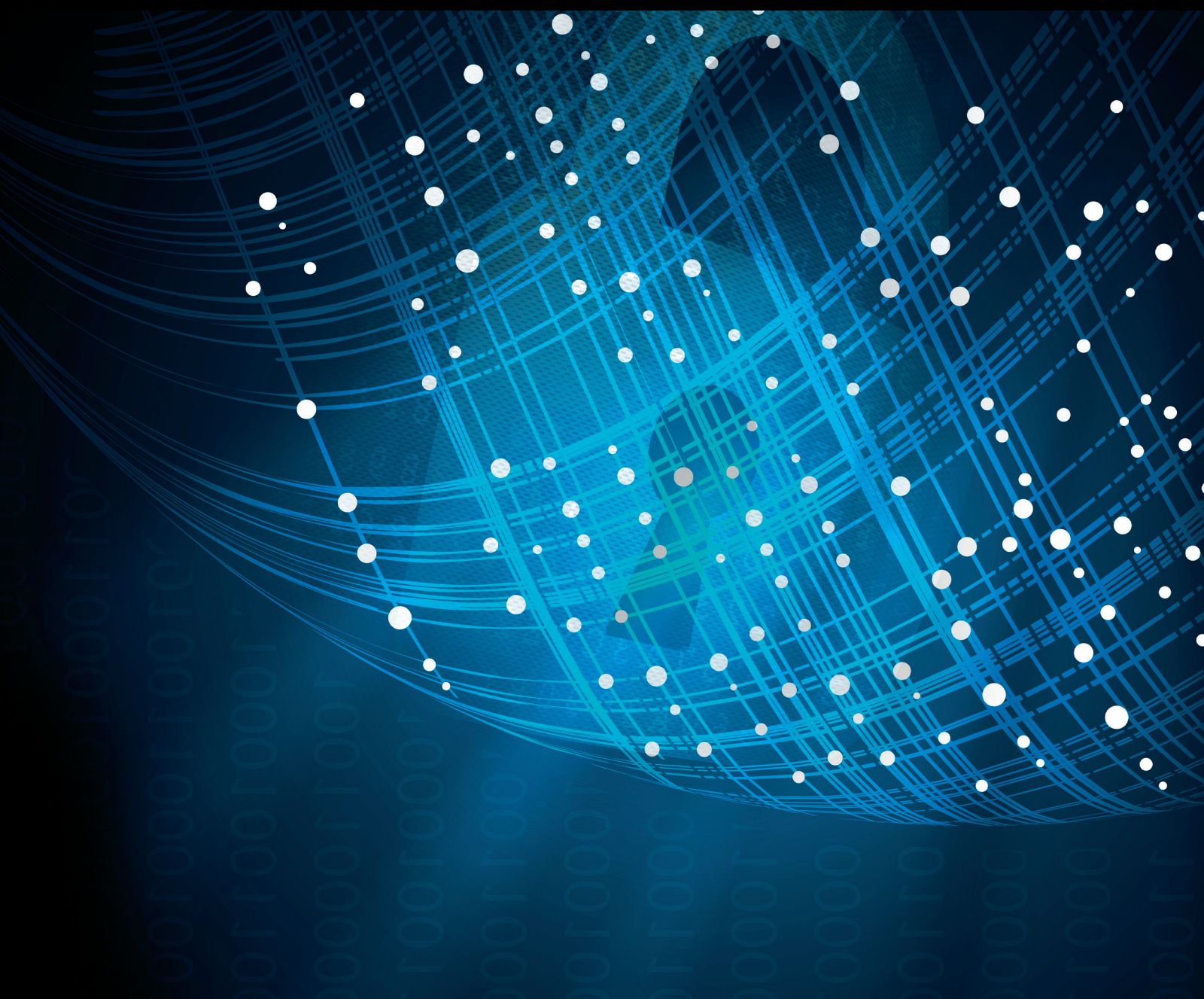
Security and Communication Networks

Machine Learning for Wireless Multimedia Data Security

Lead Guest Editor: Zhaoqing Pan
Guest Editors: Ching-Nung Yang, Victor S. Sheng, Naixue Xiong, and Weizhi Meng



Machine Learning for Wireless Multimedia Data Security

Security and Communication Networks

Machine Learning for Wireless Multimedia Data Security

Lead Guest Editor: Zhaoqing Pan
Guest Editors: Ching-Nung Yang, Victor S. Sheng, Naixue Xiong, and Weizhi Meng

Editorial Board

Contents

Hindawi
Security and Communication Networks
Volume 2019, Article ID 7682306, 2 pages
https://doi.org/10.1155/2019/7682306

Editorial

Machine Learning for Wireless Multimedia Data Security

Zhaoqing Pan,[1] **Ching-Nung Yang**,[2] **Victor S. Sheng**,[3] **Naixue Xiong**,[4] **and Weizhi Meng**[5]

[1] *School of Computer and Software, Jiangsu Engineering Center of Network Monitoring and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing, China*
[2] *Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan*
[3] *Department of Computer Science, University of Central Arkansas, Conway, AR, USA*
[4] *College of Intelligence and Computing, Tianjin University, Tianjin, China*
[5] *Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark*

Correspondence should be addressed to Zhaoqing Pan; zhaoqingpan@nuist.edu.cn

Received 17 March 2019; Accepted 17 March 2019; Published 1 April 2019

With the rapid development of multimedia technologies, the collection and modification of wireless multimedia data have become greatly convenient and easy. Meanwhile, the wireless multimedia data also made sensitive information available to potential attackers. The credibility of digital wireless multimedia data has thus decreased if the wireless multimedia data cannot be well protected. In addition, the copyright and privacy of wireless multimedia data also are easy to be infringed. Particularly, the data storage and computation have to be delegated to the powerful but always untrusted cloud, which has led to a series of challenging security and privacy threats.

Nowadays, artificial intelligence (AI) technology has been widely used in academia and industry. Machine learning can be regarded as one of the most important AI technologies, and it has been successfully used in image processing, pattern recognition, computer vision, natural language processing, and so on. Currently, the traditional steganography and security of encrypted wireless multimedia data face a lot of challenges. Thus, new types of steganography and encryption of wireless multimedia data, including audio, image, and video, are urgently needed to explore. Moreover, in new environments like cloud computing, the distribution and processing of wireless multimedia data also face more new challenges. For example, how to securely process wireless multimedia data in cloud computing to preserve the privacy of wireless multimedia data and how to reliably solve multiparty computing by outsourcing are still open questions.

This special issue aims to address the wireless multimedia data security problems by using machine learning methods. It includes seventeen papers, and the details of each paper are introduced one by one as follows.

To address the security problem of the multimedia data of the Internet of Thing (IOT), E. Min et al. proposed an intrusion detection system, which is based on the statistical and payload features. To extract the useful information from the payloads, the word embedding and text-convolutional neural network are used. Then, the random forest algorithm is applied to the final classification. Experimental results show that the proposed algorithms achieve better performance than the state-of-the-art algorithms.

To address the security problem of online transaction, Z. Zhang et al. proposed an online transaction fraud detection method based on the convolutional neural network (CNN). Since the low-dimensional and nonderivative online transaction data are used as the network input, the proposed method obtains an outstanding performance compared to the existing CNN-based fraud detection methods.

To address the limitations of the traditional encrypting algorithms, G. Xin et al. proposed an adaptive audio steganography algorithm. The proposed algorithm is based on the interval and variable low bit coding. Experimental

results show that the proposed algorithm achieves a better performance in embedding rate and invisibility than the other state-of-the-art audio steganography algorithms.

H. Dai et al. proposed privacy-preserving sorting algorithms for clouds, which are based on the basis of the logistic map. The security analysis and experimental results demonstrate that the proposed algorithms can well protect data privacy and provide efficient sorting on encrypted data.

Y. Zheng et al. proposed an improved image deep learning hash algorithm to learn the compact binary codes for image search. The proposed algorithm includes three parts, the feature extraction, deep secondary search, and image classification, respectively. Experimental results show that the proposed algorithm can efficiently identify the illegal images.

To address the security problems of lightweight block ciphers, P. Zhang and W. Zhang proposed a mixed-integer linear programming method to verify the security of Skinny-64/192. Experimental results show that the proposed method can significantly reduce the number of variables and improve the running speed of the computer.

For intrusion detection, Y. Xue et al. proposed an evolutionary computation based feature selection algorithm, in which the self-adaptive differential evolution is adopted. Experimental results show that the proposed algorithm is more promising than the state-of-the-art algorithms.

Based on the ensemble learning, Z. Zhang et al. proposed a wireless multimedia device identification system. The proposed system includes three parts, signal detection, RFF extraction, and classification model, respectively. Experimental results show that the identification rate of the proposed method can reach over 95%.

To deal with the data encryption problem, C. Ge et al. proposed a new source hiding identity-based proxy reencryption method (SHIB-PRE). The proposed SHIB-PRE method supports a proxy to convert a user's encrypted data to a new user's ciphertext when the proxy has the proxy reencryption key.

To deal with the security problem of the wireless network, L. Fang et al. proposed a fuzzy-conditional proxy broadcast reencryption method, in which the proxy uses a broadcast reencryption key to reencrypt the encrypted wireless multimedia data. The comparison results show that the proposed method can well address the security problems.

To protect the security of Android applications, S. Niu et al. proposed an improved permission management method, which is based on the machine learning algorithm. The proposed method uses a dynamic permission management database, and only the permission in the database can be used in this application. Experimental results show that the proposed method efficiently increases the flexibility of permission management.

To address the accuracy problem of single biometric method, Z. Wu et al. proposed a multimodal fusion algorithm for fingerprint and voiceprint, which is based on a dynamic Bayesian. Experimental results show that the proposed algorithm can efficiently improve the recognition rate and stability.

To deal with the problem of plaintext attack, X. Hu and Y. Zhao adopted the backpropagation neural networks to perform cryptanalysis on the Advanced Encryption Standard (AES). Experimental results show that the proposed algorithm can efficiently restore the entire byte.

To improve the image quality, C. Li et al. proposed a single image deraining algorithm by using the generative adversarial networks. Experimental results show that the proposed algorithm achieves an excellent performance in terms of image quality and computing efficiency.

L. Tan et al. proposed a visual secret sharing method for quick response (QR) code, which uses the grayscale QR codes as cover images and the binary QR code as the secret image. Experimental results show that the proposed method is robust to images with various types of distortions.

To deal with security problem of multimedia data in cloud, X. Zhao et al. proposed a hybrid homomorphic plaintext slot-wise switching algorithm; it can efficiently reduce computing and storage complexities of bootstrapping key generation. Experimental results show that the proposed algorithm can significantly optimize the bootstrapping procedure.

To address the security problem of medical image, B. Ma et al. proposed a reversible data hiding algorithm by using code division multiplexing and machine learning techniques. Experimental results prove that the proposed algorithm achieves outstanding performance in terms of data embedding capacity.

Conflicts of Interest

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

Acknowledgments

We would like to thank all of the authors who have submitted their work to this special issue. We also gratefully thank anonymous reviewers for providing helpful suggestions to improve the quality of the submissions. The launch of this special issue was supported in part by the National Natural Science Foundation of China under Grant no. 61501246, in part by the Natural Science Foundation of Jiangsu Province of China under Grant no. BK20150930, in part by the Six Talent Peaks Project of Jiangsu Province under Grant no. XYDXXJS-041, in part by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, and in part by the Project through the Priority Academic Program Development of Jiangsu Higher Education Institutions.

Zhaoqing Pan
Ching-Nung Yang
Victor S. Sheng
Naixue Xiong
Weizhi Meng

Hindawi
Security and Communication Networks
Volume 2019, Article ID 4732632, 9 pages
https://doi.org/10.1155/2019/4732632

Research Article

Code Division Multiplexing and Machine Learning Based Reversible Data Hiding Scheme for Medical Image

Bin Ma[1], **Bing Li**[1], **Xiao-Yu Wang**[1], **Chun-Peng Wang**[1], **Jian Li**[1] and **Yun-Qing Shi**[2]

[1] *School of Computer Science and Technology, Qilu University of Technology, China*
[2] *New Jersey Institute of Technology, New Jersey, USA*

Correspondence should be addressed to Bin Ma; mab@qlu.edu.cn, Chun-Peng Wang; mpeng1122@163.com, and Jian Li; ljian20@gmail.com

Received 19 June 2018; Accepted 5 November 2018; Published 17 January 2019

Guest Editor: Weizhi Meng

In this paper, a new reversible data hiding (RDH) scheme based on Code Division Multiplexing (CDM) and machine learning algorithms for medical image is proposed. The original medical image is firstly converted into frequency domain with integer-to-integer wavelet transform (IWT) algorithm, and then the secret data are embedded into the medium frequency subbands of medical image robustly with CDM and machine learning algorithms. According to the orthogonality of different spreading sequences employed in CDM algorithm, the secret data are embedded repeatedly, most of the elements of spreading sequences are mutually canceled, and the proposed method obtained high data embedding capacity at low image distortion. Simultaneously, the to-be-embedded secret data are represented by different spreading sequences, and only the receiver who has the spreading sequences the same as the sender can extract the secret data and original image completely, by which the security of the RDH is improved effectively. Experimental results show the feasibility of the proposed scheme for data embedding in medical image comparing with other state-of-the-art methods.

1. Introduction

Recently, most hospitals have developed medical information management system to provide better, safer, and efficient service for patients. In the system, the patient's medical images are usually saved in DICOM format for future diagnostic, research, and long-term transmission, in which the patient's personal information and their sufferings are usually saved simultaneously [1]. However, with the development of digital multimedia processing technology, illegally collecting and modification of medical image have become more and more easier. Most sensitive information of these multimedia is liable to be exposed to attackers in open wireless network environment; therefore, the multimedia such as the medical image need to be well protected to assure their safety. Consequently, the protection of multimedia data becomes more and more significant in the process of cloud storage and cloud computing [2]. RDH is a special kind of method that can embed secret data into the multimedia, and the cover can be losslessly reconstructed after the embedded data having been extracted completely. Presently, the traditional RDH methods for multimedia protection are facing a lot of challenges, especially for medical images. As the medical image is so sensitive, even a small change of pixels may cause huge impacts on disease diagnosis; namely, any changes of the medical image may do great harm to doctor's judgment. At the same time, with the development of artificial intelligence (AI), machine learning has been widely employed to protect the security of wireless multimedia data. Thus, the machine learning based RDH algorithm is highly desired to guarantee the security of medical images in such situation [3].

There are a lot of reversible data hiding methods that have been proposed in past few years; they can be roughly classified into three categories: the method based on lossless compression, the method based on histogram shifting, and the method based on difference expansion. The method of lossless compression based RDH was firstly presented by Fredrich et al. [4, 5]. They embedded the secret data into

the vacant room of least significant bit planes of the original image, which is achieved with compression. Celik et al. [6] proposed a high performance scheme which enhanced the lossless compression efficiency through prediction-based conditional entropy coder, and thus the capacity of lossless data embedding is improved.

Ni et al. [7] firstly presented the histogram shifting (HS) based efficient RDH method. The peak point and zero/minimum point of the histogram of original image are employed to determine the histogram bins to be shifted with one position. Then, the secret data are embedded into the empty spaces achieved through the shifting of histogram bins. From then on, a lot of RDH schemes have been proposed to improve the performance of HS-based RDH scheme. Xuan et al. [8] enhanced the performance of HS-based RDH method with IWT algorithm. The secret data are embedded into the space of medium frequency subbands to achieve high data embedding capacity and imperceptibility. Fallahpour and Sedaghi [9] separated the original image into some blocks and applied HS scheme on each one. In this scheme, the peaks and zeros (or minima) of the histogram are generated from each block and the amount of the highest histogram bins of the whole image is then increased, through which the data embedding capacity is improved at low image distortion. Xuan et al. [10] utilized the histogram pairs of image prediction-errors for data embedded, in which four thresholds were introduced to improve the data embedding performance, and thus they achieved excellent results especially at low-to-moderate embedding capacity than others. In addition, Li et al. [11] proposed a two-dimensional difference histogram modification based RDH scheme, by which the redundancy of the cover image is better exploited and high data embedding performance is achieved.

Difference expansion (DE) is another most widely used RDH method, which is firstly presented by Tian [12]. In their scheme, the difference of adjacent pixel pairs is expanded and the secret data are embedded to the expansion created spaces furtherly. As the largest data embedding capacity of Tian's method is no more than 0.5BPP (bit per pixel). Later on, various DE based RDH schemes have been proposed to improve its performance. Thodi and Rodriguez [13] presented the first prediction-error expansion based RDH scheme, in which the prediction-errors of the object pixels are utilized for data embedding. According to the close correlation inherent in the neighbourhoods of the object pixel, the distortion of the cover image is greatly reduced at high data embedding capacity. Sachnev et al. [14] improved the performance of HS-based RDH scheme through prediction-error expansion and sorting method. In the scheme, the location map is not necessarily needed even with large data embedding capacity and thus the distortion of the cover image is reduced. Wang et al. [15] presented an efficient integer-to-integer transform based RDH method and demonstrated that Tian's method can be reformulated as a special instance of integer-to-integer transform. Finally, they verified the superiority of the proposed scheme comparing with other traditional methods. Li et al. [16] suggested embedding secret information into scalable pixels according to local complexity of the cover image and adopting an adaptive prediction-error expansion method to achieve large data embedding capacity with low image distortion simultaneously. Lee et al. [17] presented an IWT algorithm based high capacity RDH scheme. The cover image is divided into nonoverlapping blocks and the secret data are embedded into the high frequency coefficients of each block. Thus, the proposed scheme obtained large data embedding capacity at a lower level of image distortion. Fallahpour et al. [18] divide the medical image into tiles and embedded data into each tile with HS-based RDH method. The experimental results demonstrate that the data embedding capacity can reach 30%-200% improvement and are still with low distortion. Furtherly, in [19], Coltuc and Chassery presented a low mathematical complexity RDH scheme based on reversible contrast mapping. As an integer-to-integer transform utilized on pixel pairs, this method does not require any additional lossless data compression. Ma et al. [20] combined the code division multiple access (CDMA) and IWT algorithm to improve the robustness of the RDH. According to the orthogonality of the spreading sequence employed in the scheme, the proposed method achieves high reversible data hiding performance especially at large data embedding capacity.

As the medical image plays a vital role on disease analysis [21–24], they are very important and sensitive in the process of disease diagnosis and treatment process. Hence, the cover image needs to be completely recovered after the embedded data having been extracted, to guarantee the reliability and security of the medical image. Alqershi et al. [25] presented a hybrid RDH algorithm for medical images; the medical image firstly is separated into two categories: the region of interest (ROI) and the region of noninterest (RONI). The secret data are embedded into ROI areas with DE based RDH scheme, while the additional information is embedded into RONI through another data hiding algorithm. Agrawal et al. [26] introduced the IWT and HS algorithms based RDH scheme for data embedding in medical images and achieved better data hiding performance comparing with other methods. However, the data embedding capacity of medical image is still needed to be improved to conceal more patient's privacy; at the same time, the robust and the security of the RDH in medical have not yet been studied in the past to enhance the credibility of the medical image in open wireless network environment.

In this paper, a new scheme based on CDM and machine learning is proposed for medical images RDH. In the process of data embedding, the original medical image is firstly converted into frequency domain with IWT algorithm; the secret data are then embedded into the medium frequency subbands of image with CDM and machine learning algorithms, so that the robustness and security of reversible data embedding are obtained. At the same time, the small-sample neural network algorithm is employed to optimize the embedding coefficients determination and thus high embedding performance is obtained. According to the orthogonality of the spreading sequences employed in the CDM algorithm, the elements of different spreading sequences would be mutually canceled when the data are repeatedly embedded, which enable the marked image to keep low image distortion even at high data embedding. In addition, as the secret data

are represented by different spreading sequences for data embedding, only the receiver who has the same spreading sequences and the same data embedding gain factor as the sender can reconstruct the secret data and original image completely, which improves the security of the medical image. Consequently, the proposed scheme achieves both high data embedding capacity and security.

The structure of the rest paper is designed as follows: In Section 2, the algorithm of CMD based RDH for medical image is described. In Section 3, the RDH scheme combined with CDM and machine learning algorithms is provided. In Section 4, the experimental results are shown and analyzed. Section 5 draws the conclusions of the paper.

2. CDM Based Reversible Data Hiding for Medical Image

DICOM is a standard format in medical image exchange sponsored by National Electrical Manufacturers Association (NEMA). As it integrates the manufacturers of imaging facility and imaging information systems together in a file, DICOM format file is now widely used in medical image management including image processing, storing, printing, and transmitting.

CDM is a kind of wireless communication algorithm developed on spectrum spreading communication techniques. In a CDM based communication system, the to-be-transmitted signals are denoted by different orthogonal spreading sequences and transmitted noninterfering to each other in the same channel to save the frequency resources. Similarly, an RDH system can be viewed as a communication system, in which the secret data are the signals to be transmitted and the cover image is the communication channel.

2.1. CDM Based Reversible Data Hiding. In the CDM based RDH system, suppose $W = \{w_i,\ 1 \le i \le n\}$ is the original secret data to be embedded. The element w_i of secret data can be represented by the opposite bits $B = \{b_i,\ 1 \le i \le n\}$ with the equation

$$b_i = \begin{cases} 1, & if\ w_i = 1; \\ -1, & if\ w_i = 0; \end{cases} \tag{1}$$

Generate k mutually orthogonal spreading sequences $S_i = \{s_1, s_2, s_3, \cdots s_k\}$ from a standard *Hadamard* matrix firstly. According to the character of *Hadmard* matrix, the number of "1" and "-1" is equivalent in each spreading sequence; as the length of the orthogonal spreading sequence l is even, the candidate spreading sequences are zero-mean and orthogonal to each other.

Let I represents the original image with the size of $N \times N$; choose pixels of the image to form the original vector $i_j = \{x_j,\ 1 \le j \le N \times N/l\}$, where l is the length of the vector i_j (the same as the length of S_i). Then, the secret data can be embedded as

$$i'_j = i_j + \alpha \left[b_1 S_1 + b_2 S_2 + \cdots + b_k S_k\right] \tag{2}$$

In (2), k bits of secret data are embedded into vector i_j. Here, k is the number of orthogonal spreading sequences which have been added repeatedly onto the original vector; α is the gain factor of data embedding, which is always a positive integer. The bigger the value of α is, the higher the embedding strength of the proposed method is, and the stronger the data embedding would be. Finally, the marked image is obtained with the vectors i'_j.

It is also clear to see that the shorter the length of spreading sequence is, the more the secret data can be embedded. On the contrary, the long the spreading sequence is, the more the security would be obtained. Moreover, as the spreading sequences are orthogonal to each other, most elements of spreading sequences would be mutually canceled when the data are embedded repeatedly into the object vectors, and thus less image distortion would be achieved even with large data embedding capacity.

2.2. CDM Based Secret Data Extraction. Suppose I' is the marked image. Constructing i'_j with the same method as in the process of data embedding and then calculating the cross correlation of vector i'_j and spreading sequence S_i, the secret bit can be extracted as follows:

$$\begin{aligned} \left\langle i'_j, S_i \right\rangle &= i'_j \cdot S_i^T \\ &= i_j \cdot S_i^T \\ &\quad + \alpha \left[b_1 S_1 \cdot S_1^T + b_1 S_1 \cdot S_1^T + \cdots b_i S_i \cdot S_i^T\right] \end{aligned} \tag{3}$$

As the spreading sequences are orthogonal to each other, (3) can be reduced to

$$\left\langle i'_j, S_i \right\rangle = i_j \cdot S_i^T + \alpha b_i S_i S_i^T \tag{4}$$

where α is always a positive integer and the result of the $S_i \bullet S_i^T$ is always positive. Hence, the sign of expression $\alpha b_i S_i \bullet S_i^T$ is determined by b_i. In the case of $i_j \bullet S_i^T < |\alpha b_i S_i \bullet S_i^T|$, the embedded data can be extracted as

$$b_i = \text{sign}\left(i'_j \cdot S_i\right) \quad if\ i_j \cdot S_i^T < \left|\alpha b_i S_i \cdot S_i^T\right| \tag{5}$$

Equation (5) shows that the condition of $|\alpha S_i \bullet S_i^T|$ is greater than $i_j \bullet S_i^T$, the value of $\text{sign}(i_j \bullet S_i^T)$ exactly equals the embedded bit b_i, and thus the secret data can be extracted completely. As S_i is a zero-mean spreading sequence, the expression of $i_j \bullet S_i^T$ equals to gather the difference of adjacent pairs of pixels. Therefore, if the elements of i_j are similar, the magnitude of $i_j \bullet S_i^T$ is quite small, which enables more secret bits to be embedded into the original image with less image distortion. Moreover, as the secret bits are denoted by different spreading sequences, only the receiver who knows the same spreading sequences with the sender can extract the secret data and recover the original image completely; the security of the proposed scheme is greatly improved compared with those traditional RDH methods.

In addition, as the proposed RDH scheme is achieved with different orthogonal spreading sequences, the secret data can be embedded into the cover image repeatedly. The

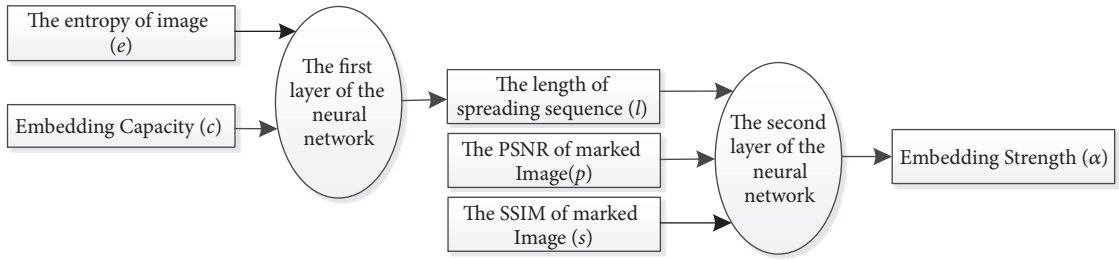


Figure 1: The flow of 2-layer small-sample BP neural network.

data embedding capacity is then improved multiply and that can be estimated with

$$C = \left(\frac{T * M * N}{l}\right) - \varepsilon \tag{6}$$

where C denotes the embedding capacity, T denotes the number of embedding levels, M and N are the rows and columns of the original image, l denotes the length of orthogonal spreading sequence, and ε represents the size of the additional message.

2.3. CDM Based Original Image Recovery. After the secret data having been extracted from the marked image, according to the equations introduced above, the original image can be recovered completely with the formula:

$$i_j = i'_j - \alpha \left[b_1 S_1 \cdot {S_1}^T + b_1 S_1 \cdot {S_1}^T + \cdots b_k S_k \cdot {S_k}^T\right] \tag{7}$$

In sum, as the secret data are embedded with different spreading sequences and gain factors, the receiver who has the embedding spreading sequences and gain factor the same with the sender can extract the corresponding secret data and recover the original cover image exactly. At the same time, most elements of different spreading sequences would be mutually canceled in the process of repeatedly data embedding. Consequently, the proposed CDM based RDH scheme achieves both high data embedding capacity and security.

2.4. Principle of Small-Sample Neural Network. As the small-sample neural network can solve the problem of large samples dependent of neural network, it is an effective machine learning algorithm widely employed for parameters optimization in complex system. The basic principle of small-sample neural network is to find the optimal parameters for a complexing system from small samples; therefore, the parameters can truly reflect the solution of the whole problem, and thus the blind selection of parameters is avoided.

In this paper, a second layer small-sample neural network is employed to optimize the parameters of the proposed RDH system. In the first layer of the small-sample neural network, the embedding capacity c and the entropy e are set as the network input of the neural network, the best length of spreading sequence l as network output. In the second layer of the neural network, the length of spreading sequence l, the Peak Signal to Noise Ratio (PSNR), and the structural similarity index (SSIM) are utilized as the input, and the gain factor α is utilized as the output of the neural network. The flow of a small-sample neural network algorithm is shown in Figure 1.

The mathematical model of the proposed small-sample neural network is

$$R(l, \alpha) = N(\mathrm{e}, c, p, s) \tag{8}$$

where e is the entropy of original cover image, c is data embedding capacity, p is PSNR, and s is SSIM of the recovered image. l and α are the length of the spreading sequence and the gain factor of data embedding separately. Part of samples for small-sample neural network is shown in Table 1.

In our experiment, the sample data are normalized firstly and the maximum training samples are 1000, the training target error and the learning rate are set to 0.01 and 0.1 separately, and the result value of training error is 0.001. The small-sample neural network is trained with pregenerated samples to optimize the coefficients for different data embedding conditions. The purpose of the training is to establish the nonlinear mapping relationship between the employed parameters of reversible data embedding and the quality of marked image successfully. The results show that, for most medical image (the data embedding capacity is no more than 5000 bits), the optimum length of the spreading sequence l and the gain factor α is 4 and 1, respectively. The training results indicate the feasibility and effectivity of the small-sample neural network for the proposed scheme.

3. Integer-to-Integer Wavelet Transform Based RDH

As the medical images generally has large flatten background areas, the IWT algorithm is then quite suitable for medical image transform, by which most low frequency parts of image can be filtered. Generally, when the data are embedded into the medium frequency subbands of image, high quality marked image and robust data embedding can be obtained even with large embedding capacity. In addition, as the high sensitive and important characters of medical image, it is necessary to recover the medical image completely after the embedded data have been extracted. However, in the condition of the image modified with conventional wavelet transform, the wavelet coefficients cannot be guaranteed to remain integer after image transform, and thus some embedded bits may be lost and the original image can not be

Table 1: Part samples for small sample neural network.

Test parameters					Test results	
e	C/bpp	$PSNR/dB$	SSIM	M	α	l
3.1554	0.01	67.02	0.9997	0.0069	1	2
4.1295	0.05	63.56	0.9854	0.8361	2	4
5.2154	0.1	50.95	0.9703	1.0374	3	6
9.6965	0.15	37.62	0.9615	8.2569	4	8
7.6924	0.2	40.77	0.9528	5.4459	5	2
...	...	...	...	...	...	...
3.5987	0.01	65.26	0.9905	0.1345	1	4
4.5654	0.05	57.85	0.9823	0.2439	2	6
6.5987	0.1	47.69	0.9685	6.5987	3	8
6.9852	0.15	43.65	0.9568	2.6392	4	2
9.1541	0.2	39.71	0.9425	25.0000	5	4
...	...	...	...	...	...	...
7.2658	0.01	53.16	0.9786	0.0129	1	6
5.0257	0.05	51.86	0.9728	5.2658	2	8
6.5993	0.1	47.52	0.9661	1.0374	3	2
6.8547	0.15	42.83	0.9543	12.2658	4	4
7.1025	0.2	41.68	0.9534	5.0934	5	6
...	...	...	...	...	...	...
7.3654	0.01	52.68	0.9718	0.5987	1	8
7.5681	0.05	53.74	0.9792	0.9679	2	2
6.7513	0.1	45.38	0.9593	8.0046	3	4
6.5924	0.15	47.53	0.9621	2.4521	4	6
10.1207	0.2	35.59	0.9316	15.2587	5	8
...	...	...	...	...	...	...

completely recovered when any floating point value is cut off. Therefore, the integer-integer wavelet transform algorithm is highly expected to guarantee the reversibility of RDH for medical images.

The algorithm of IWT on an image can be achieved as (9) on row transform and then as (10) on column transform.

Row transformation:

$$H: d_{v,m,n} = s_{v-1,2m+1,n} - \left\lfloor \frac{1}{2}\left(s_{v-1,2m,n} + s_{v-1,2m+2,n}\right) + \frac{1}{2} \right\rfloor$$
$$L: s_{v,m,n} = s_{v-1,2m,n} - \left\lfloor \frac{1}{4}\left(d_{v,m-1,n} + d_{v,m+1,n}\right) + \frac{1}{2} \right\rfloor \tag{9}$$

Column transformation:

$$HH: dd_{v,m,n} = d_{v,m,2n+1} - \left\lfloor \frac{1}{2}\left(d_{v,m,2n} + d_{v,m,2n+2}\right) + \frac{1}{2} \right\rfloor$$
$$HL: ds_{v,m,n} = d_{v,m,2n+1} - \left\lfloor \frac{1}{2}\left(s_{v,m,2n} + s_{v,m,2n+2}\right) + \frac{1}{2} \right\rfloor$$
$$LH: sd_{v,m,n} = d_{v,m,2n} - \left\lfloor \frac{1}{4}\left(dd_{v,m,n-1} + dd_{v,m,n+1}\right) + \frac{1}{2} \right\rfloor$$
$$LL: ss_{v,m,n} = d_{v,m,2n} - \left\lfloor \frac{1}{4}\left(ds_{v,m,n-1} + ds_{v,m,n+1}\right) + \frac{1}{2} \right\rfloor \tag{10}$$

where the subscripts v, m and n represent the decomposition levels of the coefficients, the column index, and the row index, respectively.

