetic operations on F_p	Time (10^3 nanoseconds)		Ratio
		Dual multiplication	Sequential multiplication	
256	Add	16.2	22.8	0.71
	Dbl	8.4	12.2	0.69
	$k \cdot P$	3376	4149	0.81
384	Add	32.7	47.6	0.69
	Dbl	17.5	24.0	0.73
	$k \cdot P$	9645	12409	0.78
521	Add	74.9	100.8	0.74
	Dbl	37.6	50.3	0.75
	$k \cdot P$	28696	34825	0.82

Table 10 shows the evaluation of the performance of two cryptographic protocols (ECDH, ECDSA) using our proposed algorithm (based on NEON) and the original protocol implemented in the RELIC library. On the ARMv8 platform, the performance of the proposed algorithms for ECDH and ECDSA protocols increases by 10% to 20% compared to the original algorithms.

Table 10: $E(F_p)$-based cryptographic primitives with ARMv8

Bit Length	Protocols based on $E(F_p)$	Time (10^3 nanoseconds)		Ratio
		SIMD	Sequential multiplication	
256	ECDH	3.4	4.2	0.81
	ECDSA Sig	1.8	2.0	0.90
	ECDSA Ver	4.7	5.5	0.85
384	ECDH	9.7	12.6	0.77
	ECDSA Sig	5.1	5.8	0.88
	ECDSA Ver	13.0	16.0	0.81
521	ECDH	28.6	34.8	0.82
	ECDSA Sig	14.7	16.0	0.92
	ECDSA Ver	37.1	42.9	0.86

6 Conclusions

Techniques for improving the efficiency of the point arithmetic operations, especially the computational speed, is always a challenge for the elliptic curve cryptosystems using hardware platforms with low processing power. The traditional Karatsuba algorithm allows multiplying integers with low computational complexity and fast speed. However, it is difficult to implement the algorithm on processors with the limited performance. Several studies focused on reducing the number of operations, and thus, the cost of computation. The application of the SIMD architecture with the NEON component can provide the possibility for calculations in parallel. In addition, we can leverage the hardware characteristics to reduce the number of intermediate calculations, as well as combine the Karatsuba algorithm with advanded hardware platforms such as ARM. However, this approach has not yet been fully investigated.

This paper proposes a method for improving the efficiency of the arithmetic operations including the point addition, doubling and scalar point multiplication on elliptic curves using the ARM hardware characteristics and the NEON. In this method, we combine the operand scanning multiplication with the Karatsuba algorithm. The proposed method performs the parallel multiplication (the dual multiplications) together with the multiplication pairing to minimize the cost of data read/write operations between the memory and the NEON. Furthermore, we use an available large number library to speed up the computation. We proposed several algorithms to implement the method on ARMv7 and ARM v8 embedded processors. The algorithms are fully integrated into the computations of the EDCH and ECDSA protocols on the $GF(p)$ fields with sizes of 256, 384, and 521 bits. Experimental results shown that the efficiency of the proposed algorithms increases from 20% to 30% for the basic operations (including the point addition, doubling, and scalar point multiplication), and increases from 10% to 20% for the calculations in the ECDH and ECDSA protocols in comparison with the previous method.

Further extensions of this work can be the improved parallelization by combining the interleavedg ARM instructions and the SIMD instructions, or the adaption of the algorithms to the other ARM-Cortex processor families.

References

[1] ARM, "Series programmer's guide," Technical Report Cortex-A, 2012.

[2] D. J. Bernstein, T. Lange, and *et al.*, "Explicit-formulas database," Jan. 27, 2022. (`https://hyperelliptic.org/EFD/`)

[3] D. J. Bernstein and P. Schwabe, "Neon crypto," in *Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems (CHES'12)*, pp. 320–339, Springer, 2012.

[4] J. W. Bos, P. L. Montgomery, D. Shumow, and G. M. Zaverucha, "Montgomery multiplication using vector instructions," in *Proceedings of Conference Selected Areas in Cryptography (SAC'13)*, pp. 471–489, Springer, 2013.

[5] D. Câmara, C. P. Gouvêa, J. López, and R. Dahab, "Fast software polynomial multiplication on arm processors using the neon engine," in *Proceedings of International Conference on Security Engineering and Intelligence Informatics*, pp. 137–154, Springer, 2013.

[6] P. G. Comba, "Exponentiation cryptosystems on the ibm pc," *IBM Systems Journal*, vol. 29, no. 4, pp. 526–538, 1990.

[7] M. S. Hwang, S. F. Tzeng, C. S. Tsai, "Generalization of proxy signature based on elliptic curves", *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 73–84, 2004.

[8] Intel Corporation, "Using streaming simd extensions (SSE2) to perform big multiplications," Technical Report Application note AP-941, July 2000.

[9] A. Faz-Hernandez, P. Longa, and A. H. Sanchez, "Efficient and secure algorithms for GLV-based scalar multiplication and their implementation on GLV-GLS curves," in *Proceedings of RSA Conference: Topics in Cryptology CT-RSA*, pp. 1–27, Springer, 2014.

[10] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Berlin: Springer-Verlag, 2004.

[11] H. Cheng, J. Grosschaedl, J. Tian, P. B. Ronne, and P. Y. Ryan, "High-throughput elliptic curve cryptography using avx2 vector instructions," in *International Conference on Selected Areas in Cryptography (SAC'20)*, pp. 698–719, Springer, 2020.

[12] C. Y. Lee, C. C. Fan, J. Xie, and S. M. Yuan, "Efficient implementation of karatsuba algorithm based three-operand multiplication over binary extension field," *IEEE Access*, vol. 6, pp. 38234–38242, 2018.

[13] S. G. Liu, S. J. An, and Y. W. Du, "Efficient and secure elliptic curve scalar multiplication based on quadruple-and-add," *International Journal of Network Security*, vol. 23, no. 5, pp. 750–757, 2021.

[14] S. G. Liu, Y. Y. Hu, and L. Wei, "Elliptic curve scalar multiplication algorithm based on side channel atomic block over GF(2^m)," *International Journal of Network Security*, vol. 23, no. 6, pp. 1005–1011, 2021.

[15] Z. Liu, J. Groschdl, Z. Hu, K. Jrvinen, H. Wang, and I. Verbauwhede, "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the internet of things," *IEEE Transaction on Computers*, vol. 66, no. 5, p. 773–785, 2017.

[16] L. Kowada, R. Portugal, and C. M. H. Figueiredo, "Reversible karatsuba's algorithm," *Journal of Universal Computer Science*, vol. 12, no. 5, pp. 499–511, 2006.

[17] P. Longa, "Fourqneon: Faster elliptic curve scalar multiplications on arm processors," in *Proceedings of Conference Selected Areas in Cryptography (SAC'16)*, pp. 501–519, Springer, 2016.

[18] P. V. Luc, H. D. Hai, and L. D. Tan, "Multilayer multiplication in binary field on armv8 processors," in *Proceedings of IEEE International Conference on Advanced Technologies for Communications (ATC'20)*, Nhatrang, Vietnam, 2020.

[19] P. V. Luc, V. T. Linh, H. D. Hai, and L. D. Tan, "Fast binary field mutiplication on armv7 embedded processors," in *Proceedings of 4th IEEE International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom'20)*, Hanoi, Vietnam, 2020.

[20] R. C. Marquez, A. J. C. Sarmiento, and S. Sánchez-Solano, "Speeding up elliptic curve arithmetic on arm processors using neon instructions," *RIELACy*, vol. 41, no. 3, pp. 1–20, 2020.

[21] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[22] National Institute of Standards and Technology (NIST), "DSS digital signature standard (DSS)," Technical Report Federal Information Processing Standards Publication 186-2, 2000.

[23] National Institute of Standards and Technology (NIST), "Recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography," Technical Report SP 800-56A Rev. 3, Apr. 2018.

[24] H. Seo, Z. Liu, J. Choi, and H. Kim, "Karatsuba-block-comb technique for elliptic curve cryptography over binary fields," *Journal of Security and Communication Networks*, vol. 8, no. 17, pp. 3121–3130, 2015.

[25] H. Seo, Z. Liu, J. Grobschadl, J. Choi, and H. Kim, "Montgomery modular multiplication on arm-neon revisited," in *Proceedings of International Conference on Information Security and Cryptology (ICISC'14)*, pp. 328–342, Springer, Cham, 2015.

[26] A. H. Sánchez and F. Rodríguez-Henríquez, "Neon implementation of an attribute based encryption scheme," in *Proceedings of International Conference on Applied Cryptography and Network Security, (ACNS'13)*, pp. 322–338, Springer, 2013.

[27] IEEE Standards, "Standard specifications for public key cryptography," Technical Report of IEEE 1363-2000, 2004.

[28] A. Weimerskirch and C. Paar, "Generalizations of the karatsuba algorithmfor efficient implementations," Technical Report of University of Ruhr, Bochum, Germany, 2003.

[29] C. C. Yang, T. Y. Chang, M. S. Hwang, "A new anonymous conference key distribution system based on the elliptic curve discrete logarithm problem", *Computer Standards and Interfaces*, vol. 25, no. 2, pp. 141–145, 2003.

Biography

Pham Van Luc received master's degree in cryptographic techniques in 2008. His research interests include Analysis and Design of Security Protocols, and Applied Cryptography.

Dang Hai Hoang, Dr.-Ing. (1999), Dr.-Ing.habil. (2002) in telematics and communication systems from Technical University of Ilmenau, Germany; Associate Professor at Posts and Telecommunications Institute of Technology, Hanoi, Vietnam. His current research interests include information security, communication protocols, communication systems, QoS mechanisms, and control systems..

Leu Duc Tan received his PhD in cryptographic techniques in 1992. His research interests include Analysis and Design of Security Protocols, and Cryptographic Theory.

BioHashing Speech Security Retrieval Algorithm Based on MSCC and Improved Hadamard Measurement Matrix

Yi-Bo Huang[1], Yuan Zhang[1], and Qiu-Yu Zhang[2]
(Corresponding author: Yi-Bo Huang)
College of Physics and Electronic Engineering, Northwest Normal University[1]
No. 967, An-ning East Road, Lanzhou 730070, China
Email: huang_yibo@nwnu.edu.cn
School of Computer and Communication, Lanzhou University of Technology[2]
No.287, Lan-Gong-Ping Road, Lanzhou 730050, China
(Received Sept. 18, 2021; Revised and Accepted Jan. 22, 2022; First Online Feb. 23, 2022)

Abstract

To solve the problem of plaintext data leakage in the existing speech retrieval system, to improve the performance of speech retrieval and the security of biometric templates, this paper proposes a BioHashing speech security retrieval algorithm based on MSCC(Mel S-Transform Cepstrum Coefficients) and improved Hadamard measurement matrix. First, the pre-processed speech clips calculate the Euclidean distance of MSCC to construct biometric speech vectors and classify them by the K-medoids algorithm. Then, the dimension vectors of the orthogonal set matrix, constructed by Schmidt orthogonalization of the improved Hadamard measurement matrix, corresponding to the biometric vectors of the corresponding class to form the diversity biosafety templates, and the templates are further quantified into BioHashing sequences. Finally, the BioHashing sequences were scrambling encrypted to construct the hash index and stored in the cloud. At the same time, the ciphertext speech clips generated by multi-key DES(Data Encryption Standard) encryption algorithm based on CBC(Cipher Block Chaining) mode are uploaded to the cloud. This paper adopts the normalized Hamming distance for matching and retrieval within the class to facilitate the query users to match and retrieve the query speech. The experimental results show that the biosafety template of the proposed algorithm makes the biometric template have better security, and the encryption algorithm can effectively prevent the leakage of private data. In addition, this paper uses. Euclidean distance of MSCC has better discrimination, robustness, and retrieval performance for the retrieval system.

Keywords: BioHashing; Hadamard Measurement Matrix; MSCC; Speech Security Retrieval

1 Introduction

With the advent of the big data era and cloud computing, the amount of multimedia applications has increased exponentially. As a multimedia tool, speech contains a large amount of private information. In order to ensure the security of sensitive data during transmission and storage, traditional speech retrieval is no longer suitable for current needs, such as retrieval based on watermark [12] and retrieval based on digital signature [7]. In practical applications, how to achieve fast and accurate ciphertext speech retrieval, how to improve the diversity of biometric templates and how to protect the safety of biometric templates have attracted widespread attention in the field of modern speech retrieval [10].

At present, the research of speech retrieval is mainly divided into two categories: text or keyword retrieval [1] and content retrieval [2], in which content retrieval is divided into biometric matching [16], deep learning [25] and sorting retrieval [11]. Biometric extraction is an important process of speech retrieval, with most studies using perceptual hash [26] or audio fingerprint [14]. In addition, in order to improve the security of the system, some good methods are also applied, such as biometric change [6], quantum hash [13], BioHashing [24]. Ref. [23] proposed a perceptual hash scheme based on band variance. This scheme realized the basic biometric of perceptual hash, and the algorithm had better discrimination and robustness, but the retrieval accuracy of the algorithm was reduced. In order to compress the storage capacity of data, Ref. [4] used piecewise aggregation approximation technology in the process of constructing perceptual hash and multifractal biometrics to improve the relative stability of the algorithm, but the retrieval efficiency of the algorithm was low. In order to solve the above problems, Ref. [27] summed the hash values and classified them, and Ref. [22]

used K-means clustering technology in machine learning algorithm for average classification, but these algorithms had poor security. Compared with perceptual hash algorithm, quantum hash algorithm has better security. Ref. [8] introduced controlled alternate quantum walks (CAQW), which proved that the quantum hash function based on CAQW had better security. Ref. [20] proposed a quantum hash function based on the broken line quantum walk on a cycle. The algorithm improved the diffusion and confusion of quantum hash, but the space of quantum hash was complex.

As a special hash function, BioHashing has higher security than perceptual hash algorithm. Compared with quantum hash algorithm, BioHashing reduces the spatial complexity and improves the efficiency. Therefore, BioHashing algorithm has great development potential. Ref. [21] introduced the disturbance term into each biometric data, and Ref. [5] presented the improved 3FA scheme. The above algorithm only provided a high degree of privacy protection, but they did not achieve the balance between security and other performance. In order to solve the above problems, Ref. [3] proposed the transformation based biomedical template protection scheme as an improvement of the BioHashing algorithm. The scheme could maintain good efficiency and effectively respond to attacks at the same time. Ref. [17] proposed a new face biometric framework based on optical transformation. On the premise of meeting the cancellability criterion, the scheme had safety strength characteristics such as anti-collision, sensitivity and irreversibility.

In view of this, this paper adopts the Euclidean distance of MSCC algorithm, which is conducive to that the system can better balance robustness and discrimination, and improves the recall rate and precision rate of the system. At the same time, the biometric vectors of the classification and the corresponding dimension vectors of the orthogonal set matrix are one-to-one corresponding inner producted to construct the biosafety templates, which effectively the biometric template and improves the diversity of the template. By adopting the multi-key DES algorithm based on CBC mode to protect the plaintext, it has better anti-attack ability.