In the experiment, the image is decomposed into four subbands in the first level: low frequency subband (LL), medium frequency subbands (HL; LH), and high frequency subband (HH). Figure 2 shows the subbands after the IWT of a medical image. As medical image has large flatten background areas, the LL subband includes much image information, and high image visual distortion would be introduced if this subband is employed for data embedding. Therefore, the data is preferred to be embedded in HL, LH subbands to improve the robust of data embedding and reduce the image distortion after data embedding. In the process of data embedding, the length of spreading sequence and the data embedding strength is determined with small-sample neural network for reversible data embedding and extracting.

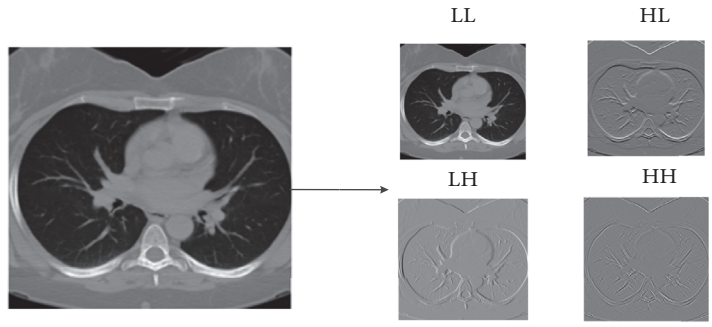


Figure 2: The subbands *LL*, *LH*, *HL*, and *HH* of the RDH method for a medical image.

When the length of spreading sequence is set to 4, the maximum data embedding capacity would be 0.125BPP when the *LH* and *HL* subbands are involved for one-time data embedding. At the same time, according to the orthogonality of spreading sequence, the secret data can be embedded repeatedly on same subband without interfering to each other. Therefore, the data embedding capacity of medical image is highly improved, which ensure that the data embedding capacity is sufficient for patient's personal privacy hiding with the proposed method. On the other hand, as most elements in subbands *LH* and *HL* are modified for data hiding in the proposed scheme, the histogram equalization is liable to be achieved and the contrast of the cover image is improved; hence, the visual quality of marked image would be enhanced with the CDM based RDH scheme.

In the process of reversible data embedding, at the sender side, the integer-to-integer wavelet transform algorithm is first utilized on the original cover image, then the CDM and machine learning based RDH is employed to embed secret bits into the medium frequency subbands of a medical image; finally, the inverse IWT algorithm is adopted to get the marked image. The process of data hiding is shown in Figure 3(a), the outline of our proposed RDH scheme in wavelet domain is as follows:

(1) Segment the background and foreground of the medical image with Sobel operator, remove the segmented background of the original image, and obtain the region of interest (ROI) in medical images for the further processing.

(2) Apply IWT to ROI region of the image, and then obtain low frequency subband *LL*, medium frequency subbands *HL*, *LH* and high frequency subband *HH*.

(3) Utilize CDM and machine learning based RDH for data embedding in the medium frequency subbands *HL* and *LH*.

(4) Construct the marked image with inverse IWT algorithm on the medical image.

At the receiver side, the whole process of data extraction is shown in Figure 3(b); the steps of data extracting in wavelet domain can be descripted in short as follows:

(1) Convert the marked image into frequency domain with IWT algorithm.

(2) According to the features of CDM based RDH, extract the embedded data correctly from *LH* and *HL* subbands of the marked image; the process is inverse to the data embedding.

(3) Convert the marked image into its original state without distortion.

4. Experimental Results and Discussion

In the experiments, 6 DICOM format gray scale medical images with the same size of 512×512 obtained from the database of The Cancer Imaging Archive (TCIA) have been employed for the evaluation of the proposed RDH scheme. The secret data is a random binary sequence only containing elements "0" and "1"; meanwhile, a location map is utilized to mark the location where the secret data is embedded, whose size usually can be compressed very small. The medical images chosen from database TCIA are shown in Figure 4.

4.1. Results Evaluation with PSNR Indicator. For reversible data hiding techniques, generally, PSNR is utilized to demonstrate the distortion between the marked image and the original one. Higher PSNR values generally indicate that the marked image obtains excellent visual quality and thus with lower distortion.

PSNR between the marked and original image can be obtained with the following equations:

$$PSNR = 10 * \log_{10}\left(\frac{255 \times 255}{MSE}\right) \tag{11}$$

where I denotes the cover image with the size of $M \times N$ and I_w is the marked image. The expression of MSE is

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I(i,j) - I_w(i,j)\right]^2 \tag{12}$$

Figure 5 shows the BPP-PSNR curves of the medical images (a)–(f) after data embedding. The results shown in the Figure 5 demonstrated the superiority of the proposed

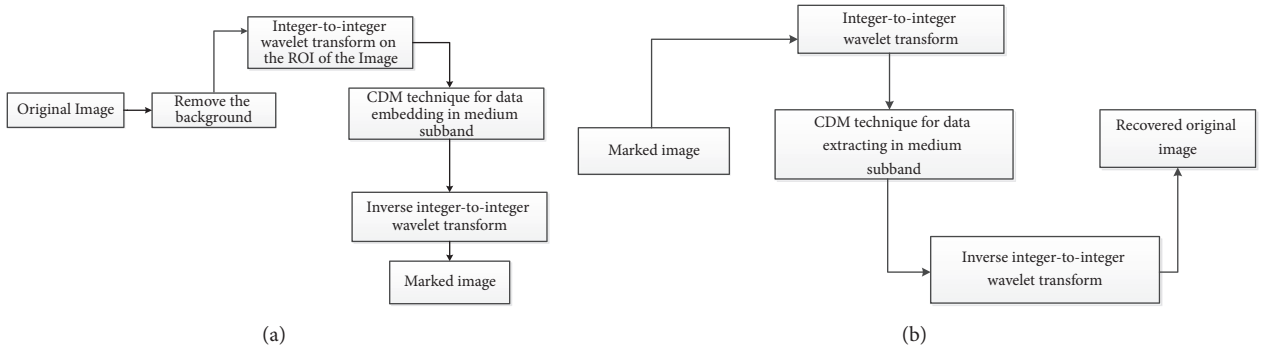


Figure 3: (a) The process of embedding. (b) The process of extracting.

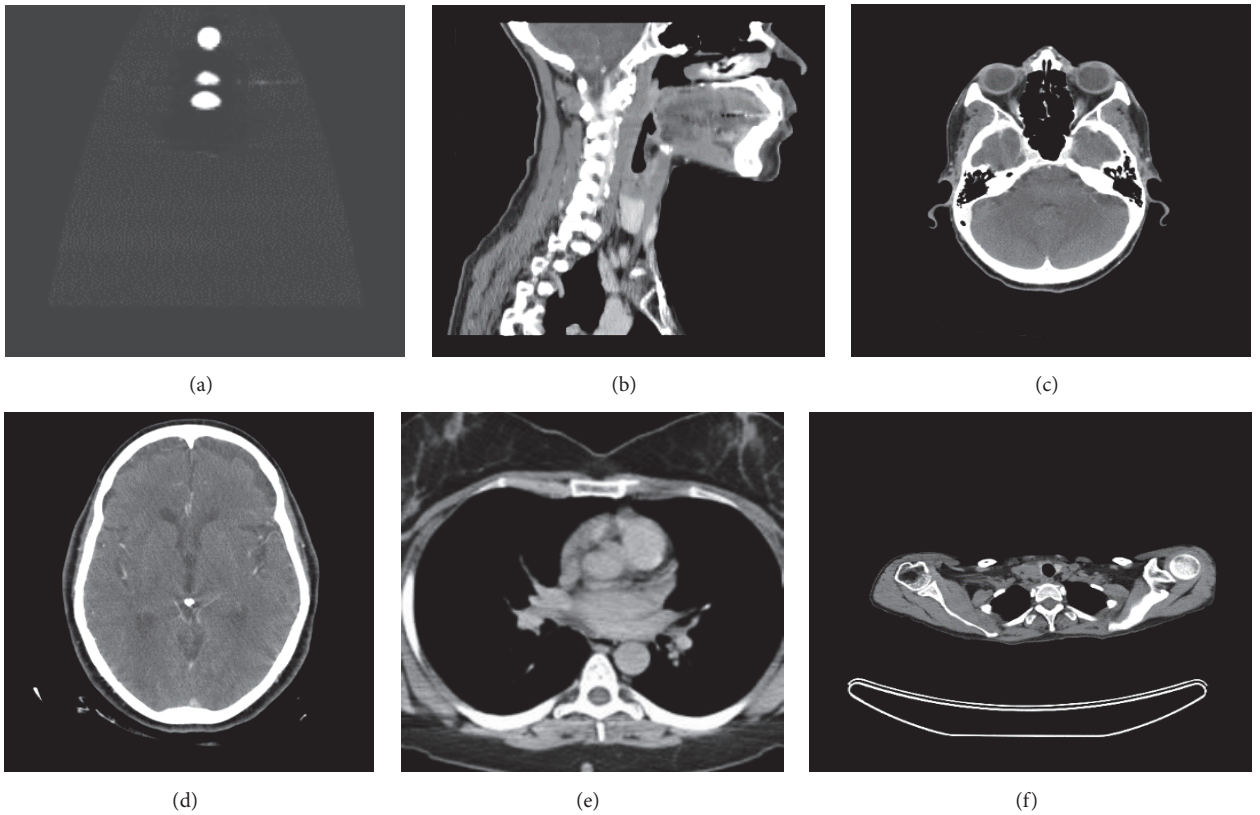


Figure 4: The medical images chosen from TCIA.

scheme. When the data embedding capacity is 0.125BPP, the PSNR value of all marked image is still above 52dB. The proposed scheme is sufficient for RDH in medical images. Meanwhile, the results also demonstrate that the image with large ROI areas achieves high PSNR than those with large RONI areas at the same image distortion. For instance, image (a) includes largest ROI areas in 6 images, and thus it achieves higher PSNR than others at the same data embedding capacity.

In sum, the proposed scheme in this paper could achieve excellent image visual quality even after high capacity RDH. Moreover, as the spreading sequences are employed to embed secret bits, the secret data and the original image can only be recovered completely by the receiver who has the same gain factor and the spreading sequences with the sender; thus, the security of the cover image is guaranteed and the patients' personal information is protected completely.

4.2. Results Evaluation with SSIM Indicator. SSIM is another widely utilized indicator to evaluate the performance of RDH scheme. Here, we further employed SSIM to denote the performance of the proposed scheme. The formula of SSIM is

$$SSIM(x, y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_1\right)} \quad (13)$$

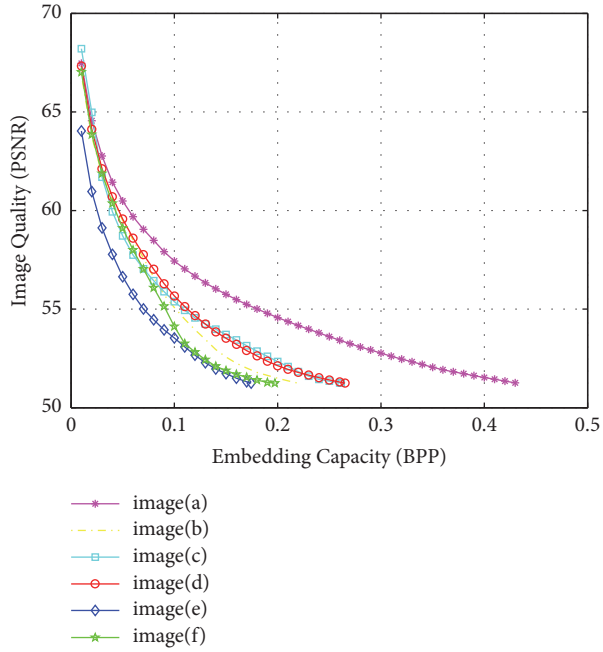


Figure 5: The BPP-PSNR curve of proposed scheme.

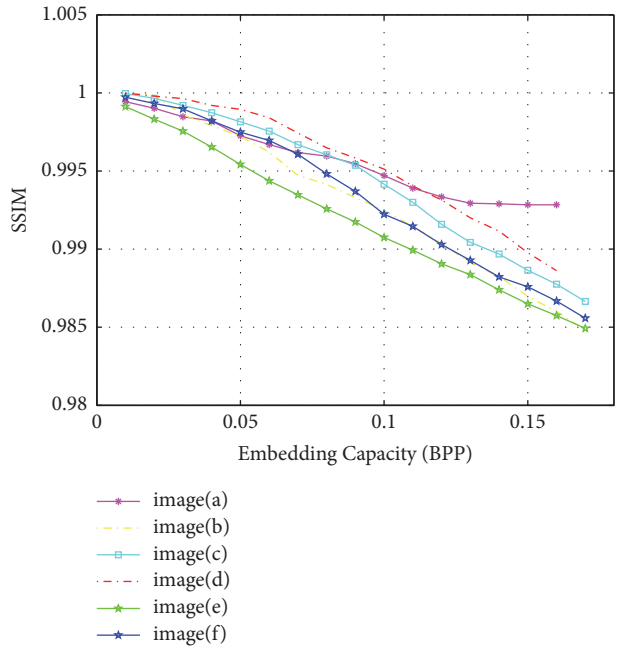


Figure 6: Comparison of SSIM versus payload on 6 medical images.

where μ_x is the average value of x and μ_y is the average values of y, respectively, σ_x^2 is the variance of x, σ_y^2 is the variance of y, and σ_{xy} is the covariance of x and y.

The experimental results on 6 medical images from TCIA image database are shown in Figure 6.

As shown in Figure 6, the results indicate that the marked image is very similar to the original one. The SSIM of the medical images drops slowly with the increases of data embedding capacity; at the same time, the SSIM of marked image with large ROI areas performs apparently superior to those with large RONI areas. When the embedding capacity is 0.1BPP, the SSIM of image (a) is 0.994, while the value is 0.991 for image (e) at the same embedding capacity. Moreover, as the image (e) has more RONI areas than other images (such as image (a)), the SSIM of image (e) drops faster than those with more ROI images. The experimental results show that the scheme proposed in this paper could achieve high data embedding capacity and security at low image distortion, which is sufficient for the protection of medical image and patient's privacy.

5. Conclusions

This paper presents a novel RDH scheme based on CDM and machine learning algorithms for medical images. In the proposed scheme, the IWT algorithm is applied to the medical image to converse the image into wavelet domain; then the secret data are embedded into the medium frequency subbands with the CDM and machine learning algorithms. According to the orthogonality of spreading sequences employed for data embedding, the secret data can be embedded into the same subband repeatedly, and most elements of different spreading sequences are mutually canceled. Therefore, the data embedding capacity is improved and the image distortion is restrained at the same time. Moreover, the secret data and the original image can only be recovered completely by the receiver who has the same spreading sequences and embedding factor the same as the sender, which improves the security of the proposed RDH system as well. In the scheme, a small-simple neural network is also employed to optimize the data embedding coefficients, by which the performance of the proposed scheme is improved effectively. The experimental result shows that the proposed scheme achieves high performance even at large data embedding capacity for medical images, which indicates the promising prospect of proposed scheme for protection of medical images and patient's privacy.

Data Availability

The format of DICOM images used to support the findings of this study has been deposited in the website http://www.cancerimagingarchive.net/.

Conflicts of Interest

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

Acknowledgments

The research reported in this paper was partially supported by National Natural Science Foundation of China (nos. 61802212, 61872203, and 61502241) and Project of Shandong Province Higher Educational Science and Technology Program (J18KA331).

References

[1] R.-C. Raúl, F.-U. Claudia, and G. D. J. Trinidad-Blas, "Data hiding scheme for medical images," in *Proceedings of the IEEE 17th International Conference on Electronics, Communications and Computers (CONIELECOMP '07)*, Cholula, Mexico, February 2007.

[2] Y. W. Yang, N. Rongrong, Z. Yao et al., "Watermark Embedding for Direct Binary Searched Halftone Images by Adopting Visual Cryptography," *Computer, Materials & Continua*, vol. 55, no. 2, pp. 255–265, 2018.

[3] B. Ma and Y. Q. Shi, "A reversible data hiding scheme based on code division multiplexing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1914–1927, 2016.

[4] M. Goljan, J. J. Fridrich, and R. Du, "Distortion-free data embedding forimages," in *Proceedings of the 4th Information Hiding Workshop*, pp. 27–41.

[5] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in *Proceedings of the Security and Watermarking of Multimedia Contents III*, vol. 4314 of *Proceedings of SPIE*, pp. 197–208, San Jose, Calif, USA, January 2001.

[6] M. Celik, G. Sharma, A. Tekalp, and E. Saber, "Reversible data hiding," in *Proceedings of the International Conference on Image Processing (ICIP '02)*, pp. II-157–II-160, Rochester, NY, USA, 2002.

[7] Z. Ni, Y. Q. Shi, N. Ansari et al., "Reversible data hiding," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[8] G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni, and W. Su, "Distortionless data hiding based on integer wavelet transform," *IEEE Electronics Letters*, vol. 38, no. 25, pp. 1646–1648, 2002.

[9] M. Fallahpour and M. H. Sedaaghi, "High capacity lossless data hiding based on histogram modification," *IEICE Electronics Express*, vol. 4, no. 7, pp. 205–210, 2007.

[10] G. Xuan and Y. Q. Shi, *Reversible data hiding*, IEEE Press, US, 2012, 8175324 B2[P].

[11] X. Li, W. Zhang, X. Gui, and B. Yang, "A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1091–1100, 2013.

[12] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[13] D. M. Thodi and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721–730, 2007.

[14] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989–999, 2009.

[15] X. Wang, X.-L. Li, B. Yang, and Z.-M. Guo, "Efficient generalized integer transform for reversible watermarking," *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 567–570, 2010.

[16] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524–3533, 2011.

[17] S. Lee, C. D. Yoo, and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 321–330, 2007.

[18] M. Fallahpour, D. Megias, and M. Ghanbari, "Reversible and high-capacity data hiding in medical images," *IET Image Processing*, vol. 5, no. 2, pp. 190–197, 2011.

[19] D. Coltuc and J.-M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 255–258, 2007.

[20] B. Ma and Y. Q. Shi, "A Reversible Image Watermarking Scheme Based on Modified Integer-to-Integer Discrete Wavelet Transform and CDMA Algorithm," in *Digital-Forensics and Watermarking*, vol. 9023 of *Lecture Notes in Computer Science*, pp. 420–432, Springer International Publishing, Cham, 2015.

[21] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[22] R. Gurusamy and V. Subramaniam, "A machine learning approach for MRI brain tumor classification," *Computers, Materials and Continua*, vol. 53, no. 2, pp. 91–109, 2017.

[23] C. Wang, X. Wang, Z. Xia, and C. Zhang, "Ternary radial harmonic Fourier moments based robust stereo image zero-watermarking algorithm," *Information Sciences*, vol. 470, pp. 109–120, 2019.

[24] Y. Zheng, B. Jeon, L. Sun, J. Zhang, and H. Zhang, "Student's t-hidden markov model for unsupervised learning using localized feature selection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[25] O. M. Al-Qershi and B. E. Khoo, "Authentication and data hiding using a hybrid ROI-based watermarking scheme for DICOM images," *Journal of Digital Imaging*, vol. 24, no. 1, pp. 114–125, 2011.

[26] S. Agrawal and M. Kumar, "Reversible data hiding for medical images using integer-to-integer wavelet transform," in *Proceedings of the IEEE Students' Conference on Electrical, Electronics and Computer Science, SCEECS '16*, pp. 1–5, India, March 2016.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 6362010, 9 pages
https://doi.org/10.1155/2018/6362010

Research Article

Analysis on Matrix GSW-FHE and Optimizing Bootstrapping

Xiufeng Zhao,[1] **Hefeng Mao**,[1] **Shuai Liu**,[1] **Weitao Song**,[1] **and Bo Zhang**[2,3]

[1]*Department of Information Research and Security, Zhengzhou Information Science Technology Institute, Zhengzhou 450001, China*
[2]*School of Information Technology, Deakin University, Victoria 3125, Australia*
[3]*School of Information Science and Engineering, University of Jinan, Jinan 250022, China*

Correspondence should be addressed to Xiufeng Zhao; zhao_xiu_feng@163.com

Received 8 August 2018; Revised 29 October 2018; Accepted 3 December 2018; Published 19 December 2018

Guest Editor: Zhaoqing Pan

With the rapid development of multimedia technologies, the multimedia data storage and outsource computation are delegated to the untrusted cloud, which has led to a series of challenging security and privacy threats. Fully homomorphic encryption can be used to protect the privacy of cloud data and solve the trust problem of third party. In this paper, we analyse circular security of matrix GSW-FHE scheme. We derive a sufficient condition of circular security for matrix GSW-FHE scheme. It allows us to choose a good secret key via "reject sample" technique and furthermore obtain circular secure matrix GSW-FHE scheme. We also give an extended version of matrix GSW-FHE by defining deterministic asymmetric encryption algorithm and propose hybrid homomorphic plaintext slot-wise switching method, which significantly reduces computation and storage complexity of bootstrapping key generation, thus optimizing the bootstrapping procedure.

1. Introduction

With the rapid development of multimedia technologies, for example, high-efficiency video coding (HEVC) is becoming popular due to its excellent coding performance [1]; the multimedia data storage and outsource computation are delegated to the untrusted cloud server, which has led to a series of challenging security and privacy threats. To tackle the security and privacy issues in cloud computing and storage, a lot of researches have been performed, such as fully homomorphic encryption [2, 3], attribute-based encryption, searchable encryption [4], and ciphertext retrieval scheme [5, 6]. The concept of homomorphic encryption is proposed by Rivest et al. [7], and Gentry [2, 3] proposed the first fully homomorphic encryption (FHE) scheme based on ideal lattice. FHE allows us to evaluate any function over ciphertext and obtain the function over corresponding plaintext by decryption. Fully homomorphic encryption can be used to protect the privacy of cloud data and solve the trust problem of untrusted third party. So the fully homomorphic encryption has a broad application prospect in the cloud computation and the big data field. There are many fully homomorphic encryption schemes based NP-hard problems, such as ideal lattice [2, 3], LWE [8, 9], RLWE [10], LWR [11], and so forth.

The difficulty of constructing fully homomorphic encryption scheme is reducing the noise in the ciphertext. The noise increases rapidly during ciphertext evaluations and eventually reaches a threshold beyond which we can no longer decrypt the resulting ciphertext correctly. Therefore, the somewhat homomorphic encryption scheme is constructed, which can homomorphically evaluates arithmetic circuits of limited depth. To get pure fully homomorphic encryption scheme, Gentry proposed bootstrapping technique. The bootstrapping technique is currently the only way to get pure fully homomorphic encryption from somewhat homomorphic encryption. Its main idea is refreshing ciphertext by homomorphic decryption and getting fresh ciphertext and realizing the purpose of reducing ciphertext noise. The critical process of bootstrapping technique is encrypting the pieces of secret key, and the corresponding ciphertexts are viewed as public evaluation key. Thus, the homomorphic encryption scheme must enjoy circular security.

Unfortunately, all known FHE schemes are supposed to be circular secure except [10, 12]. If fully homomorphic

encryption scheme satisfies circular security, it is not necessary to generate as many public evaluation keys as the depth of evaluation circuit. But being circular secure is not a naive security attribute, so it is necessary to analyse circular security for concrete fully homomorphic encryption scheme. Meanwhile, bootstrapping is used to refresh ciphertext, and the procedure is implemented frequently to get pure fully homomorphic encryption. Therefore, how to improve the bootstrapping efficiency is worth intensive studying.

Our Results. We analyse circular security of matrix GSW-FHE scheme [13]. From formal definition of circular security, we derive a sufficient condition of circular security for matrix GSW-FHE scheme. That is, the matrix GSW-FHE scheme satisfies circular security with some function, if the equations about secret key have solution over $\mathbb{Z}_q$. Therefore, we can choose a good secret key via "reject sample" technique and furthermore obtain circular secure matrix GSW-FHE scheme.

We also give an extended version of matrix GSW-FHE by defining deterministic asymmetric encryption algorithm. To simplify the homomorphic equality test procedure, we propose hybrid homomorphic plaintext slot-wise switching method using symmetric encryption and deterministic public encryption algorithms, which significantly reduces computational cost of bootstrapping key generation, thus optimizing the bootstrapping procedure of work [13].

We may implement a trade-off between computation and storage complexity of bootstrapping. We delete part of the bootstrapping keys and compute them online when running Rounding procedure. In view of that, their computation involves only matrix additions; this cuts down the size of the large public bootstrapping key by a third, paying matrix additions with negligible computation complex.

Related Works. Encryption scheme achieves circular security, if it remains secure and even the secret key is encrypted under corresponding public key. In other words, circular secure encryption scheme resists key-dependent message (KDM) attack.

In the last few years, circular secure encryption schemes have been studied extensively [14–17]. Boneh et al. constructed a circular secure public key encryption scheme based on the DDH assumption without random oracle [16]. Based on Regev's LWE-based encryption scheme [18], Applebaum et al. constructed efficient cryptosystems enjoying circular secure [17]. Brakerski and Vaikuntanathan [10] proposed circular secure homomorphic encryption scheme based on the ring-LWE assumption. The main idea in the work of [10, 17] is generating a valid ciphertext that decrypts to a message related to secret key. Because the entries of secret key are not in the message space, they introduced "noise flooding technique" and "rerandom technique" to "fit" the entries into the message space.

Brakerski and Vaikuntanathan presented a fully homomorphic encryption scheme based on the LWE assumption using relinearization technique [8]. The relinearization process allows doing one multiplication without increasing the size of the ciphertext and obtaining an encryption of the product under a new secret key. Posting a "chain" of L secret keys allows performing up to L levels of multiplications without blowing up to the ciphertext size. Yang et al. consider that if the relinearization satisfies circular security, the "chain" of L secret keys may be back down to only one secret key, and they proposed a circular secure relinearization by defining a new assumption [12].

EuroCrypt 2013, Gentry, Sahai, and Waters proposed a new fully homomorphic encryption scheme based on the *approximate eigenvector* method, which is called GSW-FHE [19]. In the GSW-FHE scheme, homomorphic addition and multiplication are just matrix addition and multiplication. But GSW scheme operates one bit every running encryption algorithm. PKC 2015, Hiromasa et al. constructed a variant of GSW scheme called matrix GSW-FHE, which encrypts matrices and supports homomorphic matrix addition and multiplication. And they optimized the bootstrapping procedure of Alperin-Sheriff and Peikert [20] using the matrix GSW-FHE scheme [13]. To achieve homomorphic matrix operation, the pubic key of matrix GSW-FHE scheme includes the ciphertexts that encrypt partial information of the secret key, so the matrix GSW-FHE scheme resorts to circular security assumption, but formal circular security proof was not given, and it remains an open problem.

There are other works to optimize the bootstrapping procedure. Ducas et al. [21] proposed FHEW scheme, which accelerates bootstrapping via embedding the cyclic group $\mathbb{Z}_q$ into the group of roots of unity: $i \longrightarrow X^i$, where i is a primitive q-th root of unity. Wang and Tang [22] proposed an integer bootstrapping scheme by introducing new methods to evaluate integer polynomials with GSW-FHE, and they extended the method to packing by encrypting the integers diagonally in a matrix, as the matrix GSW-FHE proposed by Hiromasa et al. [13]. Similarly, their scheme resorts to circular security assumption.

On the other hand, packing technique is used to evaluate efficiently a large number of ciphertexts, and it allows us to apply single-instruction-multiple-data (SIMD) homomorphic operations to all encrypted data [23, 24]. The bootstrapping procedure [13, 20] is optimized by embedding $\mathbb{Z}_q$ into symmetric group S_q, the multiplication group of q × q permutation matrix, and homomorphic permuting SIMD ciphertexts. The mathematic preliminary of SIMD technique is Chinese Remainder Theorem (CRT). The plaintext space can be split into many small spaces via the CRT. If the plaintext modulus q is a composite that factors into distinct powers $\mathrm{q} = r_1 \dots r_t$, then the ring R_q can be mapped via the CRT to direct product of ring R_{r_i}'s.

Organization. In Section 2, we describe some preliminaries on the formal definition of homomorphic encryption and circular security and the isomorphic from additive group $\mathbb{Z}_\mathrm{q}$ to a group of cyclic permutations. In Section 3, we review the matrix GSW-FHE scheme and define a new deterministic asymmetric encryption algorithm. We give the analysis on circular security of matrix GSW-FHE scheme in Section 4. In Section 5, we propose hybrid plaintext slot switching method and optimize the bootstrapping procedure. We give conclusions in Section 6.

2. Preliminaries

We denote the set of integers by $\mathbb{Z}$. Let G be some group and let P be some probability distribution, and then we use $a \xleftarrow{U} \mathrm{G}$ to denote that a is chosen from G uniformly at random and use $b \xleftarrow{R} \mathrm{P}$ to denote that b is chosen along P.

The vector is denoted by bold lowercase letter, for example, $\mathbf{x}$, and the i-th element of a vector $\mathbf{x}$ is denoted by x_i. The inner product between two vectors is denoted by $\langle \mathbf{x}, \mathbf{y} \rangle$. Matrices are written by using bold capital letters, for example, $\boldsymbol{X}$, and the i-th column vector of a matrix is denoted by $\boldsymbol{x}_i$. The $n \times n$ identity matrix is denoted by $\boldsymbol{I}_n$.

2.1. Homomorphic Encryption. Let $\mathscr{M}$ and $\mathscr{C}$ be the message and ciphertext space. A homomorphic encryption scheme consists of four algorithms $\{KeyGen, Enc, Dec, Eval\}$.

(i) $\boldsymbol{KeyGen}(1^\lambda)$: input security parameter λ and output a public encryption key pk, a secret decryption key sk, and a public evaluation key evk.

(ii) $\boldsymbol{Enc}_{pk}(m)$: input public key pk and plaintext $m \in \mathscr{M}$ and output ciphertext $c \in \mathscr{C}$.

(iii) $\boldsymbol{Dec}_{sk}(c)$: input secret key sk and ciphertext c and output the message encrypted in the ciphertext c.

(iv) $\boldsymbol{Eval}_{evk}(f, c_1, c_2, \ldots, c_l)$: input the evaluation key evk, function f, and ciphertexts $c_1, c_2, \ldots, c_l$ and output a ciphertext $c_f \in \mathscr{C}$ that is obtained by applying the function $f : \mathscr{M}^l \longrightarrow \mathscr{M}$ to $c_1, c_2, \ldots, c_l$.

2.2. Embedding $\mathbb{Z}_q$ into Symmetric Group. According to Cayley's Theorem, the additive group $\mathbb{Z}_\mathrm{q}$ is isomorphic to a group of cyclic permutations G, where $x \in \mathbb{Z}_\mathrm{q}$ corresponds to a cyclic permutation that can be represented by an indicator vector with 1 in the $(x + 1)$-th position. The permutation matrix can be obtained from the cyclic rotation of the indicator vector. The addition in $\mathbb{Z}_\mathrm{q}$ leads to the composition of the permutations; the rounding function $\lfloor x \rceil_2 : \mathbb{Z}_q \longrightarrow \{0, 1\}$ can be computed by summing the entries of the indicator vector corresponding to those in $\mathbb{Z}_\mathrm{q}$ that round 1.

By CRT, $\mathbb{Z}_q$ is isomorphic to the direct product $\mathbb{Z}_{r_1} \times \ldots \times \mathbb{Z}_{r_t}$, where $\mathrm{q} \coloneqq \prod_{i=1}^{t} r_i$, and r_i are small and powers of distinct primes. Similarly, $\mathbb{Z}_q$ embeds into symmetric group $S = S_{r_1} \times S_{r_2} \times \ldots \times S_{r_t}$.