The main contributions of our approach can be summarized as follow:

1) MSCC feature extraction can well represent the feature information of speech, and has better discrimination and robustness;

2) The multi-key DES algorithm based on CBC mode technology is used to encrypt the speech clips, which greatly improves the security of the speech clips.

3) The algorithm is proposed to classify speech biometrics by K-medoids, and biometrics combined with the improved Hadamard measurement matrix to construct the diversity biosafety template, which greatly improves the security and diversity of biometric template.

2 Related Theory Introduction

2.1 K-medoids Algorithm

K-medoids has strong stability in the process of clustering, because it uses real samples in data clustering as the center of clustering, which can reduce the impact of noise and discrete groups on clustering [15]. It randomly selects K representative objects, and assigns the rest of the objects to the cluster where the most similar representative objects are located. Its partition method is based on the principle of minimizing the sum of dissimilarity degree between all objects p and its corresponding representative object o_i, namely absolute error standard function, which is defined as

$$J = \sum_{j=1}^{K} \sum_{p \in c_i} |p - o_j| \tag{1}$$

where J is the sum of the absolute error values of all objects in the dataset c_i. p is the point of c_i which represents an object of the cluster. o_j is the center of the cluster. The algorithm iterates repeatedly until each representative object becomes the actual center of its cluster.

2.2 Multi-key DES Encryption Algorithm Based on CBC Mode

Many years of practical application has proved that DES encryption algorithm is very excellent, but there are still some deficiencies more or less. Many experts and scholars have made unremitting research on it, including the exploration of key in DES, the improvement of S-box and so on.The traditional DES encryption algorithm based on CBC mode first divides the whole plaintext into grouped plaintext, then performs XOR operation on the grouped plaintext with the initial vector or the grouped ciphertext of the previous clip, and performs DES encryption with the grouped key [9]. However, because the sampling value of speech adopts double precision storage, therefore, the XOR operation of the traditional CBC mode can not meet the requirement of speech encryption.

The improved algorithm of DES algorithm based on CBC mode changes the XOR operation of CBC mode to inner product operation. The improved DES encryption algorithm based on CBC mode is as follows:

1) According to the Sine mapping system, $(\ell + 1) \times \vartheta$ length of a vector is generated and divided into $\ell + 1$ groups, the first group of data is used as the initial vector γ_0, and the remaining ℓ groups of data are used as the grouped key $\{\gamma_n | n = 1, 2, \cdots, \ell\}$ where the length of γ_n is ϑ bit.

2) The whole plaintext P is grouped to obtain the ℓ groups of grouped plaintext data $\{P_n | n = 1, 2, \cdots, \ell\}$ where the length of P_n is ϑ bit. The first group of grouped plaintext P_1 and the initialization vector γ_0 are inner producted, and then perform DES encryption with the grouped key

γ_1 to get the first group of grouped ciphertext C_1. The remaining $\ell-1$ group of grouped plaintext P_n and the previous groups of grouped ciphertext C_{n-1} perform inner product operation, and then perform DES encryption with the grouped key γ_n to obtain the grouped ciphertext C_n, namely

$$C_n = \begin{cases} des(P_1 \bullet \gamma_0, \gamma_n) & n = 1 \\ des(P_n \bullet C_{n-1}, \gamma_n) & n = 2, 3, \cdots, \ell \end{cases} \quad (2)$$

3) The whole ciphertext data $C = \{C_1, C_2, \cdots, C_\ell\}$ is obtained by splicing the grouped ciphertext data. The chart of the multi-key DES encryption is shown in Figure 1.

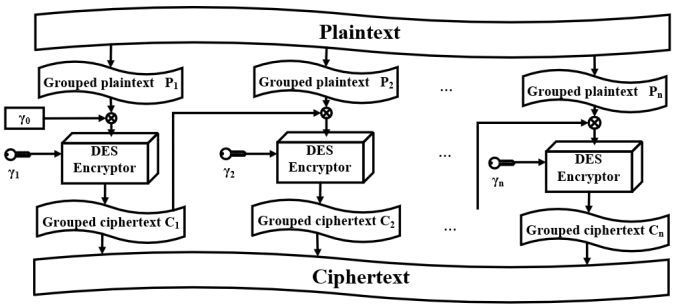


Figure 1: The flow chart of the improved DES encryption

2.3 MSCC

MSCC performs S-transform on the input speech information, and maps the spectrum information to the Mel spectrum reflecting human hearing to obtain the cepstrum coefficient. The algorithm overcomes the loss of MFCC biometrics to the spectrum biometrics, and improves the performance of discrimination and robustness of the retrieval system. Its S-transform has the advantages of short fast Fourier transform (SFFT) and wavelet transform (WT) [19].

$$S(\tau, f) = \int_{-\infty}^{+\infty} s(t) \frac{|f|}{\sqrt{2\pi}} e^{-\frac{(\tau-t)^2 f^2}{2}} e^{-i2\pi f t} dt \quad (3)$$

where $s(t)$ is the time domain signal, f is the frequency and τ is the translation factor.

$x(t)$ is the continuous time domain signal, where $t = m\Delta_T$, Δ_T is the sampling interval, the sampling sequence of $x(t)$ is defined as $x[m] = x(m\Delta_T), m = 0, 1, \cdots, N-1$. The discrete S-transform is defined as:

$$S[h,k] = \begin{cases} \frac{1}{N} \sum_{m=0}^{N-1} X[m] & k = 0 \\ \sum_{m=0}^{N-1} X[m+k] e^{(h(\frac{2\pi m j}{N}) - \frac{2\pi^2 m^2}{n^2})} & k \neq 0 \end{cases} \quad (4)$$

where $h, m = 0, 1, \cdots, N-1$, and $S[h,k]$ is the spectrum signal after S-transform.

The spectrum signal $S[h,k]$ is modeled and squared to obtain the energy spectrum signal, and the signal passes through M Mel band pass filters, and the Log energy is calculated for each time in each filter bank, namely

$$E'(h,m) = \ln(\sum_{k=0}^{N-1} |S[h,k]|^2 H_m(k)) \quad 0 \leq m \leq M \quad (5)$$

where $E'(h,m)$ is the output of the m filter, and $H_m(k)$ is the frequency response of the triangular filter.

$E'(h,m)$ is transformed by discrete cosine transform (DCT) to obtain L MSCC coefficients, namely

$$C(h,n) = \sum_{m=1}^{M} E'(h,m) \cos\left(\frac{\pi n(m-0.5)}{M}\right) \quad 1 \leq n \leq L \quad (6)$$

where L represents the order of cepstrum coefficient of MSCC, and the flow of the MSCC is shown in the Figure 2.

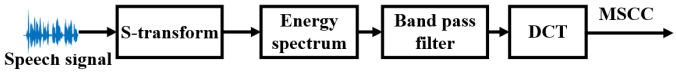


Figure 2: The chart of the improved DES encryption

2.4 Improved Hadamard Measurement Matrix

The deterministic measurement matrix (MM) [18] has the advantages of low computational complexity and easy implementation in hardware or software. The Hadamard MM is a type of deterministic MM. Compared with other deterministic MMs, the reconstruction effect of Hadamard MMs is the accurate reconstruction of the MM requires fewer measurements. Therefore, we efficiently generate a deterministic MM by combining the Modified Logistic chaotic map with a Hadamard MM. The construction method has the following steps:

1) N order of Hadamard matrix θ is generated.

2) The initial value ψ_1 and parameter ω are set, and the Modified Logistic chaotic sequence $\psi = \{\psi_1, \psi_1, \cdots, \psi_N\}$ is generated according to Formula (7), and the sequence ψ is sorted to obtain the index sequence I.

$$\psi_{n+1} = \omega\psi_n(1-\psi_n) \bmod 1, \psi_n \in (0,1), \; and \; \omega > 1 \quad (7)$$

3) The Hadamard matrix θ selects M rows of data according to the index sequence I to construct the Hadamard matrix $\Gamma = [\theta(I(1),:); \theta(I(2),:); \cdots; \theta(I(M),:)]$, where $I(i) \in [1, N]$, $1 \leq i \leq N$, $\theta(i,:)$ represents the i-th row element of the matrix θ.

3 System Model

The flow of BioHashing speech security retrieval algorithm based on MSCC and improved Hadamard measurement matrix proposed in this paper is shown in Figure 3.

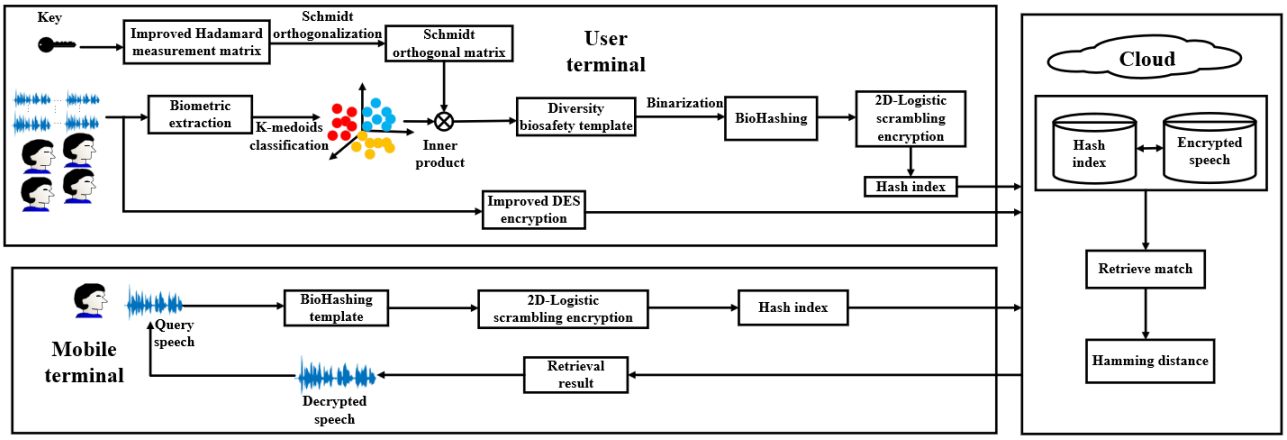


Figure 3: The flow chart of proposed algorithm

3.1 User Terminal

Step 1: Biometric extraction.

1) **Pre-processing.** The original speech signal is firstly pre-emphasized to improve the high-frequency part, then performs framing and windowing and the window function selects the Hamming window, the pre-processed signal is $X = \{X_m(i)|i = 1, 2, \cdots, N; m = 1, 2, \cdots, S\}$ with the frame length of S, the frame shift of T and the total number of frames of N.

2) **Feature extraction.** The MSCC coefficient $G = \{G_k(i)\,|i = 1, 2, \cdots, N; k = 1, 2, \cdots, L\}$ is obtained after MSCC. At the same time, the average value $\bar{G} = \left\{\overline{G(i)}\,|i = 1, 2, \cdots, N\right\}$ is calculated for the MSCC coefficient of each frame, and the Euclidean distance between G and $\bar{G}$ is calculated as the MSCC biometric vector $\Phi = \{\Phi(i)\,|i = 1, 2, \cdots, N\}$ according to Formula (8).

$$\Phi(i) = \sqrt{\sum_{k=1}^{L}\left(G_k(i) - \overline{G(i)}\right)} \tag{8}$$

L represents the order of cepstrum coefficient of MSCC.

Step 2: Construction of diversified BioHashing template.

1) **K-medoids classification.** The biometric data of M speech clips are randomly selected as the clustering center from the biometric library which constructed from the biometrics of 1200 speech clips library. According to the K-medoids algorithm, the biometric library which constructed from the biometrics of 1200 speech clips library are divided into M classes, namely $\Phi^{\lambda} = \{\Phi(i, \lambda)\,|i = 1, 2, \cdots, N; \lambda = 1, 2, \cdots, M\}$, where Φ^{λ} represents the biometric belongs to the λ-th class biometric library.

2) **Construction of orthogonal set matrix.** The constant of the Modified Logistic map is used as the key, namely $key = w$, and the measurement matrix $\Gamma^{\lambda} = \{\Gamma(i, \lambda)\,|i = 1, 2, \cdots, N; \lambda = 1, 2, \cdots, M\}$ is constructed according to the improved Hadamard measurement matrix system. Finally, the measurement matrix is Schmidt orthogonalized to obtain the orthogonal set matrix $\Upsilon^{\lambda} = \{\Upsilon(i, \lambda)\,|i = 1, 2, \cdots, N; \lambda = 1, 2, \cdots, M\}$.

3) **Construction of diversified biosafety template.** The λ-th dimensional vector of the orthogonal set matrix Υ^{λ} and the corresponding λ-th class biometric vectors D^{λ} are one-to-one correspondence inner product to generate the diversified biosafety templates $H^{\lambda} = \{H(i, \lambda)\,|i = 1, 2, \cdots, N; \lambda = 1, 2, \cdots, M\}$,which is defined as:

$$H^{\lambda} = \Phi^{\lambda} \bullet \Upsilon^{\lambda} \quad \lambda = 1, 2, \cdots, M \tag{9}$$

where H^{λ} indicates that the biometric vector belongs to the λ-th biometric template.

4) **Construction of BioHashing.** The biosafety template H^{λ} is binarized to generate the one-dimensional BioHashing sequence $h = \{h(i)\,|i = 1, 2, \cdots N\}$ according to Formula (10), where the BioHashing sequence $h(1)$ is set to 0.

$$h(i) = \begin{cases} 1 & (H^{\lambda}(i+1) > H^{\lambda}(i)) \\ 0 & otherwise \end{cases} \tag{10}$$

Step 3: Encryption algorithm.

1) The pseudo-random sequence F_1 with the same length as the hash sequence $h(i)$ is generated by

2D-Logistic chaotic system. F_1' is obtained by arranging F_1 in ascending order, namely a one-to-one mapping relationship is formed between F_1' and $h(i)$.

2) Through the above mapping relationship, $h(i)$ is assigned to F_1, and then F_1' is restored to the unordered state, namely a scrambling encryption sequence is formed for BioHashing sequence to form hash index $h'(i)$.

$$h'(i) \leftrightarrow F_1 \leftarrow F_1' \leftrightarrow h(i) \tag{11}$$

3) The whole plaintext speech x is grouped to obtain 1000 groups of 64 bits data length of grouped plaintext speech P_n, and the 1000 groups of 64 bit data bit length of grouped key γ_n and grouped plaintext speech are encrypted to obtain ciphertext C by Formula (1).

Step 4. Ciphertext C and its one-to-one corresponding hash index are sent to the cloud for storage.