3. Matrix GSW-FHE

3.1. Review Matrix GSW-FHE Scheme. In this section, we review the matrix GSW-FHE scheme. Let λ be the security parameter. The matrix GSW-FHE scheme is parameterized by an integer lattice dimension n, an integer modulus q, and a distribution χ over $\mathbb{Z}$ which is assumed to be sub-Gaussian; all of the parameters depend on λ. Let $l \coloneqq \lceil \log q \rceil$, $m \coloneqq O((n+r) \log q)$, and $\mathrm{N} \coloneqq (n+r) \cdot l$. Let r be the amount of bits to be encrypted, which defines the message space $\{0, 1\}^{r \times r}$. The ciphertext space is $\mathbb{Z}_q^{(n+r) \times N}$. The scheme uses the rounding function $\lfloor \cdot \rceil_2$ where, for any $x \in \mathbb{Z}_q$, $\lfloor x \rceil_2$ outputs 1 if x is close to $q/4$ and 0 otherwise. Recall that $\boldsymbol{g}^T = (1, 2, \ldots, 2^{l-1})$ and $\mathbf{G} = g^T \bigotimes I_{n+r}$.

(i) KeyGen$(1^\lambda, r)$: Sample a uniformly random matrix $\boldsymbol{A} \xleftarrow{U} \mathbb{Z}_q^{n \times m}$, secret key matrix $\mathbf{S}' \xleftarrow{R} \chi^{r \times n}$, and noise matrix $\mathbf{E} \xleftarrow{R} \chi^{r \times m}$. Let $\mathbf{S} \coloneqq [\boldsymbol{I}_r \mid\mid -\boldsymbol{S}']$ and $\mathbf{B} \coloneqq \begin{pmatrix} \boldsymbol{S}'\boldsymbol{A}+\boldsymbol{E} \\ \boldsymbol{A} \end{pmatrix} \in \mathbb{Z}_q^{(n+r) \times m}$. Let $\boldsymbol{M}_{(i,j)} \in \{0, 1\}^{r \times r} (\mathrm{i}, \mathrm{j} = 1, 2, \ldots, \mathrm{r})$ be the matrix with 1 in the $(\mathrm{i}, \mathrm{j})-$th position and 0 in the others. For all $\mathrm{i}, \mathrm{j} = 1, 2, \ldots, \mathrm{r}$, first sample $\boldsymbol{R}_{(i,j)} \xleftarrow{U} \{0, 1\}^{m \times N}$, and set

$$\boldsymbol{P}_{(i,j)} \coloneqq \boldsymbol{B}\boldsymbol{R}_{(i,j)} + \begin{pmatrix} \boldsymbol{M}_{(i,j)}\mathbf{S} \\ \mathbf{0} \end{pmatrix} \boldsymbol{G} \in \mathbb{Z}_q^{(n+r) \times N} \tag{1}$$

Output public key $\mathrm{pk} \coloneqq (\{\boldsymbol{P}_{(i,j)}\}_{i,j \in [r]}, \boldsymbol{B})$ and secret key $\mathrm{sk} \coloneqq \mathbf{S}$.

(ii) $\mathrm{SecEnc}_{sk}(\boldsymbol{M} \in \{0, 1\}^{r \times r})$: Sample random matrixes $\mathbf{A}' \xleftarrow{U} \{0, 1\}^{n \times N}$ and $\mathbf{E}' \xleftarrow{R} \chi^{r \times N}$, parse $\mathbf{S} = [\boldsymbol{I}_r \mid\mid -\boldsymbol{S}']$, and output the ciphertext

$$\mathrm{C} \coloneqq \left[\begin{pmatrix} \boldsymbol{S}'\boldsymbol{A}' + \boldsymbol{E}' \\ \boldsymbol{A}' \end{pmatrix} + \begin{pmatrix} \mathbf{M}\mathbf{S} \\ \mathbf{0} \end{pmatrix} \boldsymbol{G} \right]_q \in \mathbb{Z}_q^{(n+r) \times N}. \tag{2}$$

(iii) $\mathrm{PubEnc}_{sk}(pk, \boldsymbol{M} \in \{0, 1\}^{r \times r})$: Sample a random matrix $\mathbf{R} \xleftarrow{U} \{0, 1\}^{m \times N}$, and output the ciphertext

$$\mathrm{C} \coloneqq \boldsymbol{B}\mathbf{R} + \sum_{i,j \in [r]: M_{[i,j]}=1} \boldsymbol{P}_{(i,j)} \in \mathbb{Z}_q^{(n+r) \times N}, \tag{3}$$

where $\boldsymbol{M}_{[i,j]}$ is the (i, j) − th element of $\boldsymbol{M}$.

(iv) $Dec_{sk}(sk, C)$: Output the matrix $\boldsymbol{M} = (\lfloor \langle \boldsymbol{s}_i, c_{jl-1} \rangle \rceil_2)_{i,j \in [r]}$, where $\boldsymbol{s}_i^T$ is the i^{th} row of $\mathbf{S}$.

3.2. Deterministic Asymmetric Encryption. We define a new deterministic asymmetric encryption algorithm in the matrix GSW-FHE scheme as follows:

(i) $\mathrm{DetePubEnc}_{pk}(\boldsymbol{M} \in \{0, 1\}^{r \times r})$: input pk and $\boldsymbol{M} \in \{0, 1\}^{r \times r}$ and output the ciphertext

$$\mathrm{C} \coloneqq \sum_{i,j \in [r]: M_{[i,j]}=1} \boldsymbol{P}_{(i,j)} \in \mathbb{Z}_q^{(n+r) \times N}, \tag{4}$$

where $\boldsymbol{M}_{[i,j]}$ is the (i, j) − th element of $\boldsymbol{M}$. The DetePubEnc algorithm has lower computational cost than SecEnc algorithm and PubEnc algorithm, and it only involves matrix addition, whereas the SecEnc algorithm and PubEnc algorithm involve both matrix multiplication and matrix addition.

4. Analysis on Matrix GSW-FHE

In the KeyGen algorithm of matrix GSW-FHE, $\boldsymbol{M}_{(i,j)}\mathbf{S}$ needs to be computed when generating public key $\boldsymbol{P}_{(i,j)}$. We observe that

$$\boldsymbol{M}_{(i,j)}\mathbf{S} = \boldsymbol{M}_{(i,j)}\left(\boldsymbol{I}_r \parallel -\boldsymbol{S}'\right) = \left(\begin{array}{c|ccc} \boldsymbol{M}_{(i,j)} & & \mathbf{0} & \\ & -s_{j1}{}' & \cdots & -s_{jn}{}' \\ & & \mathbf{0} & \end{array}\right), \tag{5}$$

where right matrix is with $(-s'_{j1}, \ldots, -s_{jn}{}')$ in the i-th row and 0 in other rows. Let $\boldsymbol{M}_{(i,j)}{}' \in \mathbb{Z}_q^{n\times n}$ be an n × n matrix, which satisfies the following matrix equation:

$$\left(\boldsymbol{I}_r \;\; -\boldsymbol{S}'\right) \cdot \begin{pmatrix} \boldsymbol{M}_{(i,j)} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{M}_{(i,j)}{}' \end{pmatrix} = \left(\begin{array}{c|ccc} \boldsymbol{M}_{(i,j)} & & \mathbf{0} & \\ & -s_{j1}{}' & \cdots & -s_{jn}{}' \\ & & \mathbf{0} & \end{array}\right). \tag{6}$$

That is,

$$-\boldsymbol{S}' \cdot \boldsymbol{M}'_{(i,j)} = \begin{pmatrix} & \mathbf{0} & \\ -s_{j1}{}' & \cdots & -s_{jn}{}' \\ & \mathbf{0} & \end{pmatrix}. \tag{7}$$

Viewing the elements of $\boldsymbol{S}'$ as the equation parameter and the elements of $\boldsymbol{M}'_{(i,j)}$ as variables, we can get equations from the above matrix equation:

$$\begin{aligned}
&s'_{11}\cdot m'_{11} + \cdots + s'_{1n}\cdot m'_{n1} = 0\\
&s'_{11}\cdot m'_{12} + \cdots + s'_{1n}\cdot m'_{n2} = 0\\
&\vdots\\
&s'_{11}\cdot m'_{1n} + \cdots + s'_{1n}\cdot m'_{nn} = 0\\
&\vdots\\
&s'_{i1}\cdot m'_{11} + \cdots + s'_{in}\cdot m'_{n1} = s'_{j1}\\
&s'_{i1}\cdot m'_{12} + \cdots + s'_{in}\cdot m'_{n2} = s'_{j2}\\
&\vdots\\
&s'_{i1}\cdot m'_{1n} + \cdots + s'_{in}\cdot m'_{nn} = s'_{jn}\\
&\vdots\\
&s'_{r1}\cdot m'_{11} + \cdots + s'_{rn}\cdot m'_{n1} = 0\\
&s'_{r1}\cdot m'_{12} + \cdots + s'_{rn}\cdot m'_{n2} = 0\\
&\vdots\\
&s'_{r1}\cdot m'_{1n} + \cdots + s'_{rn}\cdot m'_{nn} = 0
\end{aligned} \tag{8}$$

According to the knowledge of linear algebra, the equations exit nontrivial solution if the rank of coefficient matrix is equal to the rank of the augmented matrix as below.

$$rank\begin{pmatrix} s'_{11} & s'_{12} & \cdots & s'_{1n}\\ & \cdots & & \\ s'_{11} & s'_{12} & \cdots & s'_{1n}\\ & \vdots & & \\ s'_{i1} & s'_{i2} & \cdots & s'_{in}\\ & \cdots & & \\ s'_{i1} & s'_{i2} & \cdots & s'_{in}\\ & \vdots & & \\ s'_{r1} & s'_{r2} & \cdots & s'_{rn}\\ & \cdots & & \\ s'_{r1} & s'_{r2} & \cdots & s'_{rn} \end{pmatrix}_{rn\times n} = rank\begin{pmatrix} s'_{11} & s'_{12} & \cdots & s'_{1n} & 0\\ & & \cdots & & \\ s'_{11} & s'_{12} & \cdots & s'_{1n} & 0\\ & & \vdots & & \\ s'_{i1} & s'_{i2} & \cdots & s'_{in} & s'_{j1}\\ & & \cdots & & \\ s'_{i1} & s'_{i2} & \cdots & s'_{in} & s'_{jn}\\ & & \vdots & & \\ s'_{r1} & s'_{r2} & \cdots & s'_{rn} & 0\\ & & \cdots & & \\ s'_{r1} & s'_{r2} & \cdots & s'_{rn} & 0 \end{pmatrix}_{rn\times(n+1)}. \tag{9}$$

That is,

$$rank\begin{pmatrix} s'_{11} & s'_{12} & \cdots & s'_{1n}\\ & \vdots & & \\ s'_{i1} & s'_{i2} & \cdots & s'_{in}\\ & \vdots & & \\ s'_{r1} & s'_{r2} & \cdots & s'_{rn} \end{pmatrix}_{r\times n} = rank\begin{pmatrix} s'_{11} & s'_{12} & \cdots & s'_{1n} & 0\\ & \vdots & & & \\ s'_{i1} & s'_{i2} & \cdots & s'_{in} & s'_{j1}\\ & & \cdots & & \\ s'_{i1} & s'_{i2} & \cdots & s'_{in} & s'_{jn}\\ & \vdots & & & \\ s'_{r1} & s'_{r2} & \cdots & s'_{rn} & 0 \end{pmatrix}_{(r+n-1)\times(n+1)}. \tag{10}$$

We denote the solution by $\overline{M}_{(i,j)}$, so we have

$$-\boldsymbol{S}' \cdot \overline{\boldsymbol{M}}_{(i,j)} = \begin{pmatrix} \mathbf{0} \\ -s_{j1}' \ \ldots \ -s_{jn}' \\ \mathbf{0} \end{pmatrix} = \boldsymbol{M}_{(i,j)} \cdot \left(-\boldsymbol{S}'\right). \quad (11)$$

From the above analysis, we can derivate the circular security of the matrix GSW-FHE scheme.

Theorem 1 (circular security). *If the equation*

$$-\boldsymbol{S}' \cdot \boldsymbol{M}'_{(i,j)} = \begin{pmatrix} \mathbf{0} \\ -s_{j1}' \ \ldots \ -s_{jn}' \\ \mathbf{0} \end{pmatrix} \quad (12)$$

exits nontrivial solution $\overline{\boldsymbol{M}}_{(i,j)}$ over $\mathbb{Z}_q$, then the matrix GSW-FHE scheme is circular secure with function $f_{M_{(i,j)}}(S)$.

Proof. Let c_1 be a ciphertext encrypting function $f_{M_{(i,j)}}(\boldsymbol{S}) = \begin{pmatrix} \boldsymbol{M}_{(i,j)}\mathbf{S} \\ \mathbf{0} \end{pmatrix} \boldsymbol{G} \in \mathbb{Z}_q^{(n+r)\times N}$, $c_1 = \boldsymbol{BR} + \boldsymbol{P}_{(i,j)}$, and $\mathbf{R} \xleftarrow{U} \{0,1\}^{m\times N}$. Then we have

$$\begin{aligned} c_1 &= \boldsymbol{BR} + \boldsymbol{P}_{(i,j)} = \boldsymbol{BR} + \boldsymbol{B} \cdot \boldsymbol{R}_{(i,j)} + \begin{pmatrix} \boldsymbol{M}_{(i,j)}\mathbf{S} \\ \mathbf{0} \end{pmatrix} \cdot \boldsymbol{G} \\ &= \begin{pmatrix} \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \cdot \begin{pmatrix} \boldsymbol{E} \\ -\boldsymbol{A} \end{pmatrix} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) \\ \boldsymbol{A} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) \end{pmatrix} + \begin{pmatrix} \boldsymbol{M}_{(i,j)}\mathbf{S} \\ \mathbf{0} \end{pmatrix} \cdot \boldsymbol{G} \\ &= \begin{pmatrix} \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \cdot \begin{pmatrix} \boldsymbol{E} \\ -\boldsymbol{A} \end{pmatrix} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) \\ \boldsymbol{A} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) \end{pmatrix} + \left(\left(\boldsymbol{M}_{(i,j)} \middle| \begin{matrix} \mathbf{0} \\ -s_{j1}' \ \ldots \ -s_{jn}' \\ \mathbf{0} \end{matrix} \right) \right) \cdot \boldsymbol{G} \end{aligned} \quad (13)$$

From (12), we have

$$\begin{aligned} c_1 &= \begin{pmatrix} \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \begin{pmatrix} \boldsymbol{E} \\ -\boldsymbol{A} \end{pmatrix} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) + \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \cdot \begin{pmatrix} \boldsymbol{M}_{(i,j)} & \mathbf{0} \\ \mathbf{0} & \overline{\boldsymbol{M}}_{(i,j)} \end{pmatrix} \boldsymbol{G} \\ \boldsymbol{A} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) \end{pmatrix} \\ &= \begin{pmatrix} \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \begin{pmatrix} \boldsymbol{E} \\ -\boldsymbol{A} \end{pmatrix} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) + \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \cdot \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \overline{\boldsymbol{M}}_{(i,j)} \end{pmatrix} \boldsymbol{G} + \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \cdot \begin{pmatrix} \boldsymbol{M}_{(i,j)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \boldsymbol{G} \\ \boldsymbol{A} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) \end{pmatrix} \\ &= \begin{pmatrix} \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \begin{pmatrix} \boldsymbol{E} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) \\ -\boldsymbol{A} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) + \overline{\boldsymbol{M}}_{(i,j)} \cdot \left(\boldsymbol{g}^T \otimes \boldsymbol{I}_n\right) \end{pmatrix} + \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \cdot \begin{pmatrix} \boldsymbol{M}_{(i,j)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \boldsymbol{G} \\ \boldsymbol{A} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) \end{pmatrix} \\ &= \begin{pmatrix} \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \begin{pmatrix} \boldsymbol{E} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) \\ -\boldsymbol{A} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) + \overline{\boldsymbol{M}}_{(i,j)} \cdot \left(\boldsymbol{g}^T \otimes \boldsymbol{I}_n\right) \end{pmatrix} \\ \boldsymbol{A} \cdot \left(\mathbf{R} + \boldsymbol{R}_{(i,j)}\right) - \overline{\boldsymbol{M}}_{(i,j)} \cdot \left(\boldsymbol{g}^T \otimes \boldsymbol{I}_n\right) \end{pmatrix} + \begin{pmatrix} \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \cdot \begin{pmatrix} \boldsymbol{M}_{(i,j)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \boldsymbol{G} \\ \overline{\boldsymbol{M}}_{(i,j)} \cdot \left(\boldsymbol{g}^T \otimes \boldsymbol{I}_n\right) \end{pmatrix} = \begin{pmatrix} \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \begin{pmatrix} \widetilde{\boldsymbol{E}} \\ -\widetilde{\boldsymbol{A}} \end{pmatrix} \\ \widetilde{\boldsymbol{A}} \end{pmatrix} \\ &+ \begin{pmatrix} \left(\boldsymbol{I}_r \ \ -\boldsymbol{S}'\right) \cdot \begin{pmatrix} \boldsymbol{M}_{(i,j)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \boldsymbol{G} \\ \overline{\boldsymbol{M}}_{(i,j)} \cdot \left(\boldsymbol{g}^T \otimes \boldsymbol{I}_n\right) \end{pmatrix} = \begin{pmatrix} \boldsymbol{S}'\widetilde{\boldsymbol{A}} + \widetilde{\boldsymbol{E}} \\ \widetilde{\boldsymbol{A}} \end{pmatrix} + \begin{pmatrix} \begin{pmatrix} \boldsymbol{M}_{(i,j)\cdot}\left(\boldsymbol{g}^T \otimes \boldsymbol{I}_r\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \\ \overline{\boldsymbol{M}}_{(i,j)} \cdot \left(\boldsymbol{g}^T \otimes \boldsymbol{I}_n\right) \end{pmatrix} \cdot \end{aligned} \quad (14)$$

$\widetilde{\boldsymbol{E}} \triangleq \boldsymbol{E} \cdot (\mathbf{R} + \boldsymbol{R}_{(i,j)})$; $\widetilde{\boldsymbol{A}} \triangleq \boldsymbol{A} \cdot (\mathbf{R} + \boldsymbol{R}_{(i,j)}) - \overline{\boldsymbol{M}}_{(i,j)} \cdot (\boldsymbol{g}^T \otimes \boldsymbol{I}_n)$; therefore, we derivate that

$$c_1 = \begin{pmatrix} \boldsymbol{S}'\widetilde{\boldsymbol{A}} + \widetilde{\boldsymbol{E}} \\ \widetilde{\boldsymbol{A}} \end{pmatrix} + \begin{pmatrix} \begin{pmatrix} \boldsymbol{M}_{(i,j)\cdot}\left(\boldsymbol{g}^T \otimes \boldsymbol{I}_r\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \\ \overline{\boldsymbol{M}}_{(i,j)} \cdot \left(\boldsymbol{g}^T \otimes \boldsymbol{I}_n\right) \end{pmatrix}. \quad (15)$$

As $(\widetilde{\boldsymbol{A}}, \boldsymbol{S}'\widetilde{\boldsymbol{A}} + \widetilde{\boldsymbol{E}})$ is an instance of LWE over $\mathbb{Z}_q^{(n+r)\times N}$, it satisfies uniform distribution over $\mathbb{Z}_q^{(n+r)\times N}$. Furthermore, c_1 obeys uniform distribution over $\mathbb{Z}_q^{(n+r)\times N}$.

On the other hand, suppose that c_0 is a ciphertext encrypting $\mathbf{0}$; that is,

$$c_0 = \boldsymbol{BR'} = \begin{pmatrix} \boldsymbol{S'A} + \boldsymbol{E} \\ \boldsymbol{A} \end{pmatrix} \cdot \boldsymbol{R'} \in \mathbb{Z}_q^{(n+r)\times N}, \quad \mathbf{R'} \xleftarrow{U} \{0,1\}^{m\times N}. \tag{16}$$

It is also an instance of LWE over $\mathbb{Z}_q^{(n+r)\times N}$ and obeys uniform distribution over $\mathbb{Z}_q^{(n+r)\times N}$, too. Therefore, distributions of c_0 and c_1 are computationally indistinguishable, and the advantage of probabilistic polynomial-time adversary $\mathcal{A}$ is negligible. So we can conclude that the matrix GSW-FHE is circular secure with function $f_{M_{(i,j)}}(S)$.

From Theorem 1, we can choose a good secret key that satisfies that (12) has solution via "reject sample" technique and obtain circular secure matrix GSW-FHE scheme. □

5. Optimizing Bootstrapping

In this section, we describe how to optimize the bootstrapping procedure of [13] by introducing deterministic homomorphic plaintext slot-wise permutation.

5.1. Motivation. The decryption of all LWE-based FHE schemes consists of the inner product and rounding: for secret key $s \in \mathbb{Z}_q^d$ and a binary ciphertext $\boldsymbol{c} \in \{0,1\}^d$, the decryption algorithm computes

$$\mathrm{Dec}(s, c) = \lfloor \langle s, c \rangle \rceil_2 \in \{0,1\}. \tag{17}$$

Note that the inner product itself is just a subset-sum of the $\mathbb{Z}_q$-entries of s indicated by c and uses only the additive group structure of $\mathbb{Z}_q$. Alperin-Sheriff and Peikert [20] proposed an efficient bootstrapping algorithm by embedding $\mathbb{Z}_q$ into permutation group S_q. Thus the rounding function is no longer just a sum, and it can be expressed as

$$\lfloor x \rceil_2 = \sum_{v\in\mathbb{Z}_q \; s.t. \lfloor v \rceil_2 = 1} [x = v], \tag{18}$$

where each equality test $[x = v]$ returns 0 for false and 1 for true. The equality test operation has homomorphic counterpart, called homomorphic equality test. Homomorphic equality test is an important primitive for optimizing bootstrapping procedure, and it has many other applications as mentioned in [25].

For $x, v \in \mathbb{Z}_r$, they map to the r-by-r permutation matrices of group S_r and are denoted as τ *and* σ, respectively. The Eq? algorithm is described as follows:

(i) **Eq?** $(C^\tau = c_{i,j}^\tau, \sigma \in S_r)$: given a ciphertext encrypting some permutation $\tau \in S_r$ and a permutation $\sigma \in S_r$ (in the clear), output a ciphertext c encrypting 1 if $\tau = \sigma$; otherwise, output a ciphertext c encrypting 0:

$$\mathrm{c} \longleftarrow \boxdot_{i\in[r]} c_{\sigma(i),j}^\tau \boxdot \mathrm{g}. \tag{19}$$

Note that the permutation σ goes through all permutations in S_r, and it is not masked in the homomorphic equality test **Eq?** Algorithm; that is, $\sigma \in S_r$ is *in the clear*.

Let φ_i: $\mathbb{Z}_q \longrightarrow \{0,1\}^r$ be the isomorphism of an element in $\mathbb{Z}_q$ ($\mathrm{q} := \prod_{i=1}^t r_i$) into the cyclic permutation that corresponds to an element in $\mathbb{Z}_{r_i}$, where $\mathrm{r} \triangleq \max_i\{r_i\}$. During homomorphic rounding process of work [13], $\varphi_i(x)$ is encrypted as part of public bootstrapping key and used in the homomorphic equality test algorithm.

In fact, x traverses $\mathbb{Z}_q$ and does not carry any privacy information. It is not necessary to encrypt $\pi_{\varphi_i(x)}$ using SecEnc algorithm, which would increase computation cost. We propose optimizing homomorphic equality test algorithm by defining hybrid homomorphic plaintext slot-wise switching method, which reduces the computation cost of bootstrapping key generation.

5.2. Hybrid Homomorphic Plaintext Slot-Wise Switching. Plaintext slot-wise permutation is an important operation in application of packed FHE [23, 24]. It can be achieved by multiplying the encryption of a permutation and its inverse from left and right. We propose hybrid homomorphic plaintext slot switching procedure where the switch key is encrypted by symmetric and asymmetric encryption algorithm. The nice feature of our switching procedure is that part of switch key can be computed by deterministic public encryptions, which makes our procedure more efficient than that of [13].

(i) SwitchKeyGen$(\boldsymbol{S}, \sigma)$: Input a secret key matrix $\boldsymbol{S} \in \mathbb{Z}_q^{r\times(n+r)}$ and a permutation σ; let $\pi_\sigma \in \{0,1\}^{r\times r}$ be a matrix corresponding to σ, and compute

$$W_\sigma \longleftarrow \mathrm{SecEnc}_S(\pi_\sigma), \quad W_{\sigma^{-1}} \longleftarrow \mathrm{SecEnc}_S(\pi_\sigma{}^T). \tag{20}$$

Output the switch key $\mathrm{ssk}_\sigma := (W_\sigma, W_{\sigma^{-1}})$. The algorithm is the same as the work in [13].

(ii) $SlotSwitch_{ssk_\sigma}(C)$: Input a switch key ssk_σ and a ciphertext C; output

$$C_\sigma \longleftarrow W_\sigma \bigodot \left(C \bigodot \left(W_{\sigma^{-1}} \bigodot G\right)\right), \tag{21}$$

where $G \in \mathbb{Z}_q^{(n+r)\times N}$ is the fixed encryption of I_r with noise zero.

(iii) DeteSwitchKeyGen$(\boldsymbol{S}, \sigma)$: Input a secret key matrix $\boldsymbol{S} \in \mathbb{Z}_q^{r\times(n+r)}$ and a permutation σ, and compute

$$DW_\sigma \longleftarrow \mathrm{DetePubEnc}_S(\pi_\sigma), \quad DW_{\sigma^{-1}} \longleftarrow \mathrm{DetePubEnc}_S(\pi_\sigma{}^T). \tag{22}$$

Output the deterministic switch key $\mathrm{dssk}_\sigma := (DW_\sigma, DW_{\sigma^{-1}})$.

(iv) $DeteSlotSwitch_{dssk_\sigma}(C)$: Input a deterministic switch key dssk_σ and a ciphertext C; output

$$C_\sigma \longleftarrow DW_\sigma \bigodot \left(C \bigodot \left(DW_{\sigma^{-1}} \bigodot G\right)\right), \tag{23}$$

where $G \in \mathbb{Z}_q^{(n+r)\times N}$ is the fixed encryption of I_r with noise zero.

5.3. Optimized Bootstrapping Procedure. Our optimized bootstrapping procedure can be used to refresh ciphertexts of all standard LWE-based FHE. Let $\boldsymbol{c} \in \{0,1\}^d$ be the ciphertext to be bootstrapped, and let $s \in \mathbb{Z}_q^d$ be a secret key that corresponds to $\boldsymbol{c}$. The optimized bootstrapping procedure consists of two algorithms, HybirdBootKeyGen and HybirdBootstrap.

(i) **HybridBootKeyGen**$(pk, sk, \boldsymbol{s})$: Input a secret key sk and public key pk for our bootstrapping scheme and the secret key $\mathbf{s} = (s_1, \ldots, s_d) \in \mathbb{Z}_q^d$ for ciphertext to be refreshed; output a bootstrapping key bk. For every $i \in [t]$ and $j \in [d]$, let $\pi_{\varphi_i(s_j)}$ be the permutation corresponding to $\varphi_i(s_j)$, and generate

$$\begin{aligned} \tau_{i,j} &\xleftarrow{R} \mathrm{SecEnc}_{sk}\left(\mathrm{diag}\left(\varphi_i\left(s_j\right)\right)\right), \\ ssk_{i,j} &\xleftarrow{R} \mathrm{SwitchKeyGen}\left(\mathrm{sk}, \pi_{\varphi_i(s_j)}\right), \end{aligned} \tag{24}$$

where, for a vector $x \in \mathbb{Z}^r$, $\mathrm{diag}(x) \in \mathbb{Z}^{r\times r}$ is the square integer matrix that has x in its diagonal entries and 0 in the others. Then compute the hints used in homomorphic equality test on packed indictor vectors. For every $i \in [t]$ and $x \in \mathbb{Z}_q$ such that $\lfloor x \rceil_2 = 1$, compute

$$dssk_{\varphi_i(x)} \longleftarrow DeteSwitchKeyGen\left(sk, \pi_{\varphi_i(x)}\right). \tag{25}$$

Output the bootstrapping key

$$bk := \left\{\tau_{i,j}, ssk_{i,j}, dssk_{\varphi_i(x)}\right\}_{i\in[t], j\in[d], x\in\mathbb{Z}_q:\lfloor x\rceil_2=1}. \tag{26}$$

(ii) $\boldsymbol{HybridBootstrap}_{bk}(\mathbf{c})$: Input a bootstrapping key bk and a ciphertext $\boldsymbol{c} \in \{0,1\}^d$; output the refreshed ciphertext C^*. All the FHE schemes based on the LWE problem have similar decryption algorithm; that is, the decryption algorithm needs to compute $\lfloor \langle s, c\rangle \rceil_2$. There are two phases in the HybridBootstrap algorithm: evaluate the inner product and rounding.

Inner Product: For every $i \in [t]$, homomorphically compute an encryption of $\varphi_i(\langle s, c\rangle)$. Let $h := \min\{j \in [d] : c_j = 1\}$. For $i = 1, 2, \ldots, t$, set $C_i^* := \tau_{i,h}$, and iteratively compute

$$C_i^* \xleftarrow{R} SlotSwitch_{ssk_{i,j}}\left(C_i^*\right) \tag{27}$$

for $j = h+1, \ldots, d$ such that $c_j = 1$.

Rounding: For each $x \in \mathbb{Z}_q$ such that $\lfloor x \rceil_2 = 1$, homomorphically test the equality between x and $\langle s, c\rangle$, and sum their results. The refreshed ciphertext is computed as

$$C^* \longleftarrow \bigoplus_{x\in\mathbb{Z}_q:\lfloor x\rceil_2=1} \left(\bigodot_{i\in[t]}\left(DeteSlotSwitch_{dssk_{\varphi_i(x)}}\left(C_i^*\right)\right) \bigodot P_{1,1}\right). \tag{28}$$

5.4. Correctness Analysis

Lemma 2 (correctness). *Let sk be the secret key for our scheme. Let c and s be a ciphertext and secret key of LWE-based FHE scheme. Then, for $bk \longleftarrow \mathrm{HybridBootKeyGen}(pk, sk, \boldsymbol{s})$, the refreshed ciphertext $C^* \longleftarrow HybridBootstrap_{bk}(\mathrm{c})$ is designed to encrypt $Dec_s(c) = \lfloor \langle s, c\rangle \rceil_2 \in \{0,1\}$ in the first slot.*

Proof. Firstly, C_i^* is designed to encrypt $\varphi_i(\lfloor \langle s, c\rangle \rceil_q)$, and

$$\bigodot_{i\in[t]}\left(DeteSlotSwitch_{dssk_{\varphi_i(x)}}\left(C_i^*\right)\right) \bigodot P_{1,1} \tag{29}$$

is designed to encrypt 1 in the first slot if and only if $x = \langle s, c\rangle \bmod q$. Finally, since the homomorphic sum is taken over every $x \in \mathbb{Z}_q$ such that $\lfloor x \rceil_2 = 1$, C^* is designed to encrypt 1 if and only if $\lfloor \langle s, c\rangle \rceil_2 = 1$. □

5.5. Security Analysis. If the bootstrapping scheme secret key sk is generated independently of the secret keys s of FHE scheme from LWE, then Ind-CPA security of the bootstrapping key follows immediately from the security of hybrid homomorphic plaintext slot-wise switching, and the security of hybrid homomorphic plaintext slot-wise switching scheme resorts to the security of matrix GSW-FHE and hence the security of our bootstrapping scheme from LWE assumption.

5.6. Performance Analysis. Let $q = \widetilde{O}(\lambda)$ be the modules of the ciphertext to be refreshed, and q has the form $q := \prod_{i=1}^{t} r_i$, where r_i are small and powers of distinct primes. The following lemma allows us to choose a sufficiently large q by letting it be the product of all maximal prime powers r_i bounded by $O(\log \lambda)$, and then there exists $t = O(\log \lambda / \log\log \lambda)$, where λ is security parameter.

Lemma 3 (see [13, 20]). *For all $x \geq 7$, the product of all maximal prime powers $r_i \leq x$ is all at least* $\exp(3x/4)$.

On one hand, our DetePubEnc algorithm involves matrix additions operation only, whereas SecEnc algorithm involves many matrix multiplication operations. Our bootstrapping key $dssk_{\varphi_i(x)}$ is optimized from $ssk_{\varphi_i(x)}$. Therefore, our optimized bootstrapping key generation has lower computation complexity. The comparison of computational complexity is illustrated in Table 1.