3.2 Cloud

Step 1. The hash index $h_1'(i)$ of the query speech clip $x_1(n)$ is constructed according to the same method of the user terminal, and upload it to the cloud.

Step 2. In the cloud retrieval process, the hash index of the query speech clip is accurately matched with the corresponding λ-th class hash index of the cloud according to Formula (12).

$$BER(h_1', h_2') = \frac{1}{N}\sum_{i=1}^{N} |h'_1(i) \oplus h'_2(i)| \tag{12}$$

where $BER(:,:)$ is the Hamming distance, $h_1'(i)$ and $h_2'(i)$ respectively represent the hash index generated by the query speech clip of the user terminal and a certain class of hash index of the cloud.

This paper uses BER hypothesis test to describe hash matching.

Δ_0 : If the contents of the two speech clips are the same, then $BER < \tau$.

Δ_1 : If the contents of the two speech clips are different, then $BER \geq \tau$.

When the BER of two sequences is less than the set threshold τ, the matching is successful; otherwise, the matching is not successful.

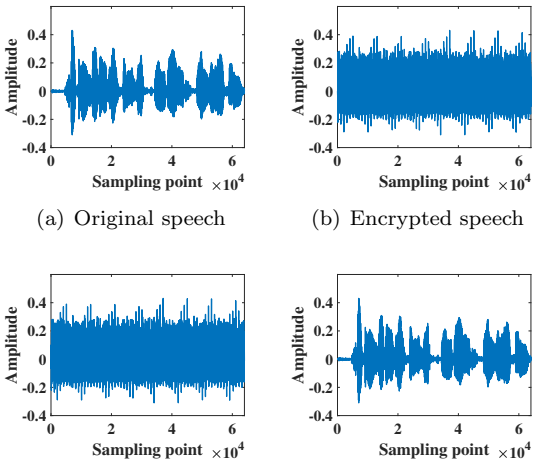
Step 3. If the matching of the hash index in the class fails, the result is directly fed back to the user. If the hash index in the class matches successfully, the ciphertext speech C corresponding to the matched index hash $h_1'(i)$ is decrypted and return it to the user, where the decryption process is the inverse process of encryption.

4 Experimental Results and Analysis

The experimental speech data comes from the Texas Instruments and Massachusetts Institute of Technology (TIMIT) speech library and the Text to Speech (TTS) speech library. And there are 1200 speech clips whose sampling frequency is 16kHz, sampling precision 16bit, mono, WAV format with a duration of 4s. The experimental hardware platform: Inter(R) Core (TM)i5-5200U CPU, 2.2GHz, memory of 4GB, software environment is Matlab R2018b under Windows 7. In this experiment, the following parameters are applied to the proposed algorithm:$N = 640$,$S = 200$,$T = 100$,$M = 35$,$L = 24$,$\omega = 3.99$, $\psi_1 = 0.7554$.

4.1 Data Security Analysis

In order to ensure the leakage of plaintext data in the cloud, this experiment uses an improved DES encryption algorithm to encrypt speech data. Figure 4 shows the speech waveform before and after encryptionn.



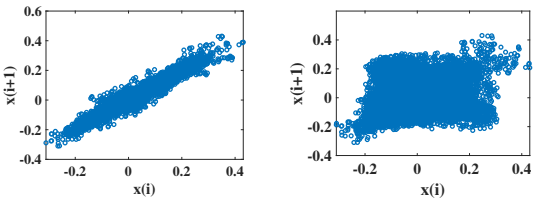
(a) Original speech (b) Encrypted speech

(c) Incorrectly decrypted (d) Correctly decrypted

Figure 4: Speech waveform before and after encryption

It can be seen from Figure 4 that the Multi-key DES encryption algorithm can hide the content of the original speech signal. Therefore, this algorithm guarantees the security of speech data.

In order to analyze the speech correlation, this paper randomly selects speech clips, and selects 32,000 sample points, where $x(i)$ is the abscissa (x axis), and $x(i+1)$ is the ordinate (y axis). The results before and after the encryption are drawn as shown in Figure 5.



(a) Original speech correlation (b) Encrypted speech correlation

Figure 5: Speech correlation before and after encryption

In order to further prove the superiority of the correlation of the encryption algorithm in this article, this article uses Spearman's correlation coefficient to measure the effect of encryption. The smaller the correlation coefficient, the better the encryption performance. Its definition is as follows:

$$r = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}} \quad (13)$$

where x_i is the original speech, y_i is the encrypted speech and r is the correlation coefficient.

It can be seen from Figure 5 that the speech correlation is destroyed after multi-key DES encryption. According to the formula calculation, the Spearman correlation coefficient value of the original speech is 0.978, and the Spearman correlation coefficient value of the multi-key DES encrypted speech is -0.0025. Therefore, it shows that the algorithm has better security.

4.2 BioHashing Performance Analysis

Discrimination is used to evaluate the reliability of the proposed algorithm to different speech or the same person speaking different content speech. The BER of the speech BioHashing value of different content basically obeys the normal distribution. The normal distribution of the BER data of the speech library in this experiment is shown in Figure 6.

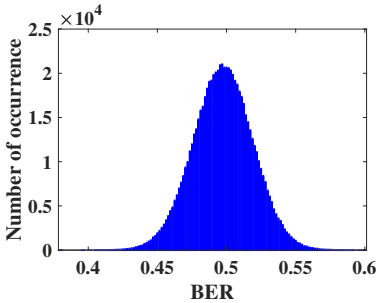


Figure 6: Normal distribution of BER

As shown in Figure 6, The BER statistical results we obtained by matching the constructed BioHashing abstracts ranged between 0.425 and 0.5703, which obeys a normal distribution with mean $\mu_1 = 0.4974$, standard deviation $\sigma_1 = 0.0216$, and a minimum value of 0.425. Therefore, the proposed algorithm has the better discrimination.

To further measure the discrimination of the algorithm under different thresholds, we also make use of the false acceptance rate (FAR) to reflect the discrimination of the proposed algorithm. The FAR is defined as:

$$FAR(\tau) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\tau} e^{\frac{-(x-\mu)^2}{2\sigma^2}} dx \quad (14)$$

where τ is the threshold, x is BER, and μ and σ are the mean value and standard deviation of the BER, respectively. The FAR values of different algorithms under different thresholds are shown in Table 1.

Table 1: FAR of the different algorithms

τ	Proposed	Ref. [24]	Ref. [27]	Ref. [22]
0.20	1.56E-43	2.99E-26	8.43E-20	1.13E-27
0.25	9.58E-31	1.67E-19	3.34E-14	4.06E-19
0.30	2.84E-20	3.89E-13	6.71E-09	6.49E-13
0.35	4.16E-12	3.88E-08	5.61E-06	4.72E-08

It can be seen from Table 1 that FAR value of the proposed algorithm is smaller than that in Refs. [22, 24, 27],so the algorithm has better the discrimination.This is because the Biometric extraction can better express the oeriginal speech signal. When matching the threshold $\tau = 0.35$, about 4.2 of every 10^{12} speech clips are misidentified, which shows that the algorithm has better anti-collision ability, that is, it has better discrimination and can meet the retrieval requirements of speech content.

For robustness analysis, in order to verify the robustness of the proposed algorithm, this paper performs content preserving operations (CPOs) on the speech clips of the speech library as shown in Table 2, resulting in 16800 speech clips of CPOs.

The robustness of the proposed algorithm compares the BioHashing sequence generated by 1200 original speech clips with the BioHashing sequence generated by speech clips of the content preserving operations. Figure 7 shows the BER mean of the 14 CPOs of the different algorithm.