On the other hand, we may implement a trade-off between computation and storage complexity. For every $k, l \in [r]$, $P_{k,l} = \mathrm{SecEnc}_{sk}(M_{k,l})$ can be used as public bootstrapping key, delete $dssk_{\varphi_i(x)}$ from the bootstrapping key, and compute $dssk_{\varphi_i(x)}$ online when running rounding procedure. In view of $dssk_{\varphi_i(x)}$ being obtained by DetePubEnc algorithm, its computation involves only matrix additions. Therefore, our optimized bootstrapping drastically cuts down the size of the large public bootstrapping key by a third, paying matrix additions with negligible computation complex. The comparison of storage complexity is illustrated in Table 2.

Table 1: Comparison of computational complexity.

Bootstrapping key	MM	MA
$ssk_{\varphi_i(x)}$ [13],$0 \le i \le t$	$O(\log\lambda/\log\log\lambda)$	$O(\log\lambda/\log\log\lambda)$
$dssk_{\varphi_i(x)}$ [ours], $0 \le i \le t$	0	$O(\log^2\lambda/\log\log\lambda)$

Note: MM denotes matrix multiplication operation; MA denotes matrix addition operation.

Table 2: Comparison of storage complexity of bootstrapping key.

Work	Bootstrapping key
[13]	$\{(\tau_{i,j}, ssk_{i,j}, ssk_{\varphi_i(x)})\}_{i\in[t],j\in[d],x\in\mathbb{Z}_q:\lfloor x\rceil_2=1}$
[ours]-1	$\{(\tau_{i,j}, ssk_{i,j}, dssk_{\varphi_i(x)})\}_{i\in[t],j\in[d],x\in\mathbb{Z}_q:\lfloor x\rceil_2=1}$
[ours]-2	$\{(\tau_{i,j}, ssk_{i,j})\}_{i\in[t],j\in[d]}$

Note: [ours]-1 denotes save computation complexity in the cost of the storage complexity; [ours]-2 denotes save storage complexity in the cost of computation complexity.

6. Conclusions

Matrix GSW-FHE scheme encrypts multibit message and supports complex homomorphic matrix operations and can be used to optimize the bootstrapping procedure. We analyse circular security of matrix GSW-FHE scheme and derive a sufficient condition of circular security for matrix GSW-FHE scheme. That is, if the equations about secret key have solution over $\mathbb{Z}_q$, the matrix GSW-FHE scheme satisfies circular security with function $f_{M_{(i,j)}}(S)$. Therefore, we can choose a good secret key that satisfies the sufficient condition via "reject sample" technique and furthermore obtain circular secure matrix GSW-FHE scheme.

We also propose hybrid homomorphic plaintext slot-wise switching method by defining deterministic public encryption algorithm in matrix GSW-FHE, which significantly reduces computational complex or space complex of bootstrapping key generation, thus optimizing the bootstrapping procedure of Hiromasa and so forth. Meanwhile, performance analysis validates the effectiveness of the proposed optimized bootstrapping scheme.

Some questions remain for further study, such as the probability analysis of our sufficient condition and the sufficient and necessary condition for circular security of the matrix GSW-FHE scheme [26]. And to make a fair comparison with the state-of-the-art bootstrapping schemes such as FHEW [21], WT [22], and so forth, detailed security, parameters, and efficiency experiment analysis remain to be a future work.

Data Availability

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

Disclosure

The abstract of this manuscript has been submitted to the 4th International Conference on Cloud Computing and Security, but it has not been published; and this manuscript cites the conference paper in the references.

Conflicts of Interest

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

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant no. 61601515 and Natural Science Foundation of Henan Province under Grant no. 162300410332.

References

[1] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[2] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st annual ACM symposium on Theory of Computing (STOC '09)*, pp. 169–178, ACM, Bethesda, Md, USA, 2009.

[3] C. Gentry, *A fully homomophic encryption scheme [Ph.D. thesis]*, Stanford University, 2009, http://crypto.stanford.edu/craig.

[4] Y. Liu, H. Peng, and J. Wang, "Verifiable diversity ranking search over encrypted outsourced data," *CMC*, vol. 55, no. 1, pp. 37–57, 2018.

[5] W. Xu, S. Xiang, and V. Sachney, "A cryptography domain image retrieval method based on Paillier homomorphic block encryption," *CMC*, vol. 55, no. 2, pp. 285–295, 2018.

[6] R. Xie, C. He, D. Xie, C. Gao, and X. Zhang, "A Secure Ciphertext Retrieval Scheme against Insider KGAs for Mobile Devices in Cloud Storage," *Security and Communication Networks*, vol. 2018, Article ID 7254305, 7 pages, 2018.

[7] R. L. Rivest, L. Adleman, and M. L. Dertouzos, *On Data Banks And Privacy Homomorphism Proc of Foundations of Secure Computation*, Academic Press, New York, NY, USA, 1978.

[8] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS '11)*, pp. 97–106, Palm Springs, Calif, USA, October 2011.

[9] M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," *Journal of Mathematical Cryptology*, vol. 9, no. 3, pp. 169–203, 2015.

[10] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent

messages," in *Advances in Cryptology—CRYPTO 2011*, R. Phillip, Ed., vol. 6841, pp. 505–524, Springer, Berlin, Germany, 2011.

[11] F. Luo, F. Wang, K. Wang, J. Li, and K. Chen, "LWR-Based Fully Homomorphic Encryption," *Security and Communication Networks*, vol. 2018, Article ID 5967635, 12 pages, 2018.

[12] X. Yang, T. Zhou, W. Zhang, and L. Wu, "Application of a circular secure variant of LWE in the homomorphic encryption," *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, vol. 52, no. 6, pp. 1389–1393, 2015.

[13] R. Hiromasa, M. Abe, and T. Okamoto, "Packing messages and optimizing bootstrapping in GSW-FHE," in *Public-key cryptography—PKC 2015*, vol. 9020 of *Lecture Notes in Comput. Sci.*, pp. 699–715, Springer, Heidelberg, 2015.

[14] D. Hofheinz and D. Unruh, "Towards key-dependent message security in the standard model," in *Advances in cryptology—EUROCRYPT 2008*, vol. 4965 of *Lecture Notes in Comput. Sci.*, pp. 108–126, Springer, Berlin, 2008.

[15] I. Haitner and T. Holenstein, "On the (im)possibility of key dependent encryption," in *Theory of cryptography*, vol. 5444 of *Lecture Notes in Comput. Sci.*, pp. 202–219, Springer, Berlin, 2009.

[16] D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky, "Circular-secure encryption from decision Diffie-Hellman," in *Advances in Cryptology*, D. Wagner, Ed., vol. 5157 of *Lecture Notes in Computer Science*, pp. 108–125, Springer, 2008.

[17] B. Applebaum, D. Cash, C. Peikert, and A. Sahai, "Fast cryptographic primitives and circular-secure encryption based on hard learning problems," in *Advances in Cryptology—CRYPTO 2009*, vol. 5677 of *Lecture Notes in Computer Science*, pp. 595–618, Springer, Germany, Berlin, 2009.

[18] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*, pp. 84–93, ACM, Baltimore, Md, USA, May 2005.

[19] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," *Proceedings of CRYPTO 2013*, vol. 8042, no. 1, pp. 75–92, 2013.

[20] J. Alperin-Sheriff and C. Peikert, "Faster bootstrapping with polynomial error," in *Proceedings of the International Cryptology Conference*, pp. 297–314, Springer, Berlin, Germany, 2014.

[21] L. Ducas and D. Micciancio, "FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second," in *Proceedings of the Advances in Cryptology – EUROCRYPT*, pp. 617–640, Springer Berlin Heidelberg, 2015.

[22] H. Wang and Q. Tang, "Efficient homomorphic integer polynomial evaluation based on GSW FHE," *The Computer Journal*, vol. 61, no. 4, pp. 575–585, 2018.

[23] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," *Designs, Codes and Cryptography*, vol. 71, no. 1, pp. 57–81, 2014.

[24] Z. Brakerski, C. Gentry, and S. Halevi, "Packed Ciphertexts in LWE-Based Homomorphic Encryption," in *Public-Key Cryptography – PKC 2013*, vol. 7778 of *Lecture Notes in Computer Science*, pp. 1–13, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[25] Y. Wang, H. Pang, N. H. Tran, and R. H. Deng, "CCA Secure encryption supporting authorized equality test on ciphertexts in standard model and its applications," *Information Sciences*, vol. 414, pp. 289–305, 2017.

[26] X. Zhao, H. Mao, S. Liu, and W. Song, "Circular-secure analysis on matrix GSW-FHE and optimizing bootstrapping," in *Proceedings of the International Conference on Cloud Computing and Security, ICCCS 2018*, 2018.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 4036815, 12 pages
https://doi.org/10.1155/2018/4036815

Research Article

Robust Visual Secret Sharing Scheme Applying to QR Code

Longdan Tan, Kesheng Liu, Xuehu Yan, Lintao Liu, Tianqi Lu, Jinrui Chen, Feng Liu, and Yuliang Lu

National University of Defense Technology, Anhui, 230037, China

Correspondence should be addressed to Xuehu Yan; publictiger@126.com

Received 18 July 2018; Accepted 8 November 2018; Published 11 December 2018

Guest Editor: Naixue Xiong

Different color patterns of quick response (QR) codes, such as RGB, grayscale, and binary QR codes, are widely used in applications. In this paper, we propose a novel XOR-based visual secret sharing (VSS) scheme using grayscale QR codes as cover images and binary QR code as secret image. First, all the codewords of the secret QR code image are encoded into n temporary binary QR code images, which are substituted for the second significant bit planes of the grayscale QR code cover images to generate n shares. Each share is a grayscale QR code image, which can be decoded by a standard QR code decoder, so that it may not attract the attention of potential attackers when distributed in the public channel. The secret image can be recovered by XORing the codewords regions of QR codes which are extracted from the second significant bit planes of the grayscale shares. More importantly, the proposed scheme is robust to JPEG compression, addition of different noises, rotation, resizing, and cropping, which is useful in practice. The effectiveness and robustness of our scheme are shown by the experimental results. The application of QR code is suitable for wireless multimedia data security.

1. Introduction

Quick response(QR) code is a kind of two-dimensional matrix codes used widely in all walks of life recently, for the advantages of its speed reading, error correction capability, multiple-format data representing, high capacity compared to one-dimensional codes, and so on. And the application of QR code is suitable for wireless multimedia data security.

QR code image is a binary image in general, each pixel in the image has only two possible values or gradation levels, which is represented by black and white ($B\&W$) or monochrome image. By graying binary QR code image, we can obtain grayscale QR code image. Each pixel of grayscale QR code image takes up 8 bits of storage space. Keeping the most significant bit plane of the grayscale QR code image unchanged and then replacing the other less significant ones with secret bits could implement information hiding in grayscale QR code image.

The main idea of visual cryptography (VC) [1], also known as visual secret sharing (VSS), is to split a secret image into numerous shares (also called share images or shadow images). Each share reveals no information on the original secret image separately, and only a specific amount of qualified shares could reconstruct the secret image. The secret image would be lost when storing and transmitting in a single image carrier which could be damaged easily, so that we may fail to extract the secret image. But, VSS can overcome the problem to a certain degree [2].

The threshold-based VSS was first proposed by Shamir [3] and Blakley [4]. Sharing one binary secret image into n corresponding random shares and then distributing them to n participants are the way to share. More than or equal to k shares are superimposed to reveal the secret image visually. However, less than k participants would reveal no information on the original secret image by stacking or inspecting their shares. The advantage of VSS in [1] is that it is easier to recover the secret image by stacking a specific amount of shares using HVS without any cryptographic knowledge and computations. However, the characteristics of pixel expansion and codebook (basic matrices) design may be be problematic in some situations.

Since Kafri and Keren proposed Random grid (RG)-based VSS [5–7], it has received much more attention, in which the pixel expansion problem is not exist and codebook design is not required. The secret image is shared into noise-like shares with the same size of the secret image.

The decryption method is the same as traditional VSS, i.e., stacking. However, the background of the reconstructed image becomes darker and darker when more and more shares are stacked in OR-based VSS (OVSS) based on RG.

XOR-based VSS (XVSS) can solve the problem of RG-based OVSS [8], since the decryption method is to perform XOR operation on the shares with a light weight computational device to reconstruct the secret image. By applying the XVSS, better image contrast and quality can be obtained [6, 9].

Due to the advantage of VSS and QR code, some combinations of them have been proposed by many researchers recently. Jonathan and Yan [12] authenticated the shares using a QR code. Wang et al. [13] proposed a scheme through embedding QR codes into the best region of given shares to prevent cheating. Chow et al. [11] proposed a (n, n) threshold scheme for cases that n is no less than 3, in which the secret image and cover images are all binary QR codes with the same version and the same level of error correction. Wan et al. [10] proposed a scheme that deeply integrated the QR code error correction mechanism with the theory of VSS and the region shared with the secret image is continuous. Chen et al. [14] proposed a VSS scheme with high security and flexible access structures for QR code applications. Chow et al. [15] investigated a method to distribute shares through embedding them into QR codes cover by a secure way using cryptographic keys. However, robustness is not considered in the above mentioned schemes, which is important in practice. The threshold is greater than 2 in [11], and the problem of image quality uniformity is not taken into account in paper [10].

Nowadays, more and more QR codes are used on mobile phones. A possible scenario to convey secret information securely in the network is described below. It is unsafe to transmit secret information in the public channel without protection, and mobile devices are widely used nowadays. Thus we encode secret information in QR code, which is shared into n different QR code cover images to generate n corresponding shares. Then the shares are transmitted in different channels over a network from one mobile phone to anther mobile phone. When JPG compression and recoding, Gaussian noise and other image attacks occur during the transmission, if the shares are robust to the attacks, secret information would be recovered, while less than n shares cannot reveal any information on the secret image. The decoding results of shares are identical with those of cover images, so they will not come into notice.

The schemes above are not suitable for the scenario. So, we propose a novel robust secret sharing scheme for (n, n) threshold that n is not less than 2. This scheme integrates the QR code error correction mechanism with the theory of XVSS, and all the QR codes are with the same version and level of error correction. First, all the codewords of a secret QR code image are encoded into n temporary binary QR code images. Then the n temporary binary QR code images are substituted for the second significant bit planes of the grayscale cover QR code images to output n shares. The selection of codewords of secret QR code is random and each share can be decoded by a standard QR code reader, which can reduce the likelihood of suspicion and potential attacking. Since the modifications of grayscale QR codes are the second significant bit planes and the error correction mechanism of QR code, the scheme is robust to the conventional image attacks.

The rest of the paper is organized as follows. QR codes and XOR-based VSS are introduced in Section 2. The secret image sharing, recovering algorithm, and analyses are described in Section 3. Section 4 demonstrates the experimental results, comparisons and test. Finally, Section 5 is the conclusion of this paper.

2. Preliminaries

2.1. QR Codes. A QR code symbol [17] consists of a square array consisting of square modules, which is developed by Denso Corporation of Japan in September 1994. The standard [16] defines forty versions of QR code versions ranging from version one to version forty. Different versions of QR code are comprised of different quantities of modules. QR code version 1 is made up of $21*21$ modules. From version 1, each version has 4 modules per side more than the previous version. For example, version 7 is made up of $45*45$ modules.

A QR code [16] consists of functional patterns and encoding regions. The encoding region includes data and error correction codewords, format information and version information. The functional patterns consist of alignment, timing, finding patterns, and separation. The amounts of data and error correction codewords and error correction blocks are relying on the version and level of error correction of the QR code. The quiet zone is the blank region around QR code that is important for reading the QR code, encoding input data stream into an array of data codewords with 8-bits length. Error correction codewords also with 8-bits length are generated by using Reed-Solomon error control algorithm which is added to the back of the data codewords.

Depended on the QR version and the level of error correction, data codewords and error correction codes are arranged in different error correction blocks. The level of error correction is divided into four categories: $L \sim 7\%$, $M \sim 15\%$, $Q \sim 25\%$, and $H \sim 30\%$. The higher the level of error correction is, the stronger the error correction ability will be. But, high level of error correction requires larger QR version to encode the same input data stream, since the proportion of error correction codewords is larger than lower level in the same version. For example, the error correction characteristics of version 7 and 8 are shown in Table 1.

In Table 1, the error correction blocks are given as (c, k, r), where c is the total amount of codewords, k is the amount of codewords and r is the error correction ability. Note that the (c, k, r) values are different for certain level of error correction in some QR code versions. It can be found that there are 5 error correction blocks in version 7 with an level of error correction of H. The (c, k, r) values for the first four blocks are $(39, 13, 13)$ where the value for the last one is $(40, 14, 13)$.

The layout of the codewords for each error correction block is in an interleaving manner, and error correction codewords are appended to the end of corresponding data codewords. Since the characteristic of this layout, the QR code decoding ability can be improved for the case of the

Table 1: The characteristics of QR code version 7 and version 8.

Version	Total codewords	Error correction level	Number of error correction codewords	Number of blocks	Error correction code per block* (c, k, r)
7	196	L	4	2	(98, 7810)
		M	72	4	(49, 31, 9)
		Q	108	2	(32, 14, 9)
				4	(33, 15, 9)
		H	130	4	(39, 13, 13)
				1	(40, 14, 13)
8	242	L	48	2	(121, 97, 12)
		M	88	2	(60, 38, 11)
				2	(61, 39, 11)
		Q	132	4	(40, 18, 11)
				2	(41, 19, 11)
		H	156	4	(40, 14, 13)
				2	(41, 15, 13)

*(c, k, r) :c=total number of codewords; k=total number of datawords; r=error correction capacity.

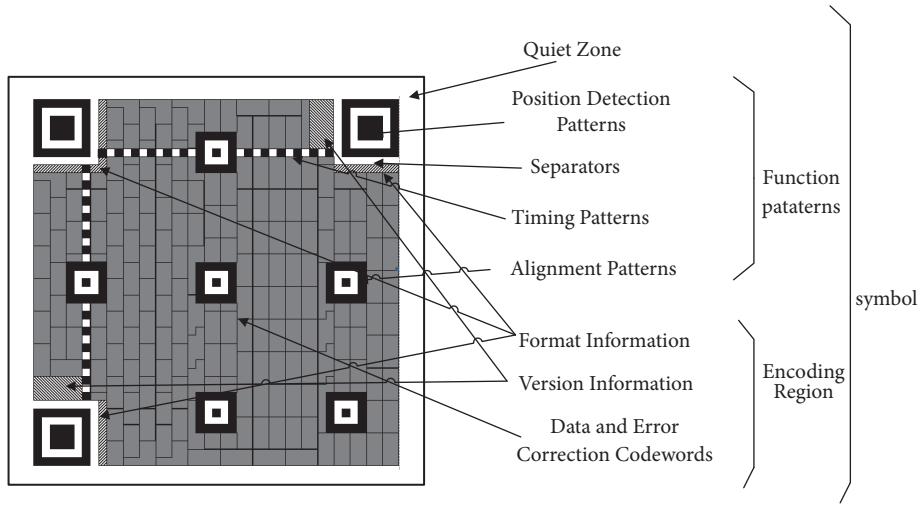


Figure 1: The structure of QR code version 7.

possibility of localized damages. Figure 1 shows the layout of QR code version 7 with the level of error correction of H. After encoding all the codewords, a data mask is required to be applied to the encoding area. There are eight data mask patterns options for balancing the light and dark modules and avoiding the confusion encoding area with functional patterns.

The layout of the codewords and the error correction capability are important for us to share the secret QR code images in the proposed scheme.

2.2. (n, n) *Threshold XOR-Based VSS.* The scheme of (n, n) threshold XOR-based VSS is to share a secret image into n corresponding noise-like shares in n participants and each share in every participant is different from each other. Then, the secret image can recover losslessly only by XORing n shares with a computational device with XOR ability. Less than n participants cannot obtain any information about the original secret image by inspecting or stacking their shares.

Figure 2 shows (2,2) XOR-based VSS application example. The generation phase is the same as that of (2, 2) RG-based VSS, while the recover method is to XOR two shares instead of stacking the shares to reveal the original secret image. By XORing the shares, the secret image can be recovered losslessly.

3. Proposed Robust VSS Appling to QR Codes

3.1. The Main Idea. In this section, we propose an (n, n) threshold XVSS scheme based on QR code. The secret image is a binary QR code, whose codewords are all shared into the codewords of n different binary QR codes with the same version and level of error correction, where their functional patterns are the same. All codewords of the secret image are shared with the theory of XVSS. Then the n binary QR codes are substituted for the second significant bit plane of n grayscale QR code cover images with the same version and level of error correction to generate n grayscale QR code

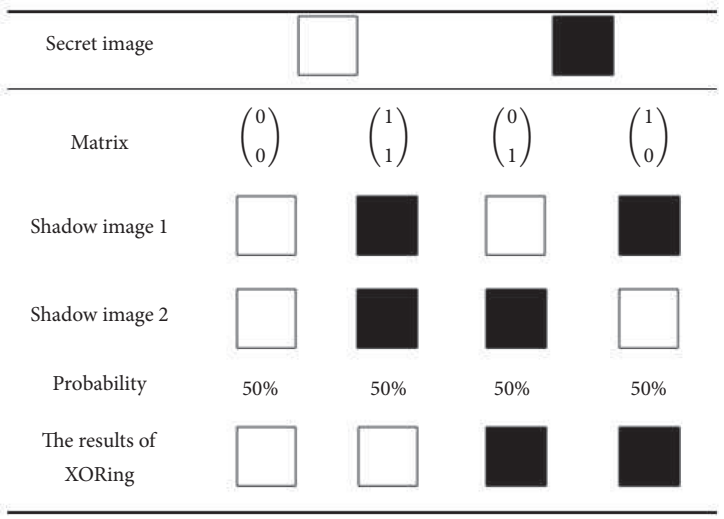


Figure 2: (2, 2) XOR-based VSS application example.

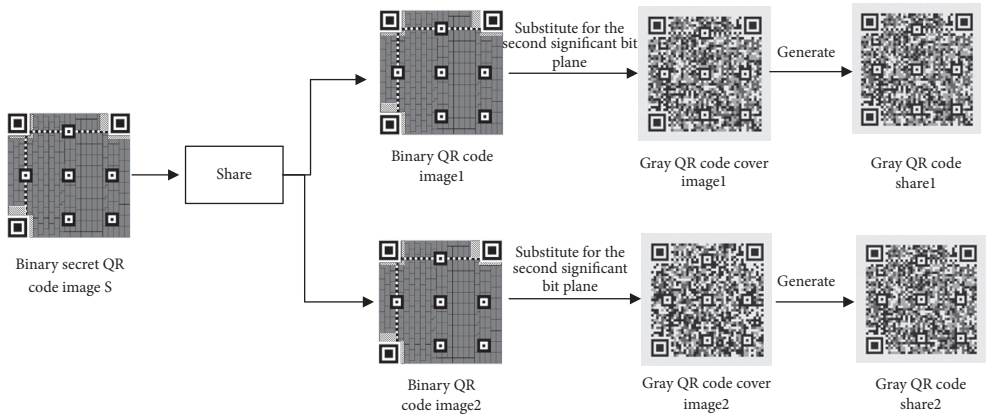


Figure 3: The idea of the grayscale QR code shares generation in our (2, 2) threshold scheme.

shares. The messages in grayscale cover images are different from each other, the grayscale QR code shares can be decoded by a standard QR code decoder.

The secret QR code image can be recovered by extracting the secondary significant bit planes of n gray QR code images to generate n binary QR code images firstly and then XORing white and dark modules in the codewords region of the n binary images, adding the functional patterns, version, and format information later. All the codewords can be recovered and the decoding result of the reconstructed secret image is identical with that of the secret image.

Since the version and level of error correction of all the QR codes are the same, codewords are at the same position and functional pattern are the same. Taking the (2, 2) threshold scheme as an example, Figure 3 illustrates the grayscale QR code shares generation of the proposed scheme.

3.2. The Sharing Phase. When sharing the secret QR code image into arbitrary binary QR code images, we only encrypt the codewords region while other parts are identical with the binary QR code images, then replace the second significant bit planes of the grayscale QR code cover images with the encrypted binary QR code images. All the QR code images have the same version and level of error correction. The corresponding algorithm steps are described in Algorithm 1.

In Algorithm 1, the codewords of the secret QR code are divided averagely and shared randomly into n different temporary binary QR code images. The artificial modification of shares is not perceptible. The shares can be decoded by a standard QR code.

3.3. The Recovery Phase. Base on XORing operation, the data of the recovery QR code image could be lossless. Suppose that n grayscale shares are provided, we can get the pixels in the second significant bit planes from shares to create n binary QR code images. Then we read the version and level of error correction by a QR decoder. We perform XOR operation on the encoding regions of n binary images, after that putting the bits in other regions based on the version and level of error correction. In this way, we can create a QR code image whose message is the same as that of the secret image. The recovery phase is described in Algorithm 2.

Input: A binary secret QR code image S; n different binary temporary QR code images $T_1, T_2, \cdots, T_n$; n different binary QR code cover images $C_1, C_2, \cdots, C_n$; the threshold parameters $(n, n)(n \geq 2)$; the error correction characteristics (c, k, r)
Output: n grayscale shares $SC_1, SC_2, \cdots, SC_n$
(1) Generate n grayscale shares $GC_1, GC_2, \cdots, GC_n$ by graying n binary QR code images $C_1, C_2, \cdots, C_n$.
(2) Averagely and randomly divide the serial of total c codewords into n subsets denoted as $A_1, A_2, \cdots, A_n$, and the number of each set is $n_1, n_2, \cdots, n_i, \cdots, n_n$, where $n_1 + n_2 + \cdots + n_i + \cdots + n_n = c$.
(3) Serial v of the codeword in the secret image S is denoted as cw_{sv}, and the serial v of the codeword in the T_i is denoted as cw_{iv}.
(4) **for** $i = 1$ to n **do**
(5) **for** each codeword v in A_i **do**
(6) **if** $cw_{sv} \neq cw_{1v} \bigoplus \cdots \bigoplus cw_{iv} \bigoplus \cdots \bigoplus cw_{nv}$ **then**
(7) Modify the codeword of binary QR code image T_i as $cw_{iv} =\sim cw_{iv}$;
(8) **end if**
(9) **end for**
(10) **end for**
(11) **for** $i = 1$ to n **do**
(12) Replace the second significant bit plane of grayscale QR code image GC_i with the modified QR code image T_i, and replace the last six significant bit planes with random binary numbers to generate grayscale share SC_i.
(13) **end for**
(14) Output n grayscale shares $SC_1, SC_2, \cdots, SC_n$.

Algorithm 1: Proposed robust VSS applying to QR codes.

Input: n shares $SC_1, SC_2, \cdots, SC_n$; threshold parameters $(n, n)(n \geq 2)$;
Output: the reconstructed secret QR code image S'.
(1) **for** $i = 1$ to n **do**
(2) Read the secondary plane of share SC_i to create binary QR image as S_i
(3) **end for**
(4) Read the version as v and level of error correction noted as l of image S_1 with QR code decoder.
(5) Based on v and l, we can get the total number of the codewords in S_1 as c according to the standard [16]. For each codeword in S_i note as cw_{ij}.
(6) **for** $j = 1$ to c **do**
(7) $cw_j = cw_{1j} \bigoplus \cdots \bigoplus \cdots \bigoplus cw_{nj}$, cw_j is the result of XORing operation.
(8) **end for**
(9) Create the bits in other regions based on the version and level of error correction, after that put them and the bits of codeword $cw_j (1 \leq j \leq c)$ together into a new image to reconstruct a QR code image S'.
(10) Output the image S'.

Algorithm 2: Secret image recovery of the proposed scheme.

3.4. Analyses. In this section, we present some theoretical analyses about the properties of our scheme. Firstly, the grayscale QR code shares generated by Algorithm 1 and the secret QR code image reconstructed by Algorithm 2 can be decoded by a QR decoder. Secondly, although the shares are damaged to some degree or have some noise, the secret QR image could be decoded. Thirdly, the threshold in our scheme threshold is $(n, n)(n \geq 2)$.

The grayscale shares are generated by replacing the secondary planes with the binary QR codes in which shared the secret image by Algorithm 1, when a QR code decoder reading the grayscale QR code, thresholding the grayscale image in the first place. The thresholding way in different QR code decoder may be not the same, but the simplest and common one is to set the number of 128 as the threshold, so when remaining the most significant digits of the pixel values in grayscale QR code cover images unchanged, the grayscale shares could be decoded.

If the 6 bits behind the secondary bit of each pixel value are changed in the range from 000000 to 111111, the value in the secondary bit will not be changed. So when the last 6 bits are kept in the range from 000000 to 111111 although some noise added, the second significant bit plane replaced with secret image will not be affected, and the secret image can still be rebuilt. Because QR codes have the ability to correct errors, even if the modules are damaged or dirty in the error correction capacity range, the secret QR image could be decoded.

The error correction characteristics are (c, k, r); when we share the codewords of secret QR image, the number of codewords c is divided into $n(n \geq 2)$ sets averagely and randomly. Modifying the codewords of temporary QR codes according to steps (4)-(10) in Algorithm 1 to generate n binary QR code images and then replace the second significant bit planes of the grayscale QR code cover images with the binary QR code images to generate n shares. Each share can be

(a) S

(b) C_1

(c) C_2

Parsed information 1:

Version: 7
Error level:H, Mask:0
Content:
Constant dropping wears the stone.

(d) The decoding information of S

Parsed information 1:

Color Normal, Direct
Version: 7
Error level:H, Mask:0
Content:
Be honest rather clever.

(e) The decoding information of C_1

Parsed information 1:

Color Normal, Direct
Version: 7
Error level:H, Mask:0
Content:
Practice makes perfect.

(f) The decoding information of C_2

(g) GC_1

(h) GC_2

(i) SC_1

Parsed information 1:

Color Normal, Direct
Version: 7
Error level:H, Mask:0
Content:
Be honest rather clever.

(j) The decoding information of GC_1

Parsed information 1:

Color Normal, Direct
Version: 7
Error level:H, Mask:0
Content:
Practice makes perfect.

(k) The decoding information of GC_2

Parsed information 1:

Color Normal, Direct
Version: 7
Error level:H, Mask:0
Content:
Be honest rather clever.

(l) The decoding information of SC_1

(m) SC_2

(n) S'

Parsed information 1:

Color Normal, Direct
Version: 7
Error level:H, Mask:0
Content:
Practice makes perfect.

(o) The decoding information of SC_2

Parsed information 1:

Color Normal, Direct
Version: 7
Error level:H, Mask:0
Content:
Constant dropping wears the stone.