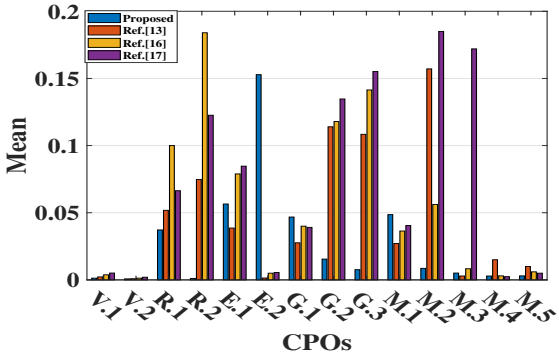


Figure 7: The BER mean of different algorithms

Table 2: Content preserving operations

Operating means	Operation method	Abbreviation
Volume Adjustment 1	Volume down 50%	V.1
Volume Adjustment 2	Volume up 50%	V.2
Low-pass Filtering 1	6 order FIR low-pass filtering, cut off frequency of 3.4 kHz	F.1
Low-pass Filtering 2	12 order FIR low-pass filtering, cut off frequency of 3.4 kHz	F.2
Low-pass Filtering 3	6 order Butterworth low-pass filtering, cut off frequency of 3.4 kHz	B.1
Low-pass Filtering 4	12 order Butterworth low-pass filtering, cut off frequency of 3.4 kHz	B.2
Resampling 1	Sampling frequency decreased to 8 kHz, then increased to 16 kHz	R.1
Resampling 2	Sampling frequency decreased to 32 kHz, then increased to 16 kHz	R.2
Echo Addition 1	Superimposed attenuation 25%, delay 300 ms	E.1
Echo Addition 2	Superimposed attenuation 60%, delay 300 ms	E.2
Narrowband Noise 1	SNR=30 dB narrowband Gaussian noise, center frequency distribution in 0∼ 4 kHz	G.1
Narrowband Noise 2	SNR=40 dB narrowband Gaussian noise, center frequency distribution in 0∼ 4 kHz	G.2
Narrowband Noise 3	SNR=50 dB narrowband Gaussian noise, center frequency distribution in 0∼ 4 kHz	G.3
MP3 Compression 1	Re-encoded as MP3, then decoding recovery, the rate is 32 k	M.1
MP3 Compression 2	Re-encoded as MP3, then decoding recovery, the rate is 64 k	M.2
MP3 Compression 3	Re-encoded as MP3, then decoding recovery, the rate is 96 k	M.3
MP3 Compression 4	Re-encoded as MP3, then decoding recovery, the rate is 128 k	M.4
MP3 Compression 5	Re-encoded as MP3, then decoding recovery, the rate is 192 k	M.5

It can be seen from Figure 7 that the BER mean values of the proposed algorithm after CPOs is small, and its mean value is distributed in (0.000758,0.1528). Therefore, the proposed algorithm has better robustness to the speech clips after 14 kinds of CPOs,And the BER mean of most CPOs of the algorithm in this paper is generally smaller than that of Refs. [22, 24, 27], except for CP0s of echo, narrow-band Gaussian noise with 30db and MP3 compression with 32k ratio, but it can meet the requirements of the retrieval system.

Similar to the discrimination analysis, the speech retrieval system uses the false rejection rate (FRR) to evaluate the robustness of the proposed algorithm. The FRR is as follows:

$$FRR(\tau) = 1 - \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\tau} e^{\frac{-(x-\mu)^2}{2\sigma^2}}dx \tag{15}$$

In order to evaluate the overall robustness and discrimination of the algorithm, the FRR-FAR curve is drawn according to the FAR obtained from the BER of different content preserving operations. The FRR-FAR curve of proposed algorithm is shown in Figure 8.

It can be seen from Figure 8 that FAR-FRR curve does not cross, which shows that the proposed algorithm has better discrimination and robustness, and can accurately recognize speech clips after CPOs and speech clips with different contents. When the threshold is selected at (0.21,0.38), both FAR and FRR values are small enough at the same time.

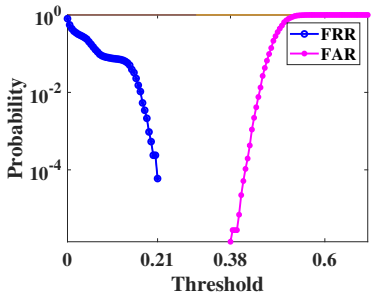


Figure 8: The FRR-FAR curve

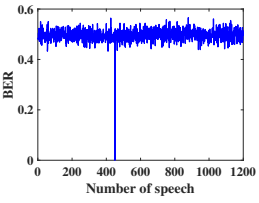
In addition, FAR and FRR can not entirely evaluate the speech hash sequence performance. We introduce entropy rete (ER) for further analysis of the performance of the hash sequence, which is the comprehensive evaluation of the algorithm discrimination and abstract.

$$ER = -[\hat{P}\log_2\hat{P} + (1-\hat{P})\log_2(1-\hat{P})] \tag{16}$$

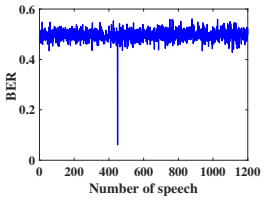
where $\hat{P} = \left(\sqrt{|\sigma^2 - \sigma_1^2|/(|\sigma^2 + \sigma_1^2|)} + 1\right)\Big/2$. This experiment randomly selected 1000 speech clips to test ER, and experimental results show that the value of ER is 0.9356, very close to 1, which means that the performance of the algorithm hash sequence is better

4.3 Retrieval Performance

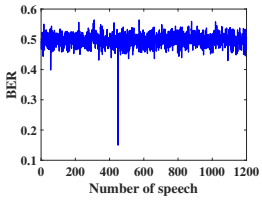
In order to evaluate the retrieval performance of the algorithm, this paper introduces precision ratio (R) and recall



(a) Original speech



(b) Echo Addition 1



(c) Echo Addition 2

Figure 9: Original speech and echo matching results

ratio (P), which are defined as follows:

$$R = \frac{f_T}{f_T + f_L} \times 100\% \quad (17)$$

$$P = \frac{f_T}{f_T + f_F} \times 100\% \quad (18)$$

where f_T is the number of speech clips related to the keyword in retrieval results, f_F is the number not related to the keyword and f_L is the number related to the keyword but not retrieved.

It can be seen from Figure 6 and Figure 7 that the minimum BER in the discriminative experiment is 0.425, and the maximum BER mean in the robustness experiment is 0.1528, the threshold value should belong to the interval (0.1528, 0.425). In this paper, in order to ensure that the algorithm still has better recall rate and precision rate after CPOs, this paper adopts 0.37 as the retrieval threshold.

In order to verify whether the system can retrieve a speech clip, this paper randomly selects an original speech, simultaneously selects echo1 and echo2 with poor robustness after CPOs as the query retrieval speech clips, and extracts the biometrics of the speech clips and generates the hash index sequences, and then match the hash index sequences with the hash index of the cloud, and calculate the BER value between the hash index sequences query speech and the hash index of the cloud. If the BER value is less than the threshold τ, that the query speech corresponding to this digest was a retrieval result, and the matching result is shown in Figure 9.

As can be seen from Figure 9, when the hash index generated by query speech clips are matched with all the hash index of the cloud, the accurate matching BioHashing sequence can be obtained only when the BER is very small, and the corresponding speech can be retrieved, and the other 1199 speech matching fails.

As can be seen from Figure 10, the larger the area enclosed by the P-R curve and the x-axis and y-axis, the

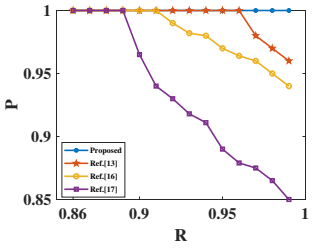


Figure 10: P-R curves of different algorithms

better the retrieval performance. In the figure, the P-R curve of this algorithm covers almost the whole image, while the area surrounded by x-axis and y-axis of P-R curve of other algorithms decreases continuously. Therefore, the proposed algorithm has better overall retrieval performance.

In order to further verify that the speech clips have better recall and precision after CPOs, the recall ratio and precision ratio of this paper are compared with Refs. [22, 24, 27], as shown in Table 3.

Table 3: FAR of the different algorithms

CPOs	Proposed	Ref. [24]	Ref. [27]	Ref. [22]
V.1	100/100	100/100	100/100	100/100
V.2	100/100	100/100	100/100	100/100
R.1	100/100	100/100	100/100	100/100
R.2	100/100	-	-	-
E.1	100/100	-	-	100/100
E.2	100/100	-	-	100/100
G.1	100/100	100/100	100/100	100/100
G.2	100/100	100/100	100/100	100/100
G.3	100/100	-	100/100	100/100
M.1	100/100	-	100/100	100/100
M.2	100/100	100/95	100/96	-
M.3	100/100	100/97	100/98	-
M.4	100/100	-	-	-
M.5	100/100	-	-	-

It can be seen from Table 3 that compared with other algorithms, the recall ratio and precision ratio of the speech clips after CPOs are 100%, indicating that the retrieval results of the proposed algorithm include all query related content and will not miss detection, while other algorithms will miss detection related content in different degrees.

To sum up, this algorithm not only has better recall ratio and precision ratio, but also P-R has a better compromise relationship. Therefore, this algorithm has better retrieval performance.

4.4 Unidirectional Verification and Analysis

In order to verify the unidirectionality with trapdoor of BioHashing that can reduce the leakage of the biomet-

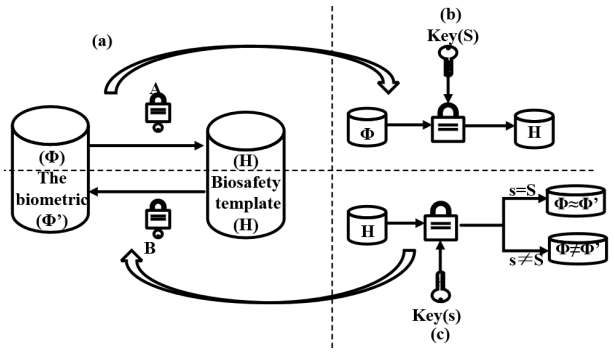
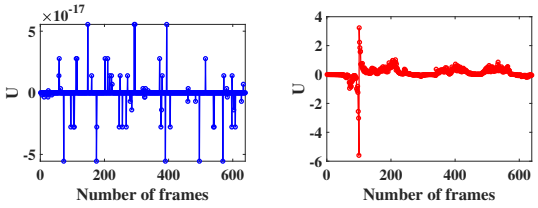


Figure 11: The analysis of unidirectional BioHashing



(a) The correct key (b) The wrong key

Figure 12: Difference of biometrics of different keys

ric, an algorithm of the unidirectionality with trapdoor based on the difference method is proposed. According to the Figure 11, part (a) shows that the original biometric vector Φ obtains the biosafety template H when passing through the direction A, and then the biometric vector Φ_1' is obtained from the biosafety template H when passing through the direction B. Part (b) shows that the original biometric vector Φ is transformed by a key S to obtain the biosafety template H, part (c) indicates that the biosafety template H extracts the biometric vector Φ_1' with the correct key s and the biometric vector Φ_2' with the wrong key s.

Firstly, the speech clip x is randomly selected in the speech library. Then, the original biometric vector Φ obtains the biosafety template H through the direction A, and then the biometric vector Φ' is obtained through the direction B. Finally, the difference method is used to prove the unidirectionality with trapdoor of BioHashing, which is defined as:

$$U(i) = \Phi(i) - \Phi'(i) \tag{19}$$

where Φ' is the biometric value obtained from direction B, Φ is the original biometric value and U represents the biometric difference state.

According to Figure 12(a)(b), only a few frames of the indexed data change and the distance between Φ_1' and Φ is distributed in $(-5.55 \times 10^{-17}, 5.55 \times 10^{-17})$ when the correct key is used to extract the biometric vector. According to Figure 12(a)(b), all frames of the indexed data change and the distance between Φ_2' and Φ is distributed in$(-5.596, 3.234)$, when the wrong key is used to extract the biometric vector. These indicate that the BioHashing has the unidirectionality with the trapdoor that prevents the leakage of the biometric.

4.5 Efficiency Analysis

In order to measure the complexity and the computational efficiency of our algorithm, this experiment selects 600 speech clips from the speech library for testing retrieval efficiency. In this experiment, the average efficiency of the retrieval algorithm includes biometric extraction and retrieval matching time, and the proposed algorithm is compared with the algorithm in Refs. [22, 24, 27]. The results are shown in Table 4.

Table 4: Retrieval efficiency of different algorithms

Algorithms	Average retrieval time
Proposed	0.1649s
Ref. [24]	0.0845s
Ref. [27]	0.0649s
Ref. [22]	0.2032s

As can be seen from Table 4, the efficiency of the algorithm in this paper is higher than that of Refs. [24,27], but lower than that of Ref. [22]. This is because the method of constructing the BioHashing template is complicated, which greatly improves the security of the proposed algorithm, but leads to a decrease in the efficiency of the proposed algorithm.

5 Conclusions

In this paper, a BioHashing speech security retrieval algorithm is proposed to construct the diversity biosafety template, which greatly improves the security and diversity of biometric template. The multi-key DES algorithm based on CBC mode technology is used to encrypt the speech clips, which greatly improves the security of the speech clips. At the same time, the biometric extraction algorithm is used to improve the robustness and discrimination of the system. But the disadvantage of this paper is that the encrypted speech retrieval object can only be fixed length speech clips, and the fuzzy retrieval of long speech clips is not considered.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.61862041), Science and Technology Program of Gansu Province of China (No.21JR7RA120).

References

[1] M. Bugaje and G. Chowdhury, "Is data retrieval different from text retrieval? an exploratory study," in *International Conference on Asian Digital Libraries*. Springer, 2017, pp. 97–103.

[2] M. Garg and G. Dhiman, "A novel content-based image retrieval approach for classification using glcm features and texture fused lbp variants," *Neural Computing and Applications*, vol. 33, pp. 1311–1328, 2021.

[3] L. Ghammam, M. Barbier, and C. Rosenberger, "Enhancing the security of transformation based biometric template protection schemes," in *2018 International Conference on Cyberworlds (CW)*. IEEE, 2018, pp. 316–323.

[4] S. He and H. Zhao, "A retrieval algorithm of encrypted speech based on syllable-level perceptual hashing," *Computer Science and Information Systems*, vol. 14, no. 3, pp. 703–718, 2017.

[5] Q. Jiang, Z. Chen, B. Li, J. Shen, L. Yang, and J. Ma, "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1061–1073, 2018.

[6] H. Kaur and P. Khanna, "Random distance method for generating unimodal and multimodal cancelable biometric features," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 709–719, 2018.