(p) The decoding information of S'

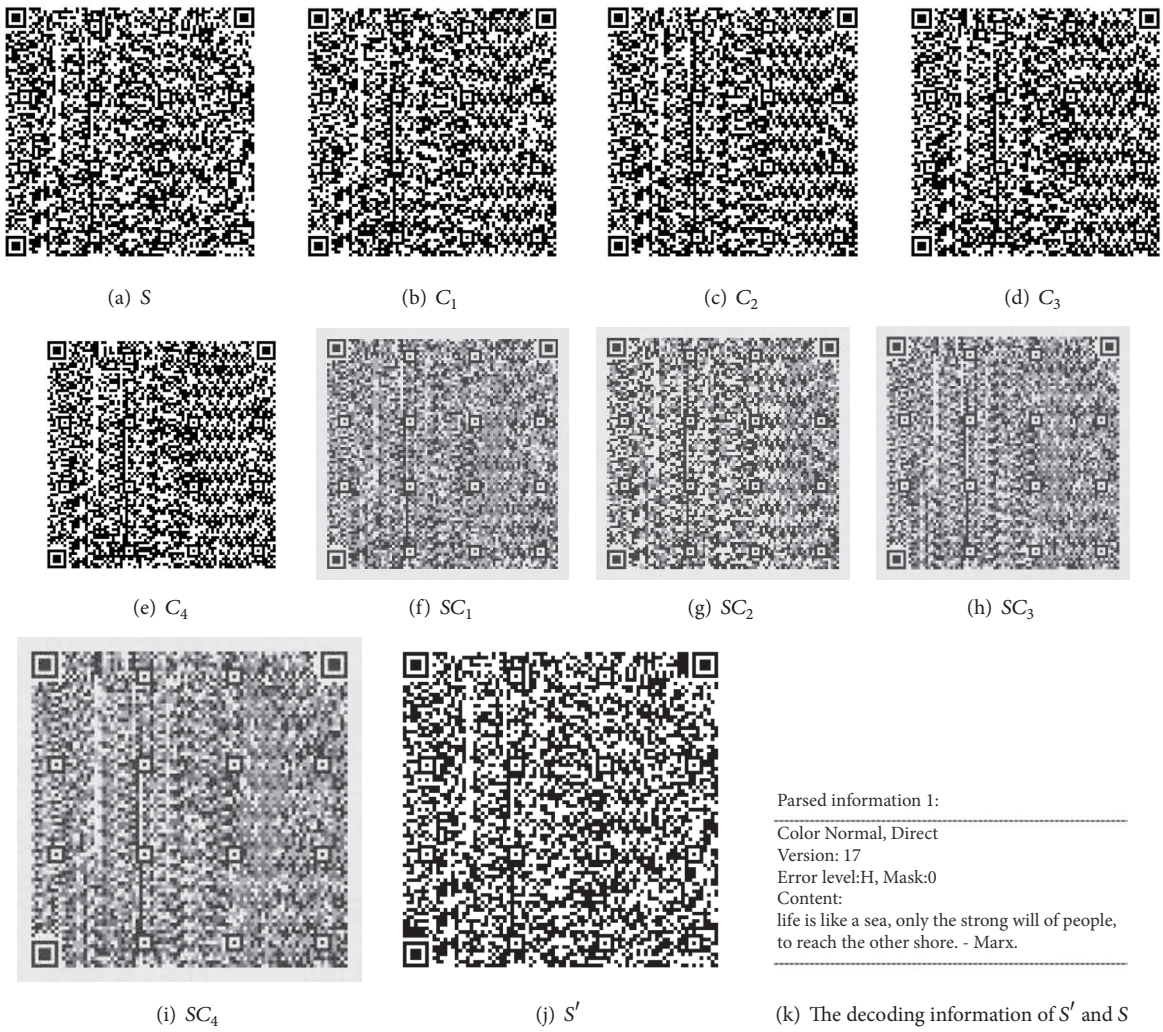
Figure 4: Our $(2, 2)$ threshold XOR-based VSS on QR code of version 7 with level of error correction H.

scanned by QR code decoder. When recovering the secret QR code image, the second significant bit planes are extracted from n shares to create n binary QR code images, performing XOR operation on the codewords regions of n binary QR code images, so that the codewords of the secret QR code image would be lossless. Put the bits in other regions to reconstruct the secret image. Since the codewords of the secret QR code image are lossless, the message of the reconstructed image decoded by a QR code decoder would be the same as that of secret image. So, we can achieve the threshold of (n, n).

4. Experiments and Comparisons

In this section, experiments and comparisons are taken into account to evaluate the effectiveness of the proposed scheme.

4.1. Image Illustration. In our experiments, the simulation environment of the proposed scheme is python language. Figure 4 is the $(2, 2)$ threshold XOR-based VSS on grayscale QR code of version 7 with level of error correction H. Figure 4(a) is the original secret image S and Figures 4(g) and 4(h) and GC_1 and GC_2 are grayscale QR code cover



(a) S (b) C_1 (c) C_2 (d) C_3

(e) C_4 (f) SC_1 (g) SC_2 (h) SC_3

(i) SC_4 (j) S' (k) The decoding information of S' and S

Figure 5: Our (4, 4) threshold XOR-based VSS on grayscale QR code of version 17 with level of error correction H.

images obtained by graying Figures 4(b) and 4(c) and C_1 and C_2, respectively. Figures 4(d), 4(e), 4(f), 4(j), and 4(k) are responding decoding results of Figures 4(a), 4(b), 4(c), 4(g), 4(h). Figures 4(i) and 4(m) are the grayscale QR code shares, SC_1 and SC_2. Figures 4(l) and 4(o) are the decoding results of them. Figure 4(n) is the reconstructed secret QR code image S', and Figure 4(p) is the decoding result of S'.

Figure 5 is our (4, 4) threshold XOR-based VSS on grayscale QR code of version 17 with level of error correction H. Figures 5(b)–5(e), C_1, C_2, C_3, and C_4 are input binary QR code cover images. Figures 5(f)–5(i), SC_1, SC_2, SC_3, and SC_4, are grayscale shares. Figure 5(j) is the reconstructed QR code image and Figure 5(k) is the decoding result of S and S'.

The reconstructed secret QR code image S' in Figures 4 and 5 are recovered by performing XOR operation on the codewords region of QR code image in the second significant bit planes of shares, then adding the functional patterns, version and format information together. The shares and the reconstructed secret image could be decoded by a QR code decoder, and the decoding result of the reconstructed secret image is identical with that of the secret QR code image.

4.2. Comparisons with Related Works. Figure 6 is our (3, 3) threshold XOR-based VSS on QR code with version 4 and level of error correction H. Figure 6(a) is the secret image S. Figures 6(b)–6(d) are input binary QR code cover images, C_1, C_2, and C_3. Figures 6(e)–6(g) are grayscale shares, SC_1, SC_2, and SC_3. Figure 6(h) is the reconstructed secret image S'. Figure 6(i) is the decoding result of Figures 6(a) and 6(h).

Figure 7 shows the VSS in [10]. Figure 7(a) is the secret image S. Figures 7(b)–7(d) are binary QR code cover images, C_1, C_2, and C_3. Figures 7(e)–7(g) are the decoding results of Figures 7(b)–7(d). Figures 7(h)–7(j) are binary QR shares, SC_1, SC_2, and SC_3. Figure 7(k) is the reconstruct secret image in QR code, S'.

The secret image in Figure 7 is encrypted from the coordinate of (7, 7) to right down corner of QR code image. The size of secret image is smaller than the QR code, and the secret image is shared in a continuous region of the cover images, threshold value is no less than 2. The QR code images are all binary images. The secret image in our scheme is shared dispersive, high image quality and high imperceptibility can be achieved. The secret image has the

(a) S (b) C_1 (c) C_2

(d) C_3 (e) SC_1 (f) SC_2

(g) SC_3 (h) S' (i) The decoding information of S' and S

Figure 6: Our (3, 3) threshold XOR-based VSS on QR code of version 4 with level of error correction H.

same size with the cover images, and the shares are grayscale QR code images.

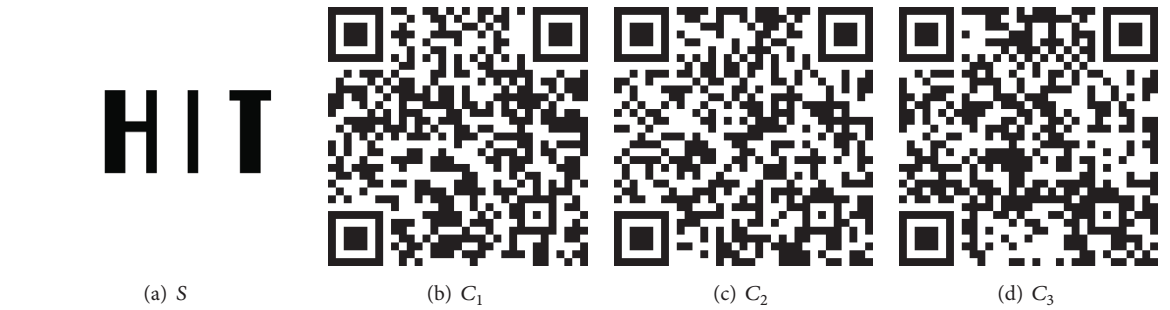
Figure 8 shows the VSS in [11]. Figure 8(a) is the secret image S. Figures 8(b)–8(d) are binary QR code cover images, C_1, C_2, C_3. Figures 8(e)–8(g) are binary QR shares, SC_1, SC_2, SC_3. Figure 8(h) is the reconstruct secret image in QR code S'. Figure 8(i) is the decoding result of Figures 8(a) and 8(i).

The shares in Figure 8 are binary QR code images. The secret QR code image could be reconstructed through XORing the white and black modules contained in the encoding region of the n QR code shares and then padding the bits in other regions. The threshold in paper [11] is not less than 3. In our scheme, the shares are grayscale QR code images and the secret image could be recovered by XORing the bits contained in the codewords regions of the second significant bit planes of n shares and then adding the function patterns, version, and format information. The threshold in our scheme is not less than 2.

Compared to the schemes in [10, 11], our scheme takes into account the advantage of threshold value greater than 2, high image quality, and high imperceptibility.

4.3. Robust Test. In the proposed technique, all the images are QR codes, which have their own abilities to correct errors, and the plane modification is the second significant bit plane of the grayscale image, in this way, the algorithm is robust to conventional image attacks. Taking our (2, 2) threshold scheme to resist Gaussian noise as an example, addition of different noises to the two shares to test the robustness of the scheme is shown by Figure 9.

Figures 9(a) and 9(b) are shares in Figure 4, SC_1, and SC_2. Figures 9(c) and 9(d) are shares added with Gaussian noise($mean = 0, variance = 1$), Figure 9(e) is the reconstructed secret image from Figures 9(c) and 9(d), S_1'. Figures 9(f) and 9(g) are shares added with Gaussian noise($mean = 0, variance = 8$), Figure 9(h) is the reconstructed secret image



(a) S (b) C_1 (c) C_2 (d) C_3

Parsed information 1:
Color Normal, Direct
Version: 4
Error level:H, Mask:0
Content:
QR Code Secret Sharing Example

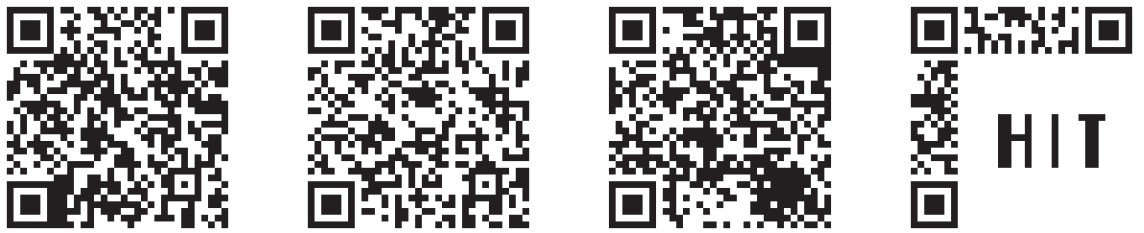
(e) The decoding information of C_1

Parsed information 1:
Color Normal, Direct
Version: 4
Error level:H, Mask:0
Content:
Secret Sharing test

(f) The decoding information of C_2

Parsed information 1:
Color Normal, Direct
Version: 4
Error level:H, Mask:0
Content:
Secret Sharing for QR Code

(g) The decoding information of C_3



(h) SC_1 (i) SC_2 (j) SC_3 (k) S'

Figure 7: (3, 3) threshold VSS on QR code of version 4 with level of error correction H In [10].

from Figures 9(f) and 9(g), S_2'. Figure 9(i) is the decoding result of S_1' and S_2'. Figure 9(j) displays the differences between S in Figure 4(a) and S_1', and the differences between S in Figure 4(a) and S_2' is showed in Figure 9(k). When the value of variance is in range from 0 to 8, the reconstructed QR code could be decoded, the greater the variance, the greater the difference. When the value of variance is larger than 9, the reconstructed QR code could not be decoded. The decoding result of S_1' and S_2' is the same as the decoding result of Figure 4(a), so that our scheme is robust to some noises.

Different attacks such as JPEG compression, rotation, resizing, and cropping are tested to further show the robustness. Whether the reconstructed secret QR code image can be decoded correctly when attacks are applied on shares is summarized in Table 2. We can see from Table 2 that we can resist some conventional attacks to certain degree, including JPEG compression, Gaussian noise, rotation, resizing, and cropping. These attacks may exist during shares transmission, so our scheme is very meaningful for practical application.

5. Conclusion

This paper proposed a novel robust VSS scheme applying to QR code. In this scheme, all the codewords of the secret QR code image are split into n temporary binary QR codes randomly with the theory of XOR-based VSS, so shares with high image quality and high imperceptibility can be achieved in the end. Each share in our scheme can be decoded by a QR code decoder when distributing via public channels, which would avoid the attentions from potential attackers. Since all the images are QR codes, which have their own abilities to correct errors, and the plane modification is the second significant bit plane of the grayscale image, our scheme is robust to conventional image attacks, such as rotation, JPEG compression, Gaussian noise, resizing and cropping, when reconstructing the secret image, performing XOR operation on the bits of the codewords region in the second significant planes of the grayscale shares and adding the functional patterns, version, and format information together. There are no wrong codewords in the reconstructed secret QR code image, so the message of which is identical with that of the original secret image. The threshold of our scheme is $(n, n)(n \geq 2)$. The reduction of the size of the shares will be the future work.

Data Availability

No data were used to support this study.

Table 2: Performance of proposed technique against various attacks.

Type of Attack	Parameter	Whether S′ can be decoded correctly (yes/no)?	Type of Attack	Parameter	Whether S′ can be decoded correctly (yes/no)?
JPEG Compression	Q = 90	yes	Rotation	15°	yes
	Q =85	yes		30°	yes
	Q = 8	yes		45°	yes
	Q = 75	yes		60°	yes
	Q = 70	no		75°	yes
Gaussian Noise	Mean = 0, Variance =1	yes	Resizing	0.8	yes
	Mean = 0, Variance =2	yes		1.1	yes
	Mean = 0, Variance =3	yes		1.4	yes
	Mean = 0, Variance =4	yes		1.7	yes
	Mean = 0, Variance =5	yes		2.0	yes
Gaussian Noise	Mean = 0, Variance =6	yes	Cropping	1/100	yes
	Mean = 0, Variance =7	yes		1/25	yes
	Mean = 0, Variance =8	yes		9/100	yes
	Mean = 0, Variance =9	**no**		4/25	yes
				1/4	yes
				9/25	**no**

(a) S (b) C_1 (c) C_2

(d) C_3 (e) SC_1 (f) SC_2

(g) SC_3 (h) S' (i) The decoding information of S' and S

Figure 8: (3, 3) threshold VSS on QR code of version 4 with level of error correction H In [11].

(a) SC_1 (b) SC_2 (c) SC_1 with Gaussian noise (0,1) (d) SC_2 with Gaussian noise (0, 1)

(e) S_1' reconstructed from shares with Gaussian noise (0, 1) (f) SC_1 with Gaussian noise (0, 8) (g) SC_2 with Gaussian noise (0, 8) (h) S_2' reconstructed from shares with Gaussian noise (0,8)

(i) The decoding information of S_1' and S_2' (j) The difference between Figure 4(a) and S_1' (k) The difference between Figure 4(a) and S_2'

Figure 9: The performance of our shares with Gaussian noises.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 61602491) and the Key Program of the National University of Defense Technology (Grant no. ZK-17-02-07).

References

[1] M. Naor and A. Shamir, "Visual cryptography," in *Advances in Cryptology—EUROCRYPT'94*, vol. 950 of *Lecture Notes in Computer Science*, pp. 1–12, Springer, Berlin, Germany, 1995.

[2] C.-N. Yang and D.-S. Wang, "Property analysis of XOR-based visual cryptography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 2, pp. 189–197, 2014.

[3] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[4] G. R. Blakley, "Safeguarding cryptographic keys," in *AFIPS*, 1979.

[5] X. Yan, S. Wang, and X. Niu, "Threshold construction from specific cases in visual cryptography without the pixel expansion," *Signal Processing*, vol. 105, pp. 389–398, 2014.

[6] X. Yan, X. Liu, and C.-N. Yang, "An enhanced threshold visual secret sharing based on random grids," *Journal of Real-Time Image Processing*, vol. 14, no. 1, pp. 61–73, 2018.

[7] X. Yan and Y. Lu, "Participants increasing for threshold random grids-based visual secret sharing," *Journal of Real-Time Image Processing*, vol. 14, no. 1, pp. 13–24, 2018.

[8] X. Wu and W. Sun, "Random grid-based visual secret sharing with abilities of or and XOR decryptions," *Journal of Visual Communication and Image Representation*, vol. 24, no. 1, pp. 48–62, 2013.

[9] P. Tuyls, H. D. Hollmann, J. H. van Lint, and L. Tolhuizen, "Xor-based visual cryptography schemes," *Designs, Codes and Cryptography. An International Journal*, vol. 37, no. 1, pp. 169–186, 2005.

[10] S. Wan, Y. Lu, X. Yan, and L. Liu, "Visual Secret Sharing Scheme with (*k*, *n*) Threshold Based on QR Codes," in *Proceedings of the International Conference on Mobile Ad-Hoc and Sensor Networks, MSN '17*, 2017.

[11] Y. W. Chow, W. Susilo, G. Yang, J. G. Phillips, I. Pranata, and A. M. Barmawi, "Exploiting the error correction mechanism in QR codes for secret sharing," in *Proceedings of the Australasian Conference on Information Security and Privacy*, 2016.

[12] J. Weir and W. Yan, "Authenticating Visual Cryptography Shares Using 2D Barcodes," in *Proceedings of the 10th International Workshop on Digital Forensics and Watermarking, IWDW '11*, pp. 196–210, Springer, Atlantic City, NY, USA, 2011.

[13] G. Wang, F. Liu, and W. Q. Yan, *Barcodes for Visual Cryptography*, Kluwer Academic Publishers, 2016.

[14] Y. Cheng, Z. Fu, and B. Yu, "Improved Visual Secret Sharing Scheme for QR Code Applications," *IEEE Transactions on Information Forensics and Security*, vol. 99, p. 1, 2018.

[15] Y. W. Chow, W. Susilo, J. Tonien, E. Vlahu-Gjorgievska, and G. Yang, "Cooperative Secret Sharing Using QR Codes and Symmetric Keys," *Symmetry*, vol. 10, no. 4, p. 95, 2018.

[16] Information technology-automatic identification and data capture techniques-bar code symbology-qr code, Iso/iec.

[17] D. Samretwit and T. Wakahara, "Measurement of reading characteristics of multiplexed image in QR code," in *Proceedings of the 3rd IEEE International Conference on Intelligent Networking and Collaborative Systems, INCoS '11*, 2011.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 7350324, 14 pages
https://doi.org/10.1155/2018/7350324

Research Article

DR-Net: A Novel Generative Adversarial Network for Single Image Deraining

Chen Li,[1,2] Yecai Guo,[1,2] Qi Liu,[3] and Xiaodong Liu[3]

[1]*School of Electronic and Information Engineering, Nanjing University of Information Science & Technology, Nanjing 210023, China*
[2]*Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing 210023, China*
[3]*School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, UK*

Correspondence should be addressed to Yecai Guo; guo-yecai@163.com

Received 31 August 2018; Revised 8 November 2018; Accepted 13 November 2018; Published 3 December 2018

Guest Editor: Weizhi Meng

Blurred vision images caused by rainy weather can negatively influence the performance of outdoor vision systems. Therefore, it is necessary to remove rain streaks from single image. In this work, a multiscale generative adversarial network- (GAN-) based model is presented, called DR-Net, for single image deraining. The proposed architecture includes two subnetworks, *i.e.,* generator subnetwork and discriminator subnetwork. We introduce a multiscale generator subnetwork which contains two convolution branches with different kernel sizes, where the smaller one captures the local rain drops information, and the larger one pays close attention to the spatial information. The discriminator subnetwork acts as a supervision signal to promote the generator subnetwork to generate more quality derained image. It is demonstrated that the proposed method yields in relatively higher performance in comparison to other state-of-the-art deraining models in terms of derained image quality and computing efficiency.

1. Introduction

In recent years, with the development of network communication, the intelligent monitoring system based on image and video processing technology has achieved promising progress. Such system plays a vital role in the maintenance of public security. Therefore, some computer vision issues related to the intelligent monitoring system have attracted a wide spread attention. Most of the computer vision algorithms proposed for addressing these issues can work well on the high visibility of video or image data. However, when the algorithms face degraded data, their performances may obviously degrade. This is because the training processes of these algorithms are based on the high visibility of the video or image dataset. Bad weather such as rainy days seriously degrades the visual quality of the captured videos or images, which may affect the performance of many computer vision algorithms like tracking, recognition, and retrieval [1]. This kind of circumstance may happen on some safely related accidents recorded by mobile phone or monitoring camera in rainy days. In such circumstance, the captured video or image data may contain a large number of fast-moving rain streaks, which leads to the image signal distortion as well as reducing the signal-to-noise ratio and image quality. These impacts caused by rainy weather on video or image data have brought great difficulties for intelligent traffic, outdoor monitoring, military reconnaissance, and so on [2, 3]. In order to enhance the reliability of outdoor computer vision systems, there is a need to exploit effective algorithms to remove rain streaks from the degraded single image caused by rainy weather.

Mathematically, the process of the deraining procedure can be written as

$$I - O = R, \tag{1}$$

where I represents the rainy image and O and R represent rain streaks and restored clean image, respectively.

In the past few decades, some algorithms have been proposed to jointly address the rain detection and removal task. According to their concerns, these methods can be roughly classified into two categories, *i.e.,* video-based methods and single-image-based methods. Video-based deraining methods concentrate on eliminating the rain streaks from video sequence [4, 5] through exploiting frequency properties and

temporal information of rain streaks. Single-image-based methods consider the problem from two aspects:

(1) A task of blind image single decomposition: These methods mainly include morphological component analysis with sparse coding [2, 3], generalized low rank model [6], structural similarity constraints [7], and nonlocal means smooth [8].

(2) A task of learning an end-to-end projection between the rainy image and its corresponding derained clean image: Recently, due to their relatively superior ability of learning nonlinear functions, some deep learning-based methods have been proposed to address this issue, in which an end-to-end projection between the rain image and its corresponding ground truth is directly learned. These methods contain convolution neural network- (CNN-) based models [9, 10] and generative adversarial network-based models (GAN) [11, 12].

Although existing methods have achieved some degree of success, there still exist several limitations caused by the following aspects: (1) For the basic operations of many existing approaches are performed on a small receptive field or a local image patch, the spatial contextual information between the patches or receptive fields is usually ignored. (2) Since the background texture patterns and rain streaks are internally overlapping, texture details in nonrain regions are removed by most approaches, which results in the restored images containing some oversmoothness regions. (3) Some of these models introduce additional image enhancement techniques to improve the visual effect of image, which reduces the efficiency of the algorithm.

In order to alleviate these limitations mentioned above, our goal is to exploit a novel architecture with the ability of removing rain and keeping the restored clean image details jointly. More specifically, we introduce a multiscale GAN-based architecture called single image deraining generative adversarial network (DR-Net) to handle the single image deraining issue. The architecture includes two subnetworks, *i.e.,* generator network and discriminator network. Generator network acts as a feature exactor that can eliminate the rain streaks while encoding the image contents. In other words, it learns a nonlinear projection function which transfers a rainy image into a restored clean image and keeps the details of raw image simultaneously. To capture more spatial information and local rain drops, we propose a multiscale parallel convolution generator network which consists of two-branch convolution operator with different kernel sizes. Discriminator network uses the restored image generated by the generator architecture and ground truth image as inputs. It aims at differentiating the restored clean image from the real ground truth image. The function of discriminator network is to boost the generator architecture to generate more quality derained image that closely resembles the ground truth.

To sum up, the contributions of this work are as follows:

(1) We design a novel generative adversarial architecture to address the single image deraining issue. The generator architecture consists of two parallel convolution subnetworks that have different kernel sizes, specifically, one subnetwork with large kernel size that captures more spatial information of the raw image, and the other subnetwork with relatively small kernel size which aims to acquire more local rain streaks knowledge. The multiscale operators are helpful for keeping the details of raw image and eliminating rain streaks at the same time. Additionally, with fewer feature maps, our network has the advantages of less parameters and less computing effort; thus, the training convergence and test speed are relatively faster among the comparison methods.

(2) Experiments on publicly synthesized dataset and real images show the effectiveness of the proposed network. The proposed model performs better than other recent state-of-the-art single image deraining techniques.

The remainder of this paper is organized as follows. A brief review on existing methods for image deraining is given in Section 2. Section 3 provides the detail of the proposed DR-Net architecture. Section 4 presents the experiment results on both synthetic and real images. Finally, a brief discussion is concluded in Section 5.

2. Related Works

In the past few years, a number of models have been presented to enhance the visibility of images captured with rain streaks. These models can be divided into two categories: video sequences-based methods and single-image-based methods. In this section, we give a brief review of these image deraining models.

(A) Video Sequences-Based Methods. Video sequences-based rainy image recovery has been extensively studied. Garg et al. [1] proposed a deraining model for the rain streak detection and removal from video sequences. The rain streaks detection has two constraints [13]: (1) First, since the rain streaks are dynamic, the changes in intensity inside their several frames are comparatively high. (2) Second, because other objects may also be dynamic, through examining whether the relations between the intensity changes along the streak and background intensity are photometrically linear, the rain streaks can be differenced. The second constraint can reduce the error alarms caused by the first. After detecting the rain streaks, the average intensity of the pixels taken from the previous and subsequent frames is used to remove the streaks. Soon after this, they further developed a postprocessing architecture for video sequences-based deraining [14]. Specifically, they first proposed a photometric method which can describe the intensities generated by individual rain. Then, a dynamic model that can capture spatiotemporal attributes of rain streaks was presented. Finally, they used these models together to describe the visual appearance of rain streaks. Zhang et al. [15] introduced another constraint named chromaticity constraint. They point out that the intensity changes in the R, G, and B channels are alike for representing rain streaks. Based on the size information and photometry properties of rain streaks, Bossu et al. [16] proposed a rain detection algorithm to fit a Gaussian distribution on rain streak histograms. They adopted a Gaussian mixture model to separate the foreground used to detect the rain streaks from the background in video sequences.

(B) Single-Image-Based Methods. Since there are no temporal information for rain streaks detection and removing, compared with the video sequences-based removal issue, single-image rain removal (SIRR) is more challenging. Some researchers regard the SIRR problem as a task of layer separation. Kang et al. [3] used a bilateral filter to decompose the rainy image into high-frequency and low-frequency parts. Then, they utilized sparse coding and dictionary learning to separate the rain component from the high-frequency part. Through analyzing the aspect ratio of elliptical kernel and the orientation angle in each pixel location, Kim et al. [8] first detected the rain streaks regions. Then, they used adaptive nonlocal means filter to these rain streak regions to remove the rain streaks. Luo et al. [17] proposed a nonlinear screen blend model to model the rainy images. Specifically, through learning a dictionary with reciprocal exclusivity, they used sparse coding to separate the rain layer and derained layer. Since the rain streaks on the imaging scene usually appear recursively with similar patterns, Chen et al. [6] proposed and generalized a low-rank rain appearance architecture to capture the spatiotemporal relationship between rain streaks. Li et al. [13] proposed a model that uses patch-based priors which are based on Gaussian mixture model for rain and background layers. Moreover, these priors could accommodate multiple scales and orientations of the rain streaks to certain degree.

Recently, deep learning-based methods have achieved outstanding performances in many domains [18–21], including image deblurring [22], image denoising [23], superresolution [24], style transfer [25, 26], and inpainting [27]. There also exists some literature that adopts the deep learning system to address the SIRR issue. These deep learning-based methods aim to learn a nonlinear mapping between the rainy image and corresponding deraining image. These methods can be mainly divided into CNN-based and GAN-based models. Fu et al. [9] designed a convolution neural network named DrainNet for removing rain streaks from single image. They first decomposed the input rainy image into its detail layer and base layer, in which the base layer keeps the structure and detail layer contains object details and rain streaks. Then, they used the detail layer as the input of deep architecture to detect and remove rain streaks. Finally, they added the output of the deep architecture to the base layer to obtain the final output. Yang et al. [28] proposed a multitask deep convolution neural network which can simultaneously learn the appearance of rain streaks, the binary rain streak, and the background. Besides, they developed a recurrent rain detecting and removing network to clear up the rain accumulation and remove rain streaks iteratively. As a popular deep learning technology, GAN [29, 30] has been adopted in many computer vision tasks. Most recently, Zhang et al. [11] proposed a conditional generative adversarial model for SIRR, in which a new refined loss function that combines perceptual loss, Euclidean loss, and adversarial loss is presented. In this paper, we also use a GAN-based method to address the SIRR problem. The architecture of the proposed generator subnetwork is different from the models mentioned above. More specifically, we design a generator with multiscale convolution operators that can simultaneously focus on the local rain drops and the spatial information of the rainy image.

3. Proposed Method

Our purpose is to learn a nonlinear projection between the input rainy image and the output derained image through constructing a GAN-based deep architecture. The proposed GAN-based network consists of two subnetworks, i.e., generator network and discriminator network. The primary target of generator subnetwork is to remove rain streaks without missing any detail message from the rainy image. The discriminator subnetwork acts as a supervised signal to boost the quality of derained image generated by generator subnetwork. In this section, we discuss the architecture in detail.

3.1. Generative Adversarial Loss. To make the derained image generated by generator subnetwork with high quality, to fool the discriminator subnetwork, and to learn, a good discriminator has enough ability to validate whether the derained image looked real. Given a rainy image I, the optimization function of the GAN can be formulated as

$$\min_G \max_D E_{I\sim P_{data(I)}} \left[\log\left(1 - D\left(I, G\left(I\right)\right)\right)\right] + E_{I\sim P_{data(I,O)}} \left[\log D\left(I, O\right)\right], \tag{2}$$

where O is the generated output image, D represents the discriminative subnetwork, and G is the generative subnetwork.

3.2. Generator Network

Architecture. As mentioned above, generator subnetwork aims to learn a mapping function from a rainy image to a derained image. The proposed generator framework is shown in the top of Figure 1. Specifically, we first implement convolution operator on the raw image with the kernel size and feature maps of 7×7 and 64, respectively. After the first convolution layer, we introduce two parallel convolution branches. The kernel size of one convolution branch is set to 3×3, and the other is set to 5×5. Both numbers of feature maps of each convolution layer for the two branches are set to 64. The generator subnetwork with multiscale branches has two advantages. Firstly, the small kernel size can capture more local rain information while the larger kernel size acquires more spatial information. Secondly, the multiscale convolution kernel is used to make the generator have a variety of filters, and then the learning process of weights and biases is more diverse, and thus the useful information of the image can be fully and effectively extracted. Both of the two settings are helpful for generating higher quality derained image. After five convolution layers for each branch, the outputs of the last convolution layer are concentrated by a simple addition operator. Besides, three skip connections between the front convolution layers and the several later convolution layers are introduced. As shown above, our network contains several convolution operations, which may seriously damage the details of the raw image. However,

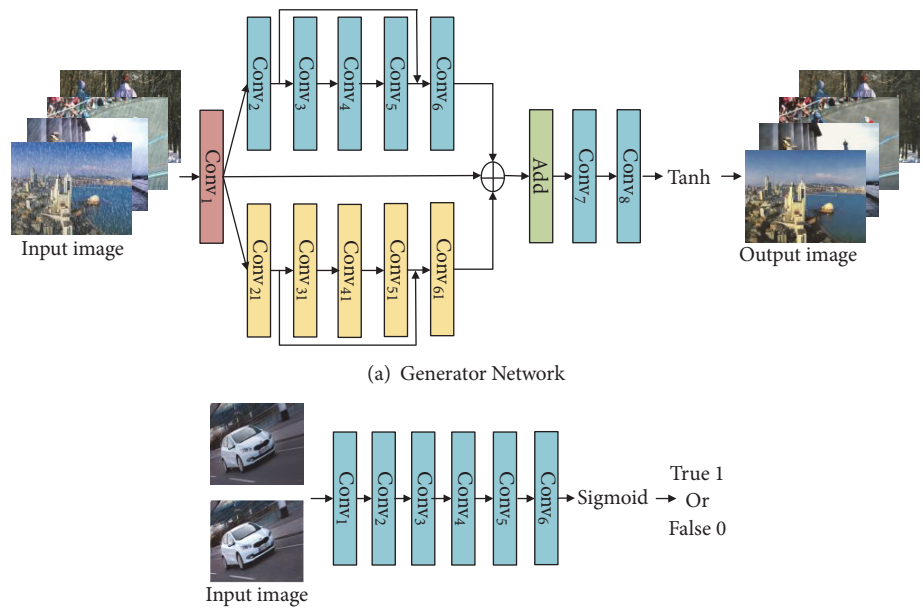


FIGURE 1: The architecture of the proposed DR-Net network. It takes a set of synthesized rainy images as inputs. First, these images are passed through a convolution layer. Then, two parallel convolution branches with five convolution layers are appended. After an additional layer and two convolution layers, the generator subnetwork outputs the derained image. Finally, the discriminator subnetwork uses the mixed derained images and the ground truths as inputs to distinguish if they are fakes or trues. Additionally, three skip connections between the front convolution layers and later convolution layers in generator subnetwork are added.

the feature maps generated by the front convolution layers contain many image details, and integrating these feature maps into the later convolution layers can help the generator to retain the image details. In addition, similar to the deep residual networks [31], the introduction of skip connections is conductive to backpropagate the gradient to the bottom layers, which makes the training phase more stable. Then, the concentrated branches are passed through two convolution layers both with 32 feature maps and 3×3 kernel size. Finally, the output layer is stacked after these convolution operators. In order to maintain the size of the raw image after the convolution operators, we set the padding to 3 pixels for the 7×7 conv layer, 1 pixel for 3×3 conv layers, and 2 pixels for 5×5 conv layers. The detailed parameters of the generative subnetwork layers are shown in Table 1.

Generative Loss Function. As the generator subnetwork aims to generate the derained image as closer as possible to the ground truth, therefore, we adopt the Euclidean loss to supervise the generator. Given a rainy image I, the loss function can be defined as

$$L_e(G) = \frac{1}{CMN}\sum_{c=1}^{C}\sum_{x=1}^{M}\sum_{y=1}^{N}\|G(I) - (R)\|_2^2, \tag{3}$$

where C, M, and N represent the channel, width, and height of the images, respectively. R is the corresponding ground truth of the input image I.

Aside from the Euclidean loss, we also introduce the perceptual loss [32] which can calculate the global difference between the features of the ground truth and those of the outputs of certain layer. The introduction of perceptual loss is helpful to improve the visual performance of generator subnetwork. The perceptual loss can be written as

$$L_p(G) = \frac{1}{SWH}\sum_{s=1}^{S}\sum_{w=1}^{W}\sum_{h=1}^{H}\left\|VGG_{16}(G(I)) - VGG_{16}(R)\right\|_2^2, \tag{4}$$

where S, W, and H represent the channel, width, and height of the output of a certain convolution layer, respectively. VGG_{16} is the VGG-16 model [33]. Following the work of [34], we use VGG-16 model to compute the feature loss at the additional layer.

Based on the two formulations above, we refined the generative loss function as

$$L_G = L_e(G) + \lambda L_p(G) \tag{5}$$

in which λ is the predefined weights for the perceptual loss.

3.3. Discriminator Architecture

Architecture. In our GAN-based network, discriminator architecture is designed for making the derained image synthesized by generator subnetwork much closer to the ground truth. It uses the mixed derained images and ground truths as inputs and classifies if the input is fake or real. Following the work of [11, 33], the convolution operator with

Table 1: Layer parameters of the generative sub-network.

Layer Name	Parameters
Input	Rain image
Layer 1	Conv1. (7,7,64), stride=1, padding=3; Tanh
Layer 2	Conv2. (3,3,64), stride=1, padding=1; Tanh
Layer 3	Conv3. (3,3,64), stride=1, padding=1; Tanh
Layer 4	Conv4. (3,3,64), stride=1, padding=1; Tanh
Layer 5	Conv5. (3,3,64), stride=1, padding=1; Tanh
Add layer	Concatenate (layer 1, layer 5)
Layer 6	Conv6. (3,3,64), stride=1, padding=1; Tanh
Layer 21	Conv21. (5,5,64), stride=1, padding=2; Tanh
Layer 31	Conv31. (5,5,64), stride=1, padding=2; Tanh
Layer 41	Conv41. (5,5,64), stride=1, padding=2; Tanh
Layer 51	Conv51. (5,5,64), stride=1, padding=2; Tanh
Add layer	Concatenate (layer 21, layer 51)
Layer 61	Conv61. (5,5,64), stride=1, padding=2; Tanh
Add layer	Concatenate (layer 1, layer 6, layer 61)
Layer 7	Conv7. (3,3,32), stride=1, padding=1; Tanh
Layer 8	Conv8. (3,3,32), stride=1, padding=1; Tanh
Output	Rain-removal image

PReLU activation and batch normalization is used as a basic unit throughout the whole discriminator subnetwork. The subnetwork contains five convolution layers. The numbers of feature map for each layer are set to 24, 48, 96, 192, and 384, respectively. We set the kernel size of the five convolution layers to 3×3. After a set of convolution layers, a sigmoid function is attached at the output layer to produce a probability value that indicates the input image as fake or real. The proposed discriminator subnetwork is shown as in the bottom of Figure 1. And the detailed parameters of the discriminator subnetwork layers are shown in Table 2.

Discriminator Loss Function. Since the goal of discriminator subnetwork is to differentiate the synthesized derained image from its corresponding ground truth, we regard it as a binary classification network. Given a mixed N images set, the discriminator loss function can be expressed as

$$L_D = -\frac{1}{N}\sum_{i=1}^{N}\left(T_i \log\left(D\left(I\right)\right) - \left(1 - T_i\right)\log\left(1 - D\left(I\right)\right)\right), \quad (6)$$

Table 2: Layer parameters of the discriminator sub-network.

Layer Name	Parameters
Input	Rain image
Layer 1	Conv1. (3,3,24), stride=2, padding=1;
Layer 2	Conv2. (3, 3, 48), stride=2, padding=1; PReLU
Layer 3	Conv3. (3, 3, 96), stride=2, padding=1; PReLU
Layer 4	Conv4. (3, 3, 192), stride=2, padding=1; PReLU
Layer 5	Conv5. (3, 3, 384), stride=2, padding=1; PReLU
Layer 6	Conv6. (3,3,1), stride=2, padding=1; Sigmoid
Output	1 or 0

where T_i is the label of input image I and T_i=1 indicates that I is a real ground truth while T_i=0 indicates that I is a fake.

4. Experiment and Results

In this section, we first introduce the dataset and evaluation protocols used in this work. Then, the details of the experiments implemented to evaluate the proposed DR-Net model are presented. Finally, some comparison experiments results are discussed.

4.1. Dataset and Evaluation Protocols

Synthetic Dataset. We use the synthesize dataset created by [11] as the training and testing data. The training set of this dataset contains 700 paired images, in which 200 images are selected from BSD training set [35] and the rest of the 500 images are chosen from the UCID dataset [36]. The test set contains a total of 100 paired images, in which 50 images were selected from BSD dataset and the rest of the 50 were chosen from UCID dataset. Besides, we also use the test set created by [9] to evaluate the proposed model.

Real-World Rainy Image. To validate the effectiveness of the proposed DR-Net, we also test it on the real-world rainy image. Specifically, we use the real-world dataset created by [11] and some traffic images downloaded from the Internet to evaluate the performance of our model.

Evaluation Protocols. We adopt the structural similarity index (SSIM) [37] and the visual information fidelity (VIF) [38] to evaluate the performance of our model and the compared state-of-the-art methods as well. The higher SSIM value is, the closer to ground truth the derained image is. For the clean image, the SSIM value is 1. Similarly, higher VIF indicates higher quality of the derained result.

4.2. Training Setting. In this study, we use the torch framework [39] to implement our network. The batch size is

Table 3: SSIM results compared with three baseline networks and one extended network on synthesized test images.

Images	3×3 Single	5×5 Single	GEN	DGAN	**Ours**
a	0.6461	0.7452	0.7737	**0.8592**	0.8571
b	0.8213	0.7915	07866	0.8476	**0.8525**
c	0.6345	0.6647	0.6466	0.8859	**0.8826**
d	0.8007	0.7586	0.7880	0.8112	**0.8176**
e	0.6331	0.6469	0.7919	0.9196	**0.9287**

Table 4: SSIM results on synthesized test image.

Images	Ground truth	Rainy image	DSC [17]	GMM-LP [13]	DrainNet [9]	ID-CGAN [11]	Ours
f	1	0.7270	0.7550	0.8067	0.8448	0.8465	**0.8604**
g	1	0.8569	0.8836	0.8865	0.8995	0.8573	**0.9186**
h	1	0.6857	0.6673	0.7663	0.8042	0.7720	**0.8607**
i	1	0.7707	0.7837	0.8532	0.8620	0.8762	**0.9101**
j	1	0.7093	0.7687	0.7637	0.8009	0.8071	**0.8064**

set to 9. We use the Adaptive Moment Estimation (Adam) algorithm with the learning rate of 0.0002 to optimize the network. All the training images are resized to 480×480 pixels. The training process converges in roughly 4-5 hours with NVIDIA GTX Taitan-xp GPU.

4.3. Comparison with Baseline Networks. In this section, we evaluate the performance of the proposed DR-Net with the following four baseline architectures:

(i) 3×3 Single: Single-scale network is trained only using 3×3 convolution branches.

(ii) 5×5 Single: Single-scale network is trained only using 5×5 convolution branches.

(iii) GEN: Only generator subnetwork is used, which equates to a traditional CNN architecture.

(iv) DGAN: The depth and kernel's numbers of network are increased.

We train these four networks as well as DR-Net on the synthetic training dataset. Table 3 shows the SSIM results of DR-Net compared with the four baseline architectures on synthesized test images. From Table 3, we can observe that the proposed DR-Net achieves the highest SSIM values among the five configurations. Compared with the DR-Net, the generator with single-scale network (i.e., 3×3 or 5×5 convolution branches) achieves the lower SSIM values. When we discard the discriminator subnetwork, the performance of the architecture is also decreased. The DGAN achieves comparable SSIM values with DR-Net. However, increasing the depth and the kernel's number leads to more running time whether in the training or testing phases. Sample results of the proposed method compared with the four networks on synthesized test images are shown in Figure 2. From Figure 2, it can be seen that the four baseline networks can improve the quality of the rain image but have obvious chromatic aberration and blurred background. The proposed DR-Net achieves the best visual effect in terms of the quality of derained image.

4.4. Comparison with State-of-the-Art Methods. We compare the proposed method with the following representative single-image deraining methods:

DSC: discriminative sparse coding-based method.

GMM-LP: layer prior-based model.

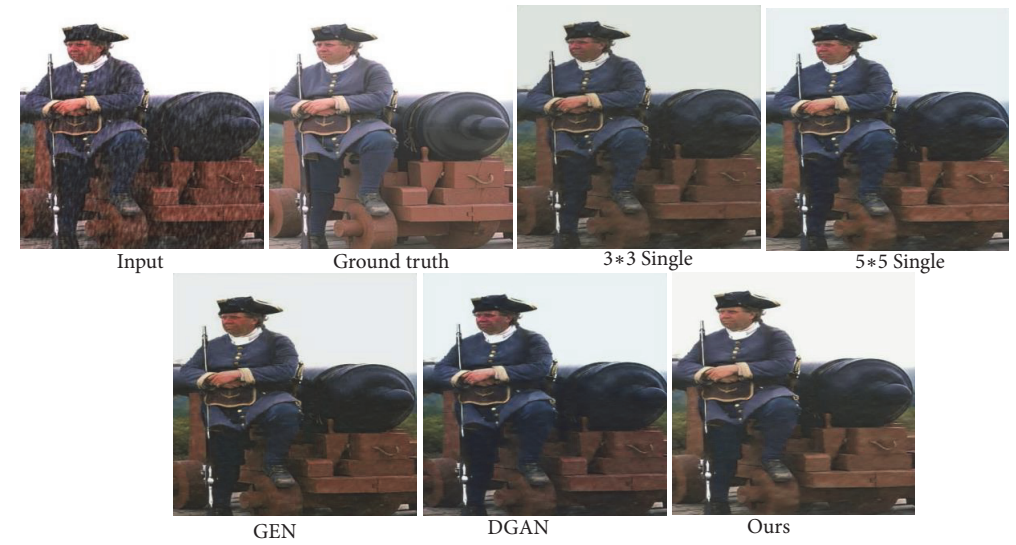
DrainNet: convolution neural network-based method.

ID-CGAN: conditional general adversarial network-based method.

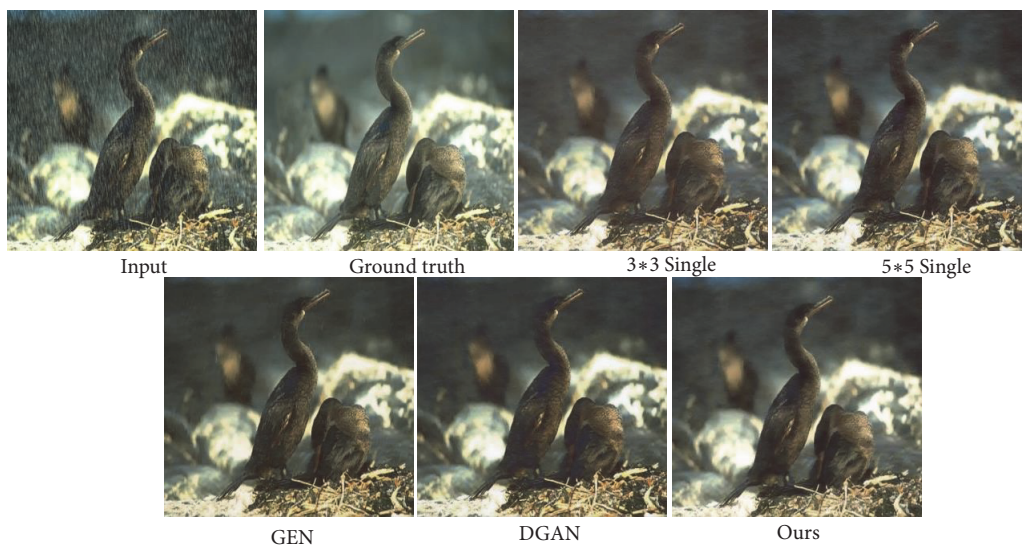
Results on Synthesized Data. In this set of experiments, we implement the comparisons between the proposed model and the four compared methods on the newly synthesized image data. For the ground truths of these test images are known, the structure similarity index (SSIM) and the visual information fidelity (VIF) for quantitative measure can be calculated. From Tables 4 and 5, we can observe that the proposed model achieves the highest SSIM and VIF values.

The visual comparisons for five synthesized images with different intensity and orientations are shown as Figure 3. As can be seen, DSC can remove the partial streaks and reduce the dense degree of rain streaks, but they cannot completely remove the rain streaks. The same situation also takes place in GMM-LP algorithm. Among these four compared algorithms, DrainNet and ID-CGAN get better visual expression. However, compared to the two models, the derained results of the proposed model are better. Moreover, our SSIM and VIF values of five derained images are higher than theirs. Both the experiment results demonstrate the effectiveness of the proposed method.

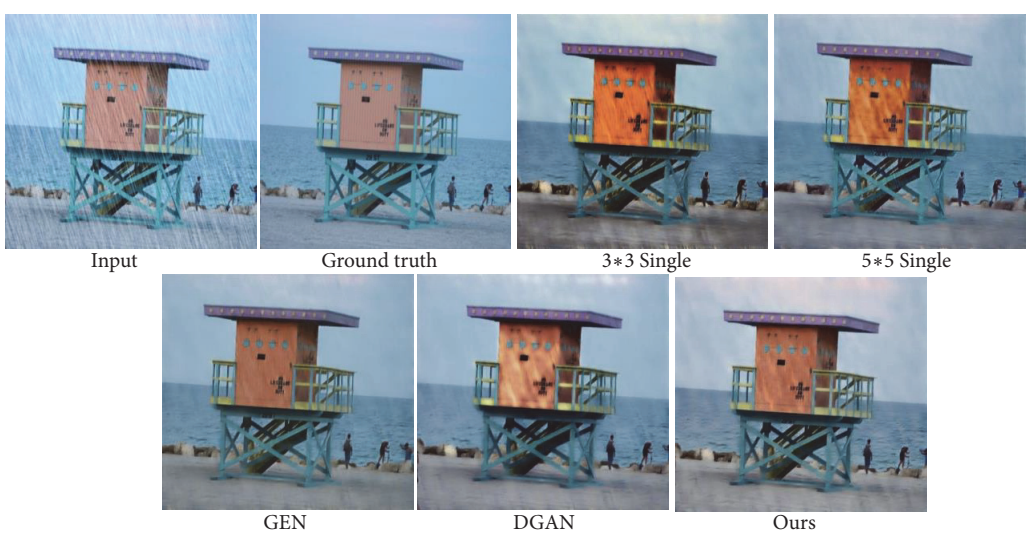
Results on Real-World Data. We also evaluate the proposed model on several real-world rainy images. Figure 4 presents the testing results on four real-world rainy images. From Figure 4, we can observe that both GMM-LP and DSC fail to completely remove rain streaks. Among them, the deraining



(a)

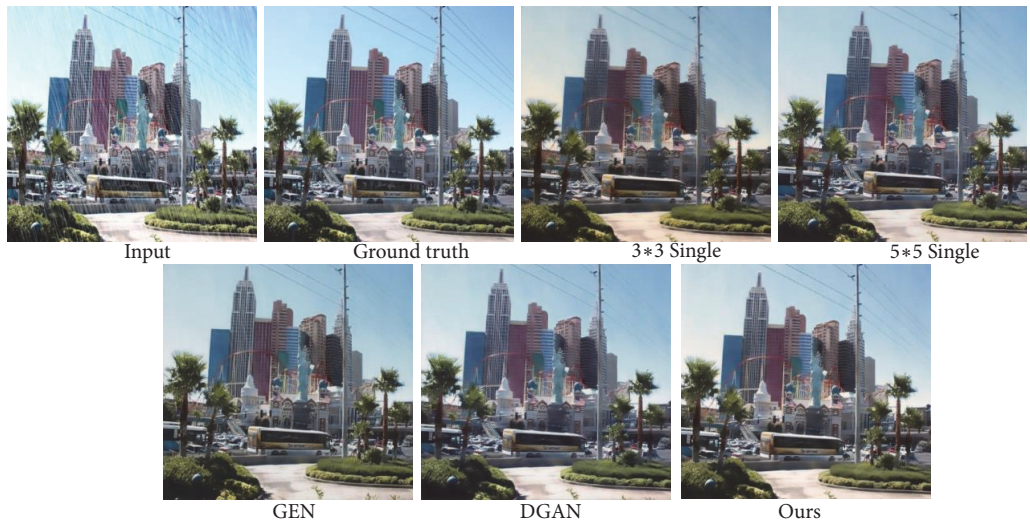


(b)

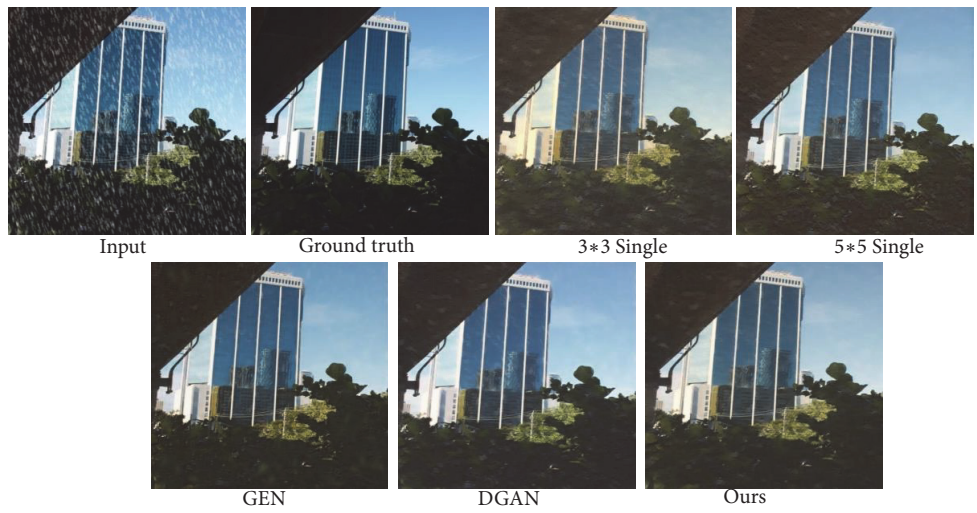


(c)

Figure 2: Continued.



(d)



(e)

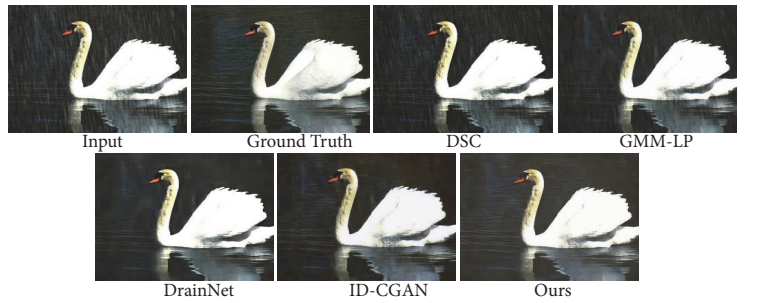
Figure 2: Results on five synthesized rainy images.

Table 5: VIF results on synthesized test image.

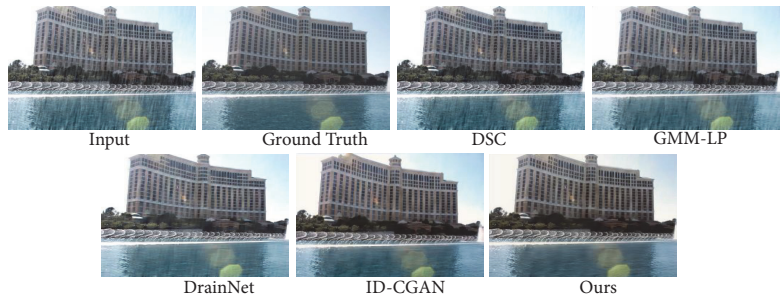
Images	DSC [17]	GMM-LP [13]	DrainNet [9]	ID-CGAN [11]	Ours
f	0.3260	0.4015	0.3899	0.3211	**0.4750**
g	0.4103	0.4389	0.4036	0.4187	**0.5058**
h	0.2144	0.2819	0.2514	0.3316	**0.3760**
i	0.3272	0.3909	0.3471	0.4224	**0.4580**
j	0.2836	0.3093	0.3076	0.0131	**0.4577**

effects of DrainNet and ID-CGAN are on a par with the proposed model from the visual perspective. However, compared with DrainNet and ID-CGAN, our derained results are able to maintain more details of the raw input images. In order to have a better comparison, we show one specific region of interest for each derained result of the five algorithms. From the regions of interests, we can see that, compared with other four algorithms, our model provides the best visual performance on jointly removing rain streaks and retaining details, which further verifies the validity of the proposed method.

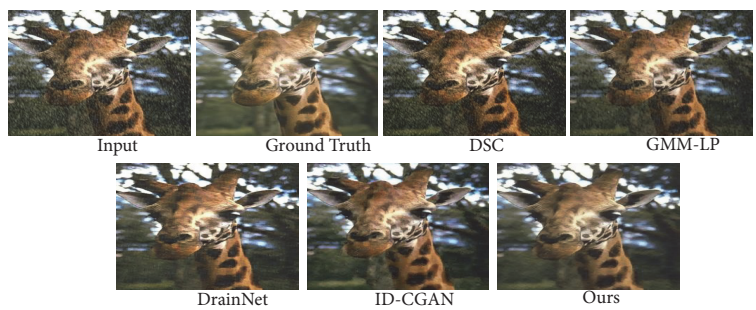
User Study Comparisons. Since there is no ground truth for real-world data, we constructed a user study to supply real feedback and quantify the subjective evaluation of the proposed model. We selected 10 real-world rainy images from



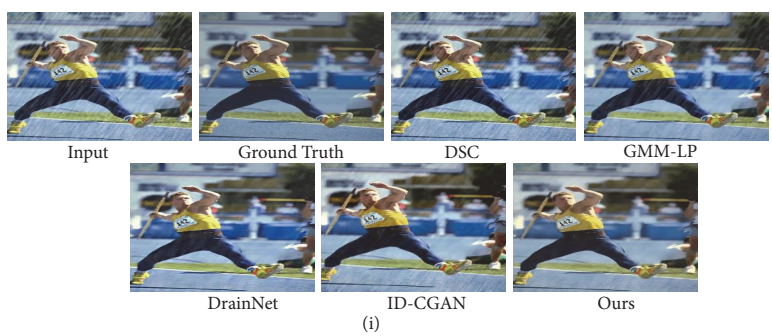
(f)



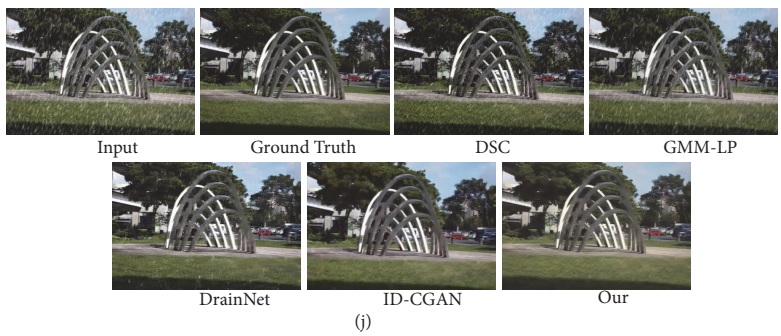
(g)



(h)

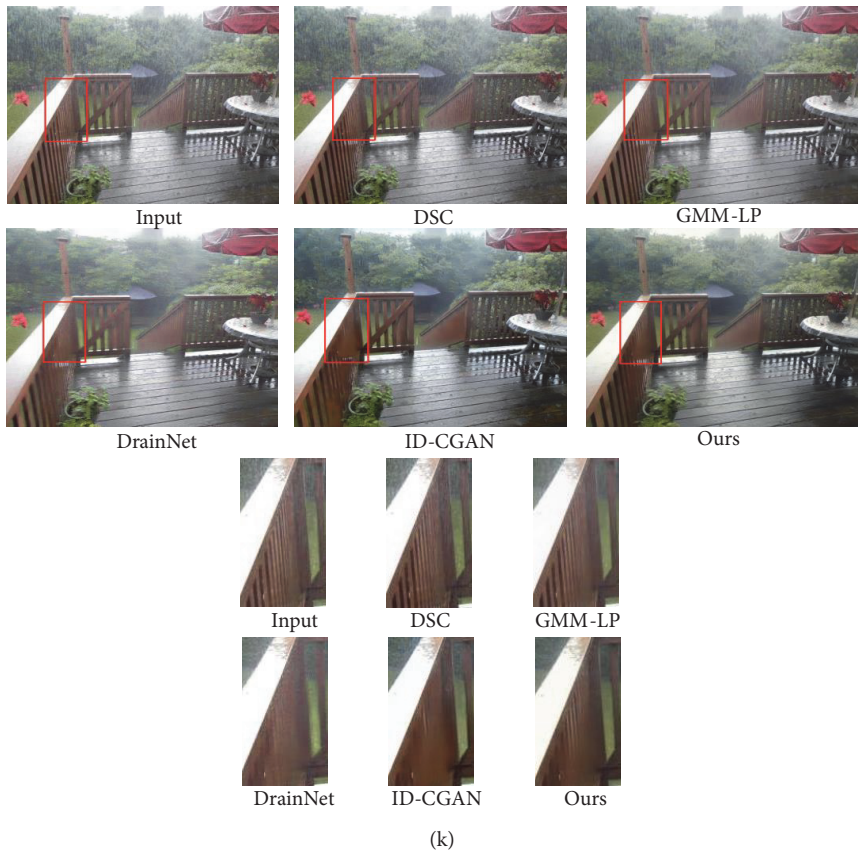


(i)

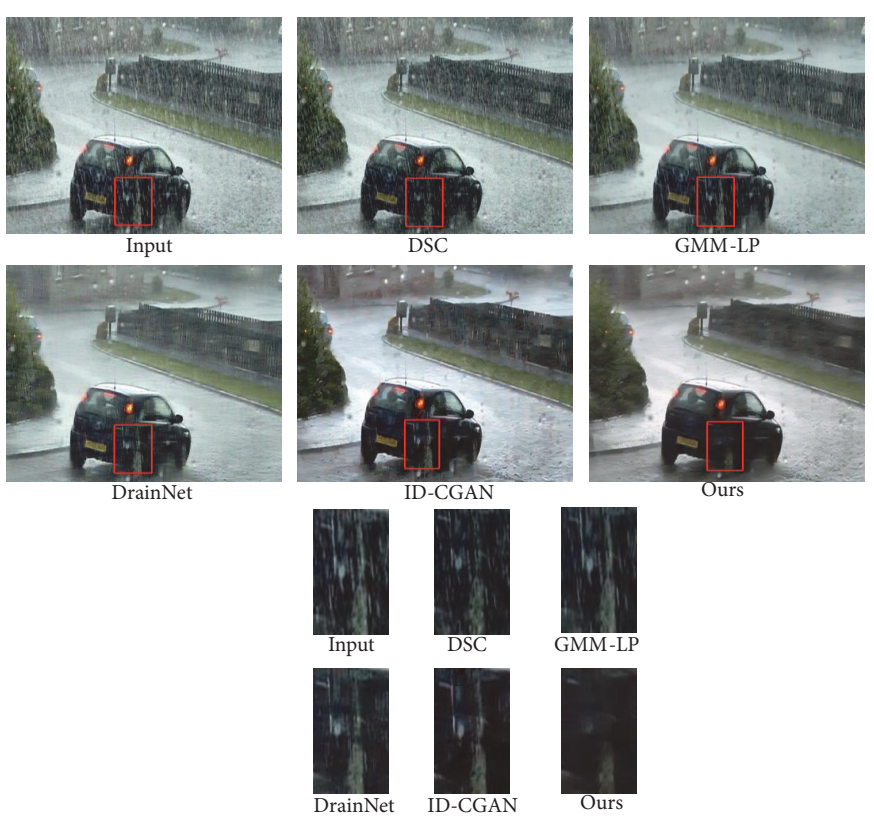


(j)

Figure 3: Results on five synthesized rainy images.



(k)



(l)

Figure 4: Continued.

Figure 4: Results on real-world rainy images.

Table 6: Average scores of user study.

Images	Input	DSC	GMM-LP	DrainNet	ID-CGAN	ours
Scores	1.34	1.87	2.06	3.55	3.98	4.07

Table 7: Running time (in seconds) of DR-GAN compared with State-of-the-art methods.

	DSC	GMM-LP	DrainNet	ID-CGAN	**Ours**
480×480	277.8s	566.1s	2.1s	1.2s	1.4s

Table 8: Parameter number for three deep learning based deraining methods.

DrainNet	ID-CGAN	ours
165888	18136	22714

the real-world dataset created by [11] and some rainy traffic images downloaded from the Internet. Figure 4 shows some derained results. For user study, we first used all methods to generate deraining images and randomly ordered the deraining results. Then, we displayed the ordered results on the screen and asked 10 participants with computer vision expertise to rank the results from 1 to 5, with 1 being the worst and 5 being the best. Table 6 shows the average subjective scores of the five methods on the real-world rainy images. We can see that our model achieves the highest average score among the five methods, which indicates that the proposed method can generate better deraining results on real-world rainy images from the subjective perspective.

Running Time and Parameter Number Comparisons. To estimate the efficiency of the proposed method, we computed the running time of the compared state-of-the-art methods as well as the proposed method. All the evaluations are implemented on the 480×480 rainy image. DSC and GMM-LP are non-deep learning methods that are implemented on CPU according to the provided code. DrainNet, ID-CGAN, and our methods are implemented on GPU. Table 7 presents the comparison results. In general, our multiscale network requires only 1.4 seconds to process a 480×480 rainy image, which is the same as the existing single-scale deep learning methods. Table 8 compares the parameter numbers of DrainNet, ID-CGAN, and the proposed network. Although our generator has two branches, it has less feature maps. It can be seen that the parameter number of our multiscale rain-removal network is slightly more than ID-CGAN, but less than DrainNet.

5. Conclusion

In this study, we have proposed a generative adversarial network-based architecture for single rainy image removal. The presented architecture consists of two subnetworks, *i.e.*, generation and discrimination subnetworks. The generation subnetwork with multiscale convolution operators can capture local rain drops and spatial information of rainy images simultaneously. To reserve more detail of the raw background of the rainy images and to improve the steadiness of the training process, three skip connections between the front convolution layers and later convolution layers are introduced. Acting as a supervisory signal, the discrimination subnetwork can be helpful in improving the quality of derained images generated by generation model. Experiments on synthetic and real-world images show that the proposed architecture outperforms other state-of-the-art methods. In the future, we will consider how to use the superresolution and attention mechanism to further improve the ability of deraining for the network.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China [Grant no. 61673222], the Jiangsu Universities Natural Science Research Project [Grant no. 13KJA510001], and the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie [Grant no. 701697].

References

[1] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004*, pp. I528–I535, USA, July 2004.

[2] D. Huang, L. Kang, Y. Wang, and C. Lin, "Self-learning based image decomposition with applications to single image denoising," *IEEE Transactions on Multimedia*, vol. 16, pp. 83–93, 2014.