[7] M. Khurana and H. Singh, "Two level phase retrieval in fractional hartley domain for secure image encryption and authentication using digital signatures," *Multimedia Tools and Applications*, vol. 79, no. 19, pp. 13 967–13 986, 2020.

[8] D. Li, Y. G. Yang, J. L. Bi, J. B. Yuan, and J. Xu, "Controlled alternate quantum walks based quantum hash function," *Scientific reports*, vol. 8, no. 1, pp. 1–7, 2018.

[9] R. Munir, "Security analysis of selective image encryption algorithm based on chaos and cbc-like mode," in *2012 7th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*. IEEE, 2012, pp. 142–146.

[10] N. M. Patil and M. U. Nemade, "Content-based audio classification and retrieval using segmentation, feature extraction and neural network approach," in *Advances in computer communication and computational sciences*. Springer, 2019, pp. 263–281.

[11] J. Qin, X. Liu, and H. Lin, "Audio retrieval based on manifold ranking and relevance feedback," *Tsinghua Science and Technology*, vol. 20, no. 6, pp. 613–619, 2015.

[12] T. Rahim, S. Khan, M. A. Usman, and S. Y. Shin, "Impact of denoising on watermarking: a perspective for information retrieval," in *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2019, pp. 685–689.

[13] M. Shafieinejad and N. N. Esfahani, "A scalable post-quantum hash-based group signature," *Designs, Codes and Cryptography*, vol. 89, no. 5, pp. 1061–1090, 2021.

[14] T. Stanko, B. Chen, and B. Skoric, "Fingerprint template protection using minutia-pair spectral representations," *arXiv preprint arXiv:1804.01744*, 2018.

[15] A. V. Ushakov and I. Vasilyev, "Near-optimal large-scale k-medoids clustering," *Information Sciences*, vol. 545, pp. 344–362, 2021.

[16] J. Vavrek, P. Viszlay, M. Lojka, J. Juhár, and M. Pleva, "Weighted fast sequential dtw for multilingual audio query-by-example retrieval," *Journal of Intelligent Information Systems*, vol. 51, no. 2, pp. 439–455, 2018.

[17] G. Verma, M. Liao, D. Lu, W. He, and X. Peng, "A novel optical two-factor face authentication scheme," *Optics and Lasers in Engineering*, vol. 123, pp. 28–36, 2019.

[18] X. Wang, G. Cui, L. Wang, X. Jia, and W. Nie, "Construction of measurement matrix in compressed sensing based on balanced gold sequence," *Chinese Journal of Scientific Instrument*, vol. 35, no. 01, pp. 97–102, 2014.

[19] L. Xunbo, Z. Chunli, and L. Xin, "Speech endpoint detection based on improvement feature and s-transform," in *Intelligent Computing and Internet of Things*. Springer, 2018, pp. 225–235.

[20] Y. G. Yang, J. R. Dong, Y. L. Yang, Y. H. Zhou, and W. M. Shi, "Usefulness of decoherence in quantum-walk-based hash function," *International Journal of Theoretical Physics*, vol. 60, no. 3, pp. 1025–1037, 2021.

[21] C. Zhang, L. Zhu, and C. Xu, "Ptbi: An efficient privacy-preserving biometric identification based on perturbed term in the cloud," *Information Sciences*, vol. 409, pp. 56–67, 2017.

[22] Q. Y. Zhang, Z. X. Ge, Y. J. Hu, J. Bai, and Y. B. Huang, "An encrypted speech retrieval algorithm based on chirp-z transform and perceptual hashing second feature extraction," *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 6337–6361, 2020.

[23] Q. Y. Zhang, Z. X. Ge, and S. B. Qiao, "An efficient retrieval method of encrypted speech based on frequency band variance." *J. Inf. Hiding Multim. Signal Process.*, vol. 9, no. 6, pp. 1452–1463, 2018.

[24] Q. Y. Zhang, G. L. Li, and Y. B. Huang, "An efficient retrieval approach for encrypted speech based on biological hashing and spectral subtraction," *Multimedia Tools and Applications*, vol. 79, no. 39, pp. 29 775–29 798, 2020.

[25] Q. Y. Zhang, Y. Z. Li, and Y. J. Hu, "A retrieval algorithm for encrypted speech based on convolutional neural network and deep hashing," *Multimedia Tools and Applications*, vol. 80, no. 1, pp. 1201–1221, 2021.

[26] Q. Y. Zhang, F. J. Xu, and J. Bai, "Audio fingerprint retrieval method based on feature dimension reduction and feature combination." *KSII Transactions on Internet & Information Systems*, vol. 15, no. 2, 2021.

[27] Q. Zhang, Z. Ge, L. Zhou, and Y. Zhang, "An efficient retrieval algorithm of encrypted speech based on inverse fast fourier transform and measurement matrix," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 27, no. 3, pp. 1719–1736, 2019.

Biography

Yi-Bo Huang received Ph.D degree from Lanzhou University of Technology, and now working in Northwest Normal University, he main research interests include Multimedia information processing, information security, speech recognition.

Yuan Zhang received the B.S. degree from the Wuhan Institute of Technology, Hubei, China, in 2017. His research interests include audio signal processing and application, multimedia authentication.

Qiu-yu Zhang (Researcher/Ph.D supervisor), worked in Lanzhou university of technology. His research interests include information security and information hiding.

Guide for Authors
International Journal of Network Security

IJNS will be committed to the timely publication of very high-quality, peer-reviewed, original papers that advance the state-of-the art and applications of network security. Topics will include, but not be limited to, the following: Biometric Security, Communications and Networks Security, Cryptography, Database Security, Electronic Commerce Security, Multimedia Security, System Security, etc.

1. Submission Procedure

Authors are strongly encouraged to submit their papers electronically by using online manuscript submission at http://ijns.jalaxy.com.tw/.

2. General

Articles must be written in good English. Submission of an article implies that the work described has not been published previously, that it is not under consideration for publication elsewhere. It will not be published elsewhere in the same form, in English or in any other language, without the written consent of the Publisher.

2.1 Length Limitation:

All papers should be concisely written and be no longer than 30 double-spaced pages (12-point font, approximately 26 lines/page) including figures.

2.2 Title page

The title page should contain the article title, author(s) names and affiliations, address, an abstract not exceeding 100 words, and a list of three to five keywords.

2.3 Corresponding author

Clearly indicate who is willing to handle correspondence at all stages of refereeing and publication. Ensure that telephone and fax numbers (with country and area code) are provided in addition to the e-mail address and the complete postal address.

2.4 References

References should be listed alphabetically, in the same way as follows:
For a paper in a journal: M. S. Hwang, C. C. Chang, and K. F. Hwang, ``An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445--446, 2002.

For a book: Dorothy E. R. Denning, *Cryptography and Data Security*. Massachusetts: Addison-Wesley, 1982.

For a paper in a proceeding: M. S. Hwang, C. C. Lee, and Y. L. Tang, ``Two simple batch verifying multiple digital signatures," in *The Third International Conference on Information and Communication Security (ICICS2001)*, pp. 13--16, Xian, China, 2001.

In text, references should be indicated by [number].

Subscription Information

Individual subscriptions to IJNS are available at the annual rate of US$ 200.00 or NT 7,000 (Taiwan). The rate is US$1000.00 or NT 30,000 (Taiwan) for institutional subscriptions. Price includes surface postage, packing and handling charges worldwide. Please make your payment payable to "Jalaxy Technique Co., LTD." For detailed information, please refer to http://ijns.jalaxy.com.tw or Email to ijns.publishing@gmail.com.