[3] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742–1755, 2012.

[4] K. Garg and S. K. Nayar, "Photorealistic rendering of rain streaks," *Acm Transactions on Graphics*, vol. 25, pp. 996–1002, 2006.

[5] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2658–2670, 2015.

[6] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proceedings of the 14th IEEE International Conference on Computer Vision, ICCV 2013*, pp. 1968–1975, Australia, December 2013.

[7] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang, "Exploiting image structural similarity for single image rain removal," in *Proceedings of the IEEE International Conference on Image Processing*, pp. 4482–4486, 2015.

[8] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim, "Single-image deraining using an adaptive nonlocal means filter," in *Proceedings of the 20th IEEE International Conference on Image Processing, ICIP 2013*, pp. 914–917, Australia, September 2013.

[9] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: a deep network architecture for single-image rain removal," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2017.

[10] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 1715–1723, USA, July 2017.

[11] H. Zhang, V. Sindagi, and V. M. Patel, *Image De-Raining Using a Conditional Generative Adversarial Network, arXiv preprint*, arXiv, 1701.05957, 2017, arXiv:1701.05957.

[12] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, "Attentive Generative Adversarial Network for Raindrop Removal from a Single Image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2482–2491, 2018.

[13] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain Streak Removal Using Layer Priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2736–2744, 2016.

[14] K. Garg and S. K. Nayar, "Vision and rain," *International Journal of Computer Vision*, vol. 75, no. 1, pp. 3–27, 2007.

[15] Z. Xiaopeng, L. Hao, Q. Yingyi, K. L. Wee, and K. N. Teck, "Rain removal in video by combining temporal and chromatic properties," in *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME 2006*, pp. 461–464, Canada, July 2006.

[16] J. Bossu, N. Hautière, and J.-P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *International Journal of Computer Vision*, vol. 93, no. 3, pp. 348–367, 2011.

[17] Y. Luo, Y. Xu, and H. Ji, "Removing Rain from a Single Image Via Discriminative Sparse Coding," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3397–3405, 2015.

[18] H. Zhang, B. Chen, Z. Wang, and H. Liu, "Deep Max-Margin Discriminant Projection," *IEEE Transactions on Cybernetics*, pp. 1–13, 2018.

[19] H. Zhang, B. Chen, D. Guo, and M. Zhou, "WHAI, "Weibull hybrid autoencoding inference for deep topic modeling," in *Proceedings of the in International Conference on Learning Representations*, 2018.

[20] Y. Cong, B. Chen, H. Liu, and M. Zhou, "Deep latent Dirichlet allocation with topic-layer-adaptive stochastic gradient Riemannian MCMC," in *Proceedings of the International Conference on Machine Learning*, vol. 70, pp. 864–873, 2017.

[21] B. Chen, G. Polatkan, G. Sapiro, D. Blei, D. Dunson, and L. Carin, "Deep learning with hierarchical convolutional factor analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1887–1901, 2013.

[22] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to Deblur," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 38, pp. 1439–1451, 2014.

[23] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 2808–2817, USA, July 2017.

[24] H. Huang, R. He, Z. Sun, and T. Tan, "Wavelet-SRNet: A Wavelet-Based CNN for Multi-scale Face Super Resolution," in *Proceedings of the 16th IEEE International Conference on Computer Vision, ICCV 2017*, pp. 1698–1706, Italy, October 2017.

[25] L. Gatys, A. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *Journal of Vision*, vol. 16, no. 12, p. 326, 2016.

[26] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu, "Automatic Photo Adjustment Using Deep Neural Networks," *ACM Transactions on Graphics*, vol. 35, no. 2, pp. 1–15, 2016.

[27] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proceedings of the Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS '12)*, vol. 1, pp. 341–349, December 2012.

[28] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Joint Rain Detection and Removal Via Iterative Region Dependent Multi-Task Learning," *CoRR*, vol. abs/1609.07769, 2016.

[29] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least Squares Generative Adversarial Networks," in *Proceedings of the 16th IEEE International Conference on Computer Vision, ICCV 2017*, pp. 2813–2821, Italy, October 2017.

[30] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," https://arxiv.org/abs/1701.07875.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 770–778, July 2016.

[32] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," in *Computer Vision – ECCV 2016*, vol. 9906 of *Lecture Notes in Computer Science*, pp. 694–711, Springer International Publishing, Cham, 2016.

[33] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," https://arxiv.org/abs/1511.06434.

[34] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," https://arxiv.org/abs/1409.1556.

[35] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[36] G. Schaefer and M. Stich, "UCID - An uncompressed colour image database," in *Proceedings of the Storage and Retrieval Methods and Applications for Multimedia 2004*, pp. 472–480, USA, January 2004.

[37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[38] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, 2002.

[39] R. Collobert, K. Kavukcuoglu, and C. Farabet, *Torch7: A Matlab-Like Environment for Machine Learning*, NIPS Workshop, BigLearn, 2012.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 6868506, 9 pages
https://doi.org/10.1155/2018/6868506

Research Article

Research on Plaintext Restoration of AES Based on Neural Network

Xinyi Hu[1] **and Yaqun Zhao**[1,2]

[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi, China*
[2]*State Key Laboratory of Cryptography and Science, Beijing, China*

Correspondence should be addressed to Xinyi Hu; 290760485@qq.com

Received 12 July 2018; Revised 16 October 2018; Accepted 31 October 2018; Published 18 November 2018

Guest Editor: Ching-Nung Yang

Known plaintext attack is a common attack method in cryptographic attack. For ciphertext, only known part of the plaintext but unknown key, how to restore the rest of the plaintext is an important part of the known plaintext attack. This paper uses backpropagation neural networks to perform cryptanalysis on AES in an attempt to restore plaintext. The results show that the neural network can restore the entire byte with a probability of more than 40%, restoring more than half of the plaintext bytes with a probability of more than 63% and restoring more than half of the bytes above 89%.

1. Introduction

With the development of machine learning technology, the research of cryptanalysis is not limited to traditional manual deciphering. The intelligent deciphering based on machine learning, especially the emergence of intelligent deciphering based on neural network, provides a new development direction for cryptanalysis.

Neural network-based cryptanalysis can solve the shortcomings of traditional cryptanalysis methods in terms of attack difficulty and the amount of data required for attacks. First of all, neural network as an ideal black box recognition tool can effectively analyze and simulate the black box problem (cryptanalysis problem), infinitely approach the cryptanalysis problem, and finally get the algorithm equivalent to the encryption and decryption algorithm to achieve cryptanalysis. In this case, the neural network does not need to restore the key of the algorithm and the specific setting parameters and only needs to input the ciphertext, and after training, the corresponding plaintext can be obtained with a certain probability. Secondly, as a machine learning method, neural network can effectively achieve the corresponding cryptanalysis results and can achieve the final deciphering algorithm with less training set (plaintext-ciphertext pair). Therefore, neural networks are rapidly recognized by the industry as a method of cryptanalysis and are gradually called a new research direction in the field of cryptanalysis.

The application of neural network in cryptanalysis is mainly used for the global deduction of cryptographic algorithms [1] (the algorithm that the attacker obtains and encrypts and decrypts may not know the key) and the complete crack [1] (the attacker obtains the key).

In 2008, Bafghi et al. [2] used a neural network model to represent the differential operation of block ciphers to find the difference features. The differential feature space of the block cipher can be represented by a multilevel weighted directed graph, so the problem of finding the best differential feature can be transformed into the problem of finding the least weight multibranch path between two known nodes in the directed graph. In this paper, the cyclic neural network (RNN) is used to find the path by minimizing the network cost function in the differential operation graph of the block cipher. In 2010, Alallayah et al. [3–5] performed neural network-based cryptanalysis on classical cryptography, sequence ciphers, and simplified DES (SDES). They regard cryptanalysis as a black box problem, using neural networks as an ideal tool for identifying black boxes, combining system identification technology with adaptive system technology, and constructing a neuroidentifier that simulates the target cryptosystem. A neural model and the key can be determined

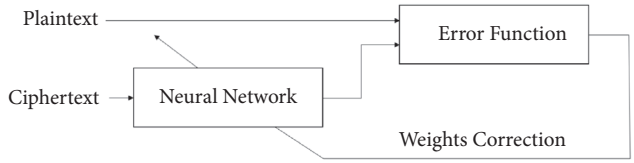


Figure 1: System structure of neural network cryptanalysis.

from a given pair of ciphertexts. In 2012, Alani et al. [6, 7] implemented a known plaintext attack based on neural networks for DES and 3DES. In the experiment, they trained a neural network to retrieve plaintext from ciphertext without having to retrieve the keys used in encryption. This kind of attack is successfully applied to DES and 3DES. For DES, an average of 2^{11} pairs of ciphertext pairs are needed, and cryptanalysis can be completed in about 51 minutes to achieve a global deductive effect. For the 3DES cryptanalysis, only an average of 2^{12} plaintext-ciphertext pairs are needed, and the analysis can be completed in about 72 minutes. Corresponding results significantly reduce the number of known plaintexts required and reduce the time required to perform a full attack compared to other attacks. In 2013, Bahubali Akiwate et al. [8] proposed a neural network-based cryptanalysis of the DES algorithm, aiming to analyze the nonlinear characteristics of DES through neural networks. The text data is used as the plaintext in the analysis, and the ciphertext encrypted by DES is used as the input of the neural network and is trained together with the plaintext to obtain the corresponding output. The neural output is compared with the plaintext, and performance error indicators are established for analyzing the data to improve efficiency. In 2014, Danziger et al. [9] used the method in [5] to find out the mapping relationship between plaintext, ciphertext, and key in SDES. When experiments are performed using 102,400 sample data, the keys of the first, second, and fifth bits of the 10-bit key can be obtained; when the number of samples is reduced to 2000, the keys of the first and second bits can be obtained. Together with the differential analysis, they improved the S-box of SDES, which reduced the correlation between adjacent keys in the key space, making it resistant to key restoration attacks in neural network cryptanalysis.

Inspired by [6, 7], this paper uses the methods in [6, 7] to construct an analysis framework similar to them. According to the block length and data type of AES algorithm, the data processing process suitable for AES algorithm is set. Then, according to the research object and the structural characteristics of neural network, the specific experimental system structure is designed. The AES algorithm in ECB and CBC mode is used to perform cryptanalysis using neural network. Without knowing the key, the plaintext is restored from the ciphertext, trying to achieve the effect of global deduction. The results show that the plaintext restored by the AES-128 and AES-256 algorithms over the neural network is more than 63% higher than the plaintext compared with the original plaintext.

Section 2 of this paper introduces the specific process of cryptanalysis based on neural networks. Section 3 designs the steps of the overall experiment and selects the relevant parameters. Section 4 is the experimental results. Section 5 gives the conclusion.

Table 1: Experimental equipment.

Version	macOS High Sierra 10.13.1
Processor	1.2 GHz Intel Core m5
RAM	8 GB 1867 MHz LPDDR3
Graphics Card	Intel HD Graphics 515 1536 MB

2. Cryptanalysis Process Based on Neural Network

The cryptanalysis method based on neural network utilizes the learning ability of the neural network to train the neural network with the known Ming ciphertext. After the training is completed, the neural network can restore the plaintext from the ciphertext that does not belong to the training set. The corresponding system structure is shown in Figure 1. The system contains an information forward propagation process (ciphertext ⟶ neural network ⟶ plaintext), and an error backpropagation process (plaintext ⟶ error function ⟶ weight correction ⟶ neural network).

The ciphertexts are input in the neural network, and the output results are compared with the known plaintexts to obtain an error function. The weight is continuously corrected according to the error until the neural network is successfully trained. And finally the plaintext can be restored with a certain probability. This attack method is considered to be a global deductive attack, which is functionally equivalent to the original decryption algorithm without knowing the key. This analysis method is similar to the global approximation method for multilayer feedforward neural networks in [11].

In order to train a neural network with an acceptable error rate, it is necessary to expand the network size, so it is necessary to increase the time of each training cycle. There are many related parameters that need to be set in the neural network, such as the number of neurons, the number of hidden layers, the training function, etc. These parameters will be specified in Section 3.

3. Experimental Design and Parameter Selection

3.1. Experimental Environment. The relevant experiments were carried out in MATLAB_R2016a with a neural network toolbox. The equipment used in the experiment was MacBook, and the relevant data is shown in Table 1.

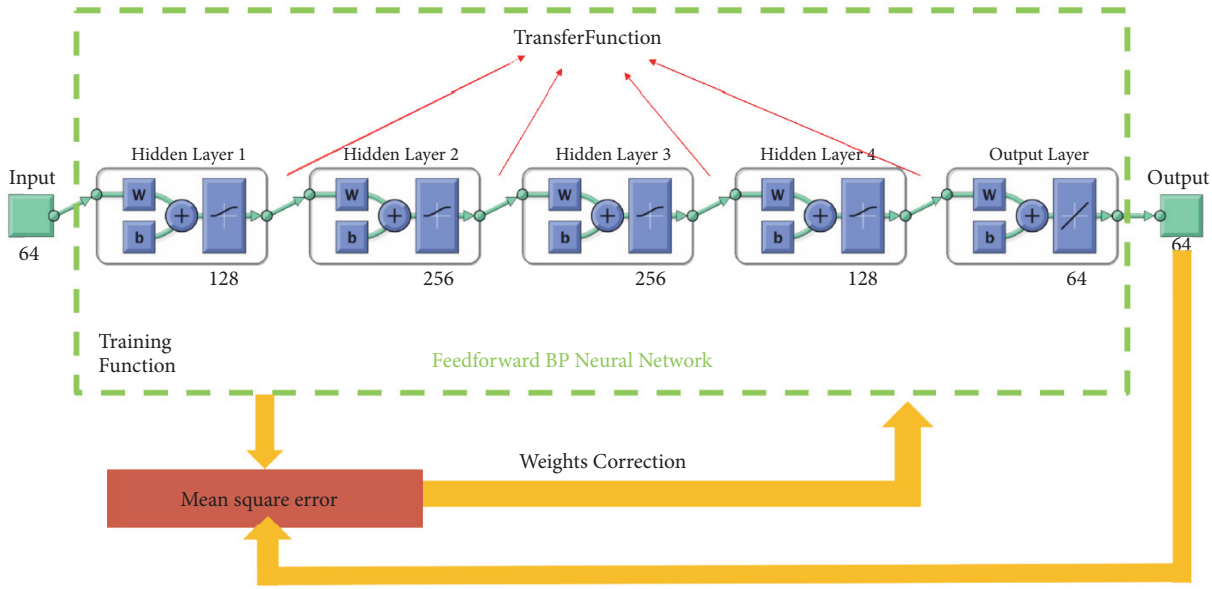


Figure 2: System structure based on feedforward BP neural network.

3.2. Data Representation. We divide the ciphertext data into 64-bit blocks, that is, get 64 channels of input and output, and represent them in matrix form in MATLAB.

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,k} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,k} \\ p_{3,1} & p_{3,2} & \cdots & p_{3,k} \\ \vdots & \vdots & \ddots & \vdots \\ p_{64,1} & p_{64,2} & \cdots & p_{64,k} \end{bmatrix},$$
$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,k} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,k} \\ c_{3,1} & c_{3,2} & \cdots & c_{3,k} \\ \vdots & \vdots & \ddots & \vdots \\ c_{64,1} & c_{64,2} & \cdots & c_{64,k} \end{bmatrix} \quad (1)$$

where P is the plaintext matrix, C is the ciphertext matrix, $p_{(i,j)}$ is the i-th plaintext bit in the j-th plaintext block, $c_{(i,j)}$ is the i-th in the j-th ciphertext block A ciphertext bit, and k is the number of blocks of the ciphertext pair used in the training. Each bit in C corresponds to each bit in P. According to the number of inputs and outputs of the neural network, the number of rows in the two matrices is selected, and all ciphertexts in the ciphertext pair are encrypted by the same key.

3.3. Size and Layout of Neural Network. In the experiment, we choose feedforward backpropagation neural network and cascaded feedforward backpropagation neural network, in which the hidden layer and output layer of the feedforward BP neural network are only affected by the previous layer. The hidden layer and the output layer of the cascaded feedforward BP neural network are related to each of the previous layers. The specific experimental structure is shown in Figures 2 and 3.

The relevant parameters of the neural network are set as follows:

(i) Set four hidden layers, each of which has 128, 256, 256, and 128 neurons. Since the experiment in [6] contains several hidden layer settings. When the setting is four layers, and the neurons in each layer are 128, 256, 256, and 128, respectively, the experiment is successful and the results are better. So we choose this setting. In the future research, we will also change the settings of the hidden layer, taking into account the impact of different settings on the results.

(ii) The training function of the neural network selects the quantized conjugate gradient method 'trainscg', which is suitable for large networks and occupies storage due to the conjugate gradient method. The space is small, and the 'trainscg' quantized conjugate gradient method saves more time than other conjugate gradient methods.

(iii) The error function that selects the correction weight during training is the mean square error (MSE). The error function is also called the loss function. The two most commonly used loss functions are cross entropy and mean square error. Among them, the cross entropy characterizes the distance between two probability distributions, which is a loss function that is used more in the classification problem. Unlike the classification problem, the regression mainly solves the prediction of specific numerical values. The restoration of plaintext from the known ciphertext

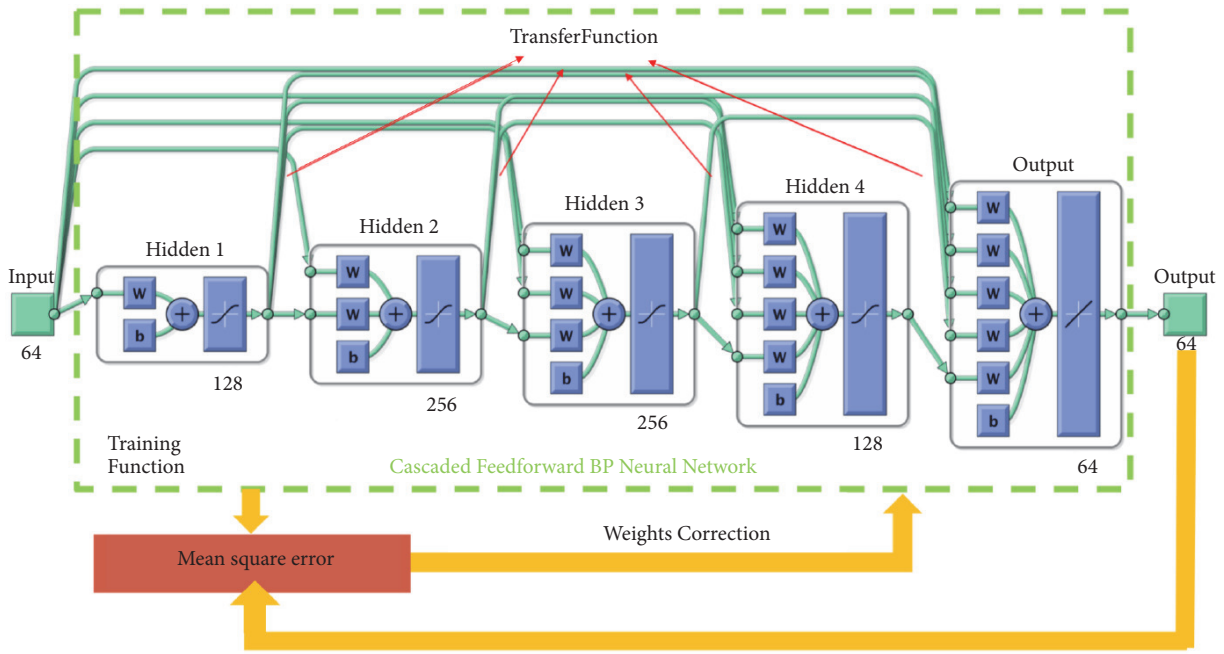


Figure 3: System structure based on cascaded feedforward BP neural network.

belongs to the regression problem. For the regression problem, the most commonly used loss function is the mean square error.

3.4. Training Stop Condition. The maximum number of training cycles is set to 500, and there are three training stop conditions, namely, (1) reaching the final training cycle number of 500; (2) the acceptable mean square error limit of 0.05; (3) continuous verification failure The maximum number of times is 20.

3.4.1. Training Phase. (1) The training process first creates a neural network and selects the network layout, that is, determines the corresponding neurons in each layer, and then sets the training stop conditions.

(2) At the beginning of the training, part of the data in the data set is used as the training set, and the rest is used as the test set. The training of the ciphertext in the training set is performed, the ciphertext is input and the corresponding result is output, and the result is XORed with the original plaintext, and the weight is continuously corrected according to the error function.

(3) The training is over and it is judged whether the training is successful. If the number of consecutive verification failures reaches 10, and the mean square error is higher than 0.1, the training fails; if the mean square error is less than 0.1, the training is successful.

(4) If the training fails, reinitialize the network and return to step (2) to start retraining.

(5) If the training is successful, enter the test process.

3.4.2. Test Phase. (1) After the training is completed, the ciphertext matrix C_{train} used in the training is input into the neural network to obtain the corresponding plaintext result P'_{train}. The P'_{train} is subjected to a bitwise XOR operation with the plaintext P_{train} of the training set, and the error percentage is calculated, which is called the training error. Training errors can be expressed by the following formula:

$$train\_error = \frac{\sum_{i=1}^{m_{train}} \sum_{j=1}^{n_{train}} p'_{train}(i,j) \oplus p_{train}(i,j)}{m_{train} \times n_{train}} \quad (2)$$

where m_{train} is the number of bits per block in the training set, n_{train} is the number of blocks in the training set, $p'_{train}(i,j)$ is the j-th bit in the i-th block of the neural network output, and $p_{train}(i,j)$ is the j-th bit in the i-th block of the plaintext used in the training. In this experiment, $m_{train} = 64$.

(2) The ciphertext C_{test} different from the training set is input into the neural network, and the corresponding result P'_{test} of the neural network is obtained. The P'_{test} and the plaintext P_{test} of the training set are XORed bit by bit, and the error percentage is calculated, which is called the test error. The test error can be expressed by the following formula:

$$test\_error = \frac{\sum_{i=1}^{m_{test}} \sum_{j=1}^{n_{test}} p'_{test}(i,j) \oplus p_{test}(i,j)}{m_{test} \times n_{test}} \quad (3)$$

where m_{test} is the number of bits per block in the test set, n_{test} is the number of blocks in the test set, $p'_{test}(i,j)$ is the j-th bit in the i-th block of the output during the test, and $p_{test}(i,j)$ is the j-th bit in the i-th block of the plaintext used in the test. In this experiment, $m_{test} = 64$.

(3) The ciphertext C_{total} of the entire data set is input into the neural network, and the corresponding result P'_{total} is obtained, and the output P'_{total} and the original plaintext P_{total} are XORed bit by bit, and the calculated error percentage is

Table 2: 2^{11}-size file in the feedforward BP neural network error byte 0-8 bits per byte (unit: byte).

Algorithms	0	1	2	3	4	5	6	7	8	total
AES-128_ECB	118	46	24	23	22	15	7	1	0	256
AES-128_CBC	126	45	20	21	21	13	7	2	0	256
AES-256_ECB	127	44	18	21	20	16	8	2	0	256
AES-256_CBC	117	48	24	21	21	16	7	2	0	256

Table 3: 2^{11}-size file in the cascaded feedforward BP neural network error byte 0-8 bits per byte (unit: byte).

Algorithms	0	1	2	3	4	5	6	7	8	total
AES-128_ECB	128	31	22	24	25	16	8	2	0	256
AES-128_CBC	104	48	29	23	24	17	9	2	0	256
AES-256_ECB	151	24	15	18	22	15	9	2	0	256
AES-256_CBC	104	35	32	33	24	18	7	3	0	256

called the overall error. The overall error can be expressed by the following formula:

$$total\_error = \frac{\sum_{i=1}^{m_{total}} \sum_{j=1}^{n_{total}} p'_{total}(i,j) \oplus p_{total}(i,j)}{m_{total} \times n_{total}} \quad (4)$$

where m_{total} is the number of bits per block in the overall data set, n_{total} is the number of blocks in the data set, $p'_{total}(i,j)$ is the j-th bit in the i-th block of the output, and $p_{total}(i,j)$ is plaintext of the j-th bit in the i-th block. In this experiment, $m_{total} = 64$.

(4) Compare the output result P'_{total} with the original plaintext P_{total}, and count the number of pairs of each byte and the number of errors 1-8 bits, and count the number of consecutive two bytes, four bytes, and eight bytes. The reason for counting the total number of consecutive bytes is that, in Chinese, one Chinese character is two bytes, and the words are generally composed of two Chinese characters, that is, four bytes, and four Chinese characters (eight bytes) may form a phrase or a sentence. So if the plaintext is composed of Chinese, part of the plaintext can be understood by restoring some of the Chinese characters. The same applies to English and other languages. The more the consecutive pairs of bytes correct, the better the understanding of the content of the plaintext.

4. Experimental Result

Experiments are specific to the AES-128, ECB, and CBC modes of the AES-256 algorithm. According to the experimental results in [6] and our current ciphertext data, the experiment consists of two parts: one is to select the image in the Caltech Image Database [10] and splicing into 1000 files of 512 KB size. In plain text, the above four kinds of cryptographic algorithms are used to encrypt each, 1,000 ciphertext files are, respectively, obtained, then the ciphertext files are truncated, the file length is still controlled to 512 KB, a total of 4,000 ciphertext files of 512 KB size are obtained, and then the ciphertext is obtained. Converted to 01 sequence, respectively, and intercepted the first 2^{11} digits for experiment. Second, Python randomly generated 1,000 files of 2^{17} characters as plaintext, respectively, encrypted with AES-128, AES-256 algorithm ECB and CBC modes to get 1,000 ciphertext files, and then cut off the ciphertext files, control the file length to 2^{17} characters. A total of 4,000 ciphertext files with a length of 2^{17} characters are obtained, and then the ciphertexts are converted into 01 sequences. A 01 sequence with a size of 2^{20} is obtained. The result of the experimental output is between $[0, 1]$, we round it up, the result of the result less than 0.5 is 0, and the result of the result greater than or equal to 0.5 is 1.

In the specific experiment, in the first part, we select the first 85% of the data set as the training set and the last 15% as the test set and count the output results of each size of 2^{11}, each byte error 0-8 bits number. The corresponding results are shown in Tables 2 and 3. Since [3, 8] and this paper are all aimed at restoring plaintext, the restoration accuracy of this paper and [3, 8] is compared. The corresponding results are shown in Table 4. In the second part, we select the first 2^{11}-, 2^{12}-, 2^{13}-, 2^{14}-, 2^{15}-, and 2^{16}-bit training neural networks in a data set and count each 0-8-bit error in each byte of the output result of 2^{20}. The corresponding results are shown in Figures 4 and 5. And the number of consecutive two bytes, four bytes, eight pairs of pairs, and the corresponding results are shown in Figures 6, 7, and 8.

It can be seen from Tables 2 and 3 that the two neural networks have little difference in the plaintext restoration results of the four algorithms, and the full pair of bytes can be more than 100, accounting for more than 40% of all bytes. The number of more than half of the bits in each byte is more than 89% of all bytes. Therefore, these two neural networks have a good effect on the plaintext restoration in cryptanalysis, which can effectively reduce the number of exhaustive, significantly shorten the analysis time, and greatly reduce the required resources.

It can be seen from Table 4 that the epoch of the four AES algorithms are all below 50, the MSE is more than 0.05, the epoch of DES and 3DES are both above 200, and the MSE is less than 0.04. This is because the AES algorithm is the latest data encryption standard, and the security is higher than DES and 3DES. It is difficult to restore the plaintext with high

Table 4: Accuracy comparison of AES, DES, and 3DES.

Algorithms	AES-128_ECB	AES-256_ECB	AES-128_CBC	AES-256_CBC	DES_ECB	3DES_ECB
Average epoch	44	37	46	45	352	239
Average MSE	0.0501	0.0536	0.0569	0.0611	0.0308	0.0372
Average error	0.1768	0.2095	0.1699	0.1909	0.083	0.114
Experiment data source and size	[10], 2^{11}bit				unexplained, 2^{20}bit	
Average size of data required for successful experiments	≈1741bit				2^{11}bit	2^{20}bit
Results source	this paper				[6]	[7]

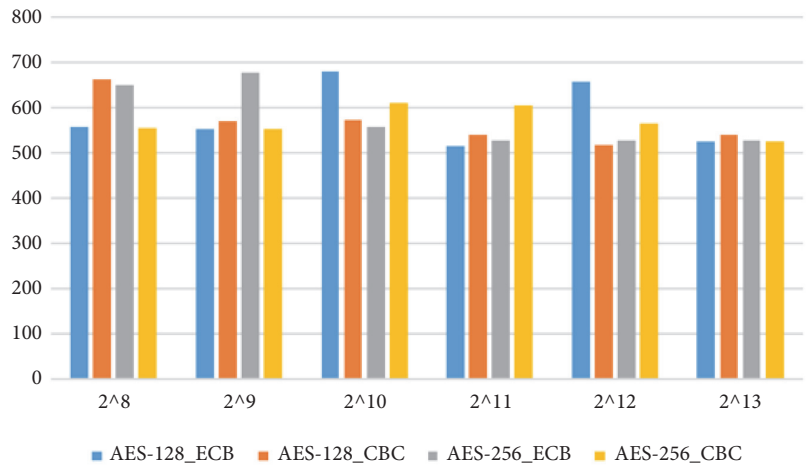


Figure 4: Number of bytes all correct of feedforward BP neural networks.

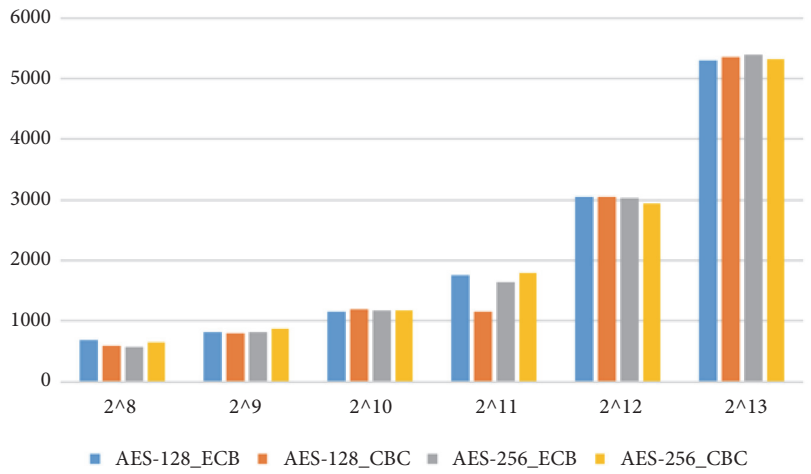


Figure 5: Number of bytes all correct of cascaded feedforward BP neural networks.

accuracy only through 2048 plaintext-ciphertext pairs, and this experiment only uses personal laptop for experiment; memory and performance are limited. Therefore, the epoch is small, resulting in higher error than DES and 3DES. However, only the DES and 3DES algorithms in the ECB mode are considered, the CBC mode is not considered, and the block length of the algorithms and the experimental data source are not explained in the [6, 7].

From Figure 4, in the experimental results of the feedforward BP neural network, the number of bytes in the pair is between 500 and 700, and the number of more than half of the bits in each byte accounts for more than 63%

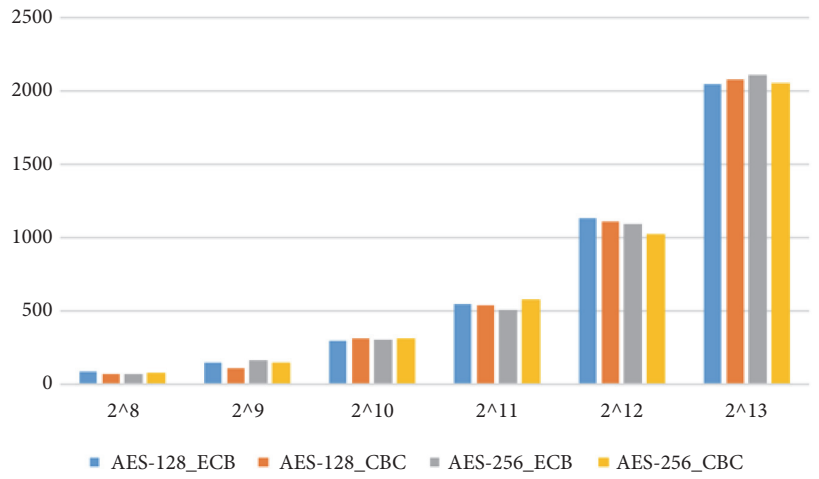


Figure 6: Number of two consecutive bytes all correct of cascaded feedforward BP neural networks.

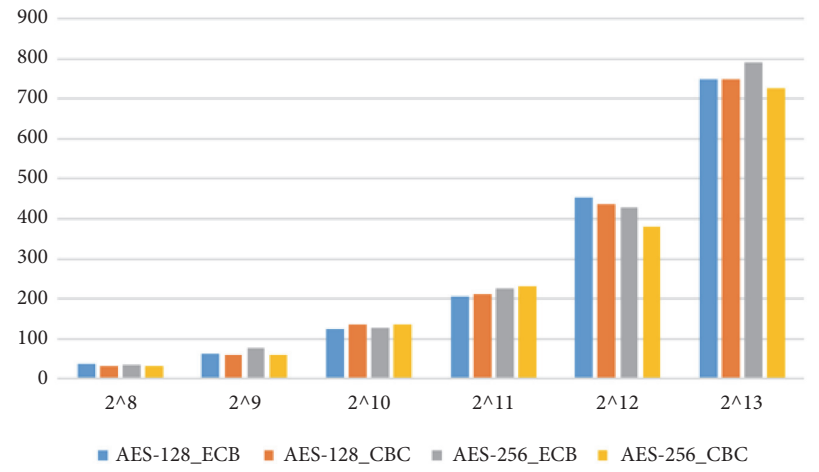


Figure 7: Number of four consecutive bytes all correct of cascaded feedforward BP neural networks.

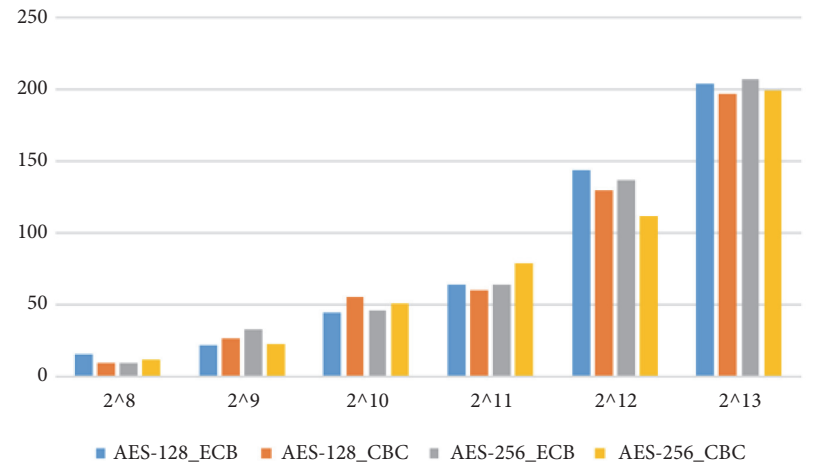


Figure 8: Number of eight consecutive bytes all correct of cascaded feedforward BP neural networks.

of all bytes. And the statistical number of each algorithm is normally distributed. It can be seen from Figure 5 that the cascaded feedforward BP neural network also has a normal distribution for each algorithm, and as the training set data increases, the number of bytes of the pair increases. In the experiment, we also found that for the feedforward BP neural network, when the training set is increased to 2^{10} bytes, the training continues to fail, and the MSE is always above 0.1; while the training success rate of the cascaded feedforward BP neural network is always, it stays above 99%, and as the training set increases, Epoch also increases. For the feedforward BP neural network, there is no obvious linear relationship between the size of the training set data and the number of bytes of the full pair. For the cascade feedforward BP neural network, the size of the training set data, and the pair of words there is a significant positive correlation between the number of sections. As the training set data increases, the number of bytes in the pair increases. Therefore, when the amount of training data increases to a certain extent, the cascade feedforward BP neural network can effectively improve the efficiency of plaintext restoration and reduce the number of exhaustive times required.

Since the feedforward BP neural network is not good for restoring consecutive bytes, this paper only shows the restoration results of the cascade feedforward BP neural network. It can be seen from Figures 6–8 that as the training set increases, the number of bytes all correct increases. When the training set size is the same, the total number of bytes all correct of the four algorithms does not differ much. Figure 6 shows the number of consecutive two bytes all correct. When the training set reaches 2^{13} bytes, there are an average of more than 2,000 two bytes all correct, accounting for more than 3% of the whole. Figure 7 shows the number of four consecutive bytes all correct, when the training set reaches 2^{13} bytes, there are more than 700 four bytes all correct, accounting for more than 2.2% of the whole. Figure 8 shows the number of consecutive eight bytes all correct. When the training set reaches 2^{13} bytes, there are about 200 eight bytes all correct, accounting for about 1.2% of the whole.

5. Conclusion

This paper discusses the global deductive study of AES-128 and AES-256 algorithms. For the research goal, we use the feedforward BP neural network and the cascade feedforward BP neural network to restore the ciphertexts of the AES-128 and AES-256 algorithms in ECB and CBC modes. As a new method for cryptanalysis, neural network can restore the corresponding plaintext according to ciphertext. In the restored result, the number of bytes in all pairs is above 40%, and the number of bytes in more than half is 89%. Above, and for the cascade feedforward BP neural network, as the training set data increases, the error rate decreases, and the number of pairs of all pairs increases. In the global deductive study, we found that different neural networks will get different results, and different data types will lead to differences in error rates. Therefore, we will use this as an opportunity to continue to study different neural networks and consider the impact of more data types on error rates.

As an emerging cryptanalysis method, the cryptanalysis of plaintext restoration based on neural network is still in the experimental stage. The research on AES is also a new attempt. To get better results, we need to know more plaintext in advance and set more restrictions. Future research may require specific structural features specific to the AES algorithm to perform cryptanalysis more efficiently.

Data Availability

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

Disclosure

Conflicts of Interest

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

Acknowledgments

This work is supported by the Open Project Program of the State Key Laboratory of Mathematical Engineering and Advanced Computing; National Key Research and Development Project 2016-2018 (2016YFE0100600); Open Fund Project of Information Assurance Technology (KJ-15-008); State Key Laboratory of Cryptography and Science.

References

[1] L. R. Knudsen, "Block ciphers—a survey," in *State of the art in applied cryptography (Leuven, 1997)*, vol. 1528 of *Lecture Notes in Comput. Sci.*, pp. 18–48, Springer, Berlin, 1998.

[2] A. G. Bafghi, R. Safabakhsh, and B. Sadeghiyan, "Finding the differential characteristics of block ciphers with neural networks," *Information Sciences*, vol. 178, no. 15, pp. 3117–3131, 2008.

[3] A. Hussein Al-Hamami, K. Alallayah, A. Mohamed, and W. Abdelwahed, "Applying neural Networks for simplified data Encryption Standards (SDES) Cipher System Cryptoanalysis," *International Arab Journal of Information Technology*, vol. 9, no. 2, pp. 2423–2432, 2012.

[4] K. Alallayah, M. Amin, W. A. El-Wahed, and A. Alhamami, "Attack and construction of simulator for some of cipher systems using Neuro-Identifier," *International Arab Journal of Information Technolog*, vol. 7, no. 4, pp. 365–372, 2010.

[5] K. M. Alallayah, W. F. Abd El-Wahed, M. Amin, and A. H. Alhamami, "Attack of against simplified data encryption standard cipher system using neural networks," *Journal of Computer Science*, vol. 6, no. 1, pp. 29–35, 2010.

[6] M. M. Alani, "Neuro-Cryptanalysis of DES and Triple-DES," in *Neural Information Processing*, vol. 7667 of *Lecture Notes*

in Computer Science, pp. 637–646, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[7] M. M. Alani, "Neuro-cryptanalysis of DES," in *Proceedings of the World Congress on Internet Security, WorldCIS-2012*, pp. 23–27, Canada, June 2012.

[8] A. Bahubali and V. Desai, "Artificial Neural Networks for Cryptanalysis of DES," *International Journal of Innovations in Engineering and Technology*, vol. 2, no. 4, pp. 11–17, 2013.

[9] M. Danziger and M. A. Henriques, "Improved cryptanalysis combining differential and artificial neural network schemes," in *Proceedings of the 2014 International Telecommunications Symposium (ITS)*, pp. 1–5, Sao Paulo, Brazil, August 2014.

[10] G. Griffin, A. Holub, and P. Perona, *Caltech-256 Object Category Dataset*, California Institute of Technology, 2007.

[11] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 5783976, 12 pages
https://doi.org/10.1155/2018/5783976

Research Article

Multibiometric Fusion Authentication in Wireless Multimedia Environment Using Dynamic Bayesian Method

Zhendong Wu[1], **Jiajia Yang**[2], **Jianwu Zhang**[2] **and Hengli Yue**[1]

[1]*School of Cyberspace, Hangzhou Dianzi University, Hangzhou, Zhejiang, China*
[2]*School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, Zhejiang, China*

Correspondence should be addressed to Zhendong Wu; wzd@hdu.edu.cn

Received 10 May 2018; Revised 28 August 2018; Accepted 10 September 2018; Published 18 November 2018

Guest Editor: Zhaoqing Pan

Single biometric method has been widely used in the field of wireless multimedia authentication. However, it is vulnerable to spoofing and limited accuracy. To tackle this challenge, in this paper, we propose a multimodal fusion method for fingerprint and voiceprint by using a dynamic Bayesian method, which takes full advantage of the feature specificity extracted by a single biometrics project and authenticates users at the decision-making level. We demonstrate that this method can be extended to more modal biometric authentication and can achieve flexible accuracy of the authentication. The experiment of the method shows that the recognition rate and stability have been greatly improved, which achieves 4.46% and 5.94%, respectively, compared to the unimodal. Furthermore, it also increases 1.94% when compared with general multimodal methods for the biometric fusion recognition.

1. Introduction

Biometric feature analysis has been widely studied for decades as it is a vital way for authentication and safeguard in computer vision. However, traditional biometrics, such as fingerprinting and vein recognition, gradually reveals some of its drawbacks that it can already be assigned and mimicked by forging fingerprints or faces [1]. As fusing features such as facial features, fingerprints, palm prints, sounds, and irises improves the stability, accuracy, and unforgeability of biometrics, multimodal biometric systems could help relieve the problem brought by the single-modal biometric systems and provide tremendous help for more secure authentication and identification.

There have been some researches about multimodal biometrics. Conti et al. [2] fused fingerprint and iris using homogeneous biometric vector through Log-Gabor filtering. Nagar et al. [3] studied the fusion of three biological feature (iris, fingerprint, and face) by using fuzzy vault and fuzzy commitment model to form a biometric encryption system framework. Snelick et al. [4] used a new method of normalization and fusion strategies to fuse and identify the biometrics of fingerprints and face at the score level. Muthukumar et al. [5] fused iris and fingerprint at the score level based on an evolutionary algorithm, Particle Swarm Optimization, which can help the authentication system adapt to different security needs. Shekhar et al. [6] used sparse matrices fusing the same three characteristics (iris, fingerprint, and face). Sparse matrix method has good recognition robustness. The above improvement in biometric identification demonstrates that there are many advantages of multimodal biometric identification.

On the other hand, we find some limitations about the existing researches, they are as follows: (1) the above articles all chose to integrate multibiometrics at a certain level but did not take into account the fact that multiple biometric features may interfere with each other, thereby reducing the recognition effect. (2) Most of the fusing at the decision layer always takes fixed weights. This is based on the overall average quality, but it is not the best solution for every decision. For example, if the fingerprint recognition rate is higher than the voiceprint overall, then it will be given a higher weight in the fusion recognition; however, the fingerprints are not always better than the voiceprint quality.

Inspired by these ideas, we propose a multibiometric fusion authentication solution by using dynamic Bayesian decision method, which is named MFDB-decision (Multibiometric Fusion using Dynamic Bayesian decision). The key idea of this work is that using matching layer score assists decision layer in fusing fingerprints and voiceprints aiming at recovering identity information lost in decision layer and, besides, overcoming the above problems caused by fixed weights. This paper uses fingerprint and voiceprint in multimodal fusion, because of the stability of the fingerprint and the high user acceptance of the voiceprint. The method proposed in this paper can be extended to more dimensional biometric fusion authentication, instead of being limited to the voiceprint and fingerprint.

The outline of this paper is organized as follows. Section 2 presents some related work. The preliminary research about fingerprint feature extraction and voiceprint feature extraction is introduced in Section 3. The multibiometric authentication fusion algorithm MFDB-decision is described in Section 4. The analysis of the experiments and results is given in Section 5. Finally, the paper would be concluded in Section 6.

2. Related Work

There has been a great deal of research on the application based on single biometric identification, especially in the fields of biological key [7–9], cloud computing data security [10–13], blockchain [14], privacy preserving [15–18], and biological template protection [19–21]. However, there are still not so many studies on multibiometrics. The study by Windsor Holden [22] further increased the application of multibiometric methods in the fields of common life other than criminal investigation.

Multimodal biometrics research attempts to overcome the shortcomings of single-modal biometrics in recognition accuracy, robustness, and flexibility and provides richer and more reliable biometrics applications. At present, the multimodal biometric system mainly focuses on the fusion extraction of multimode features at different levels to provide a unified data manipulation interface at the application layer [23]. Mehrotra et al. [24] proposed a class of multimodal classification for relevance vector classifier, which combined incremental and granular learning, which could handle large-scale unbalanced datasets and achieve better performance in multimodal biometrics classification and evaluation. Abdolahi et al. [25] proposed a multimodal fusion system using fingerprint and iris with fuzzy logic, and the obvious improvement in recognition rate was achieved. However, this method does not give a quantitative analysis of the effectiveness of the fusion process, and the obtained effect is poorly generalized. Miao et al. [26] proposed a framework of bin-based classifier method for the fusion of multibiometrics, which embedded matching scores into a new image pixel space, and obtained richer feature information when performing image-based biometrics. Chen et al. [27] proposed a framework for face and fingerprint images fusion using a type of middle-layer semantic features extracted from local feature-image matrix. However, it is still not clear whether this feature has good feature expression for all kinds of biometrics. Khellat et al. [28] proposed a feature level fusion method for three biological traits, which mainly used the Fisher dimensionality reduction technique, which caused the occurrence of the feature fusion in the dimensionality reduction space. Mai et al. [29] proposed a binary feature fusion method, which was generated from the sequence of feature bits using a machine learning algorithm that minimizes intraclass differences by minimizing interclass differences. The above proposed feature fusion algorithms still lack effective theoretical proof. Some work has been done on the application of multimodal biometrics. Liu et al. [30] applied the multimodal biometrics authentication method to single difficult biometrics, fused the different feature modalities to recognize the short utterance speaker, and achieved remarkable performance improvement. Gomez et al. [31] studied the protection of multibiometric template and proposed a multibiometric template protection technology based on homomorphic probability encryption. Gurusamy et al. [32] studied the biometric characteristics of MRI, and it was found that wavelet transform could better highlight the features of MRI images. Meng et al. [33] proposed a method for effectively detecting image hidden information by combining various image features through a fast R-CNN network. At present, the interpretability of the R-CNN network is insufficient.

To the best of our knowledge, although there are quite a few studies on multimodal biometrics, the definitive demonstration of multimode biometrics fusion has not been discussed yet. In this paper, we conducted a study on the deterministic effect of multimodal biometrics, using fingerprints and voiceprint as a template. At present, the research on the fusion of these two types of biometrics is still very limited.

3. Preliminary Research

Fingerprints, irises, human faces, voiceprints, and finger veins are the most commonly used biological features in human biometrics. The samples collection of fingerprint, voiceprint, and face is more convenient, and the application rate is higher. However, the face needs a large number of face sample banks, and the training and operation cost is high. In this paper, we use the low-cost fingerprint and voiceprint characteristics as the research objects.

3.1. Fingerprint Authentication Technology. The fingerprint authentication process is mainly divided into 4 steps, as shown in Figure 1: (1) acquisition of fingerprint images, usually using optical instruments and other equipment; (2) fingerprint image preprocessing, which finally gets the fingerprint thinning map; (3) fingerprint feature extraction, which extracts the fingerprint feature point information which is serialized as fingerprint feature vector and stored as fingerprint feature template; (4) fingerprint matching, which matches the extracted fingerprint feature vector with the feature template in the fingerprint database to confirm the authenticator.

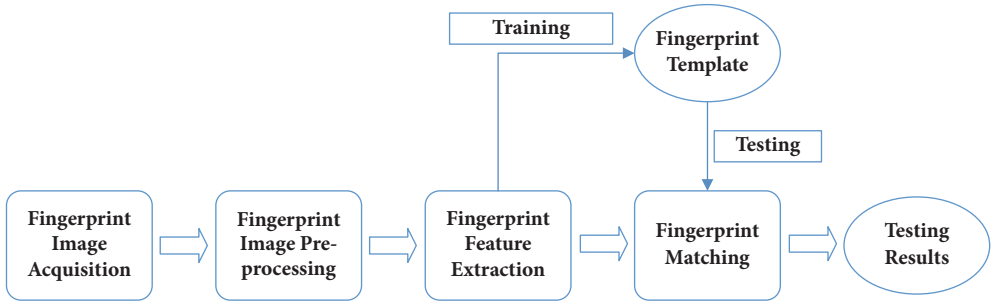


Figure 1: Fingerprint authentication process flowchart.

In general, fingerprint image authentication mainly depends on the uniqueness of individual fingerprints in the texture. Fingerprint image preprocessing is the process of removing noise and highlighting and clarifying fingerprint texture. The general process is shown in Figure 2. Fingerprint image preprocessing directly affects the performance of the entire fingerprint authentication system, and its mainly steps include the following: (1) fingerprint image enhancement: in this step, a specific algorithm such as frequency domain transform, filtering, denoising, and splicing of small block fingerprints is used to improve the quality of the image so that the fingerprint lines can have better connectivity and clearness, avoid false feature points, and improve fingerprinting characteristics accuracy of extraction. (2) Fingerprint image binarization: in this step, the fingerprint image is converted into a black-and-white binary image by the method of deleting the local image pixel point while maintaining the connectivity. As a result, the adhesion between the lines is removed, the complexity of the fingerprint feature extraction is reduced, and the subsequent image thinning operation is facilitated. (3) Fingerprint image refinement: in this step, the binarized fingerprint texture is refined into single-pixel lines, preserving the trend of the fingerprint lines without regard to the thickness of the lines. The refined fingerprint can extract details such as feature points more conveniently, so as to improve the accuracy of fingerprint matching.

Fingerprint feature extraction is a key step in the fingerprint template generation. Its main task is to obtain fingerprint feature point information. In the fingerprint identification process, the fingerprint feature point information is generally used as its main feature information, including the attribute point, position, and direction field value of the feature point. The comparison process determines whether the two feature points are the same according to the feature information. When two fingerprints have a certain number of the same feature points, the two fingerprints can be considered as one. The extracted feature point information is serialized to obtain a biometric template.

3.2. Voiceprint Authentication Technology. A complete speaker recognition system consists of acoustic feature extraction, voiceprint models establishing, and voiceprint matching calculations, as shown in Figure 3. The process of feature extraction is to extract the acoustic features of speech, such as Mel-scale Frequency Cepstral Coefficients (MFCC) from the original waveform signal, and to obtain a voiceprint model, such as Gaussian Mixture Model (GMM), which is used as a template to identify personal speech features. By calculating the voiceprint matching score, the system outputs the speaker authentication result.

The GMM is the most common voiceprint recognition model of the existing voiceprint models, as shown in Figure 4. The basic process is to extract the speech MFCC feature sequence and use the training data to calculate the model parameters and obtain the individual GMM template. The specific process is as follows.

For any D-dimensional vector x_t, $t = 1, 2, \ldots, M$, the Gaussian mixture probability density function used to calculate the likelihood is as follows:

$$p(x_t \mid \lambda) = \sum_{i=1}^{M} \omega_i b_i(x_t), \tag{1}$$

where ω_i is the $i-$th Gaussian component weight, satisfying $\sum_{i=1}^{M} \omega_i = 1$. Speaker model $\lambda = \{\omega_i \ \mu_i \ \Sigma_i\}$, where Σ_i is usually a diagonal matrix. And M is the number of Gaussian components; that is, the $i-$th mixed Gaussian probability density $b_i(x_t)$ is defined as follows:

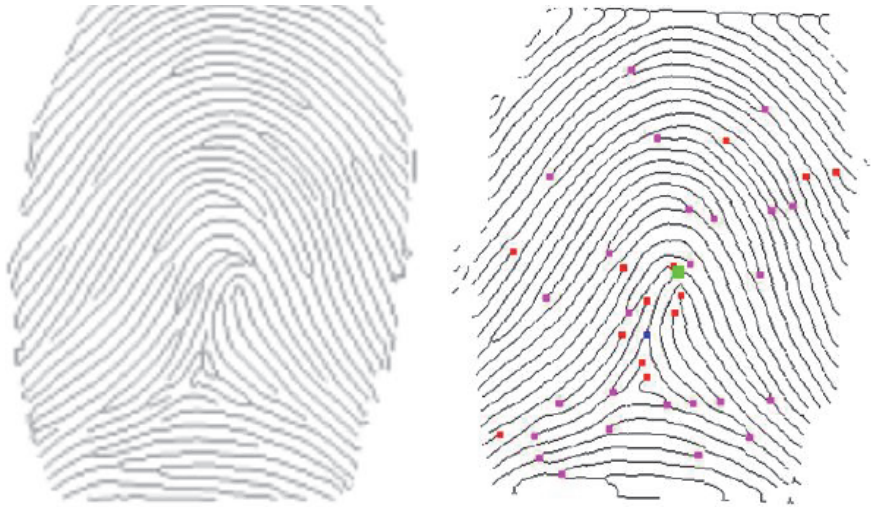
$$b_i(x_t) = \frac{1}{(2\pi)^{D/2} \left|\sum_i\right|^{1/2}} \cdot \exp\left\{-\frac{1}{2}(x_t - u_i)' \left(\sum_i\right)^{-1} (x_t - u_i)\right\}. \tag{2}$$

If $X = \{x_1, x_2, \ldots, x_m\}$ is the acoustic feature vector set of speaker I training and x_t is a D-dimensional vector, then the whole process of parameter estimation can be described as updating the model parameter λ^* satisfying $p(X \mid \lambda^*) \geq p(X \mid \lambda)$ iteratively until convergence. Given the trained feature vector set of speaker I, the model parameters are usually obtained by the EM iterative algorithm, and the iteration process is as follows:

(1) Weight iterative formula is as follows:

$$\omega_i = \frac{1}{T} \sum_{k=1}^{N} P(i \mid X_k, \lambda). \tag{3}$$

(a) Primitive image (b) Enhanced image (c) Binary image



(d) Thinning image (e) Feature point image

Figure 2: Fingerprint authentication process flowchart.

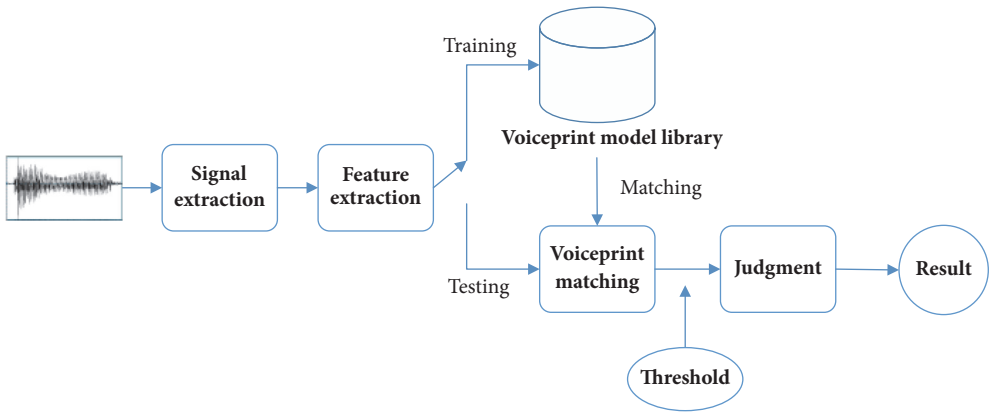


Figure 3: Speaker recognition system structure.

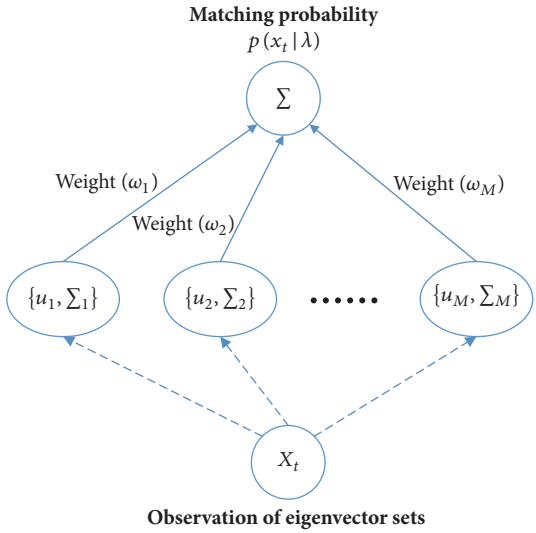


FIGURE 4: Fingerprint authentication process flowchart.

(2) Mean iterative formula is as follows:

$$u_i = \frac{\sum_{k=1}^{N} P(i \mid X_k, \lambda) \cdot X_k}{\sum_{k=1}^{N} P(i \mid X_k, \lambda)} \tag{4}$$

(3) Variance iterative formula is as follows:

$$\sum_i = \frac{\sum_{k=1}^{N} P(i \mid X_k, \lambda) \cdot X_k^2}{\sum_{k=1}^{N} P(i \mid X_k, \lambda)} \tag{5}$$

In the above formula, the posterior probability of component i is as follows:

$$P(i \mid X_k, \lambda) = \frac{\omega_i \cdot b_i(X_k)}{\sum_{t=1}^{M} \omega_t \cdot b_t(X_k)}. \tag{6}$$

The GMM parameters ω_i, u_i, $\sum_i$, etc. constitute a voiceprint biometric template.

Fingerprint and voiceprint features have their own characteristics. The fingerprint features are presented in the form of images. The features are hidden in the image texture. The recognition process requires fine processing of the image texture, which is susceptible to contamination and other kinds of interference and reduces the recognition accuracy. The high recognition accuracy of general fingerprints requires the assistance of a high-quality fingerprint collector. Speech frequency spectrum is the main analysis object of voiceprint feature. Fine processing of frequency domain features is needed in the process of recognition. The specificity of the features is not intuitive, but the anti-interference ability is slightly stronger than the fingerprint. The effective fusion of the two types of features can complement each other and enhance the anti-interference ability of feature recognition.

4. Multibiometric Fusion Authentication Algorithm MFDB-Decision

Multimodal biometric authentication based on image feature fusion generally achieves the ideal recognition accuracy in the limited sample test. However, the validity of the algorithm often lacks theoretical proof, making the generalization of the algorithm questionable. In this chapter, a demonstrable multimode fusion algorithm is derived from the combination of matching level and decision level making.

4.1. Strategy for Multimodal Fusion Optimizing. Multimodal biometric system uses various levels of fusion to combine two or more modalities [23], according to the different levels of integration. From low to high, it can be divided into the following:

(1) At the sensor layer, the captured images are pixel-level fused. It is worth paying attention to that retaining as much as information as possible is inefficient and has poor real-time performance due to the large amount of sensor data processed. Furthermore, considering the differences in signal acquisition equipment, sensor layer fusion is not feasible in most cases.

(2) At the feature extraction level, two or more modalities in the form of feature vectors are concatenated. Such a fusion often leads to very high dimensional vectors. But at present, the selection of characteristics is more random, and the specificity of the selected features generally lacks large-scale test.

(3) At the matching score level, it mainly combines the matching scores from different modalities. But for the two modes of pattern with different calculation methods, fusion will be difficult.

(4) At the decision level, the judgments of multiple verdicts are consolidated, and it has little requirement on the data relevance.

Through the abovementioned analysis, at present, although feature layer fusion may produce new effective features, it is difficult to guarantee the stability and reliability of new features. Considering that single-mode biometrics can easily excavate stable and specific features, we use the dynamic Bayesian method to combine feature recognition in the score layer and that in the decision layer based on fingerprint and voiceprint single-mode feature extraction, and the higher recognition accuracy and stability are obtained.

4.2. Dynamic Bayesian Decision for Minimizing the Error Risk. Because Bayesian judgments can achieve high judgment accuracy in mutually independent biometric modalities, we use Bayesian decision theory [34, 35] as the underlying mechanism. In the ideal case where all the relevant probabilities are known, Bayesian decision theory considers how to choose the optimal category marker based on these probabilities. Firstly, we suppose there are N possible category collections that can be shown like $Y = \{c_1, c_2, \ldots, c_n\}$, and λ_{ij} is the loss of classifying a sample of true labeled c_j as c_i. Based on the posterior probability $P(c_i \mid x)$, the expected loss produced by

classifying the sample x as c_i ("conditional error risk" on the sample x) can be expressed as follows:

$$R(c_i \mid x) = \sum_{j=1}^{N} \lambda_{ij} P(c_j \mid x). \tag{7}$$

Our task is to find a decision criterion $h : X \longrightarrow Y$ to minimize the cost of the error risk:

$$R(h) = E_x [R(h(x) \mid x)]. \tag{8}$$

Obviously, for each sample x, if h can minimize the conditional risk $R(h(x) \mid x)$, the overall cost of the error risk $R(h)$ will also be minimized. This produces dynamic Bayesian decision rule: To minimize the overall risk, it is needed to choose on each sample a category marker that minimizes the conditional risk $R(c \mid x)$; it can be written as follows:

$$h^* = \arg\min_{c \in Y} R(c \mid x), \tag{9}$$

where h^* means Bayesian optimal classifier, corresponding to the overall risk $R(h^*)$ called Bayesian risk, and $1 - R(h^*)$ reflects the notion that the classifier can achieve the best performance. When it comes to classification issues, λ_{ij} can be expressed as

$$\lambda_{ij} = \begin{cases} 0 & if\ i = j \\ 1 & otherwise. \end{cases} \tag{10}$$

Thus, the Bayesian optimal classifier that minimizes the classification error rate is

$$h^*(x) = \arg\max_{c \in Y} P(c \mid x). \tag{11}$$

It is obviously observed that the significance of maximizing posterior probability is to minimize the expected risk.

4.3. Multibiometric Fusion Authentication using Dynamic Bayesian Decision (MFDB-Decision). In order to fuse the fingerprints and voiceprint recognition systems together, a score vector $X = (X_1, X_2)$ containing multiple recognition system is constructed, where X_1 and X_2, respectively, represent the scores obtained from the fingerprint and voiceprint recognition system. Then, the question of identity conversion translates into the problem of classifying the two-dimensional fraction vector $X = (X_1, X_2)$ as accepting (genuine) or rejecting (imposter).

We know each modal classifier should have a different weight in multimodal classifier; in this paper, we should not fix the weight of biometrics because the dominant biometric is not always the same one. So we propose an algorithm called dynamic Bayesian decision (MFDB-decision) to get the best fusion recognition accuracy. Algorithm 1 is described in detail as follows.

The output of the algorithm is the category to which this feature belongs. The matching scores from each of the two classifiers $X = (X_1, X_2)$, $X_1 = P(x_{ij} \mid R_1)$, $X_2 = P(y_{ij} \mid R_2)$, $i = 1, 2, \ldots, N$, $j = 1, \ldots, k$; N means a total of N categories,

Input: $X = (X_1, X_2)$, $X_1 = P(x_{ij} \mid R_1)$,
$X_2 = P(y_{ij} \mid R_2)$, $i = 1, 2, \ldots, N$, $j = 1, \ldots, k$
(1) **repeat**
$fs \longleftarrow \max_{i=1,\ldots,N} P(x_{ij} \mid R_1)$
$fn \longleftarrow \arg\max_{i=1,2,\ldots N} P(x_{ij} \mid R_1)$
$vs \longleftarrow \max_{i=1,\ldots,N} P(y_{ij} \mid R_2)$
$vn \longleftarrow \arg\max_{i=1,2,\ldots N} P(y_{ij} \mid R_2)$
(2) **if** $fs < \sigma_1$ && $vs < \sigma_2$ **then**
(3) $n = false$
(4) **else if** $fs \geq \sigma_1$ && $vs < \sigma_2$ **then**
(5) $n = \arg_i \left(\max_{i=1,2,\ldots N} \sum_{m=1}^{2} \alpha_m vote_{im} \right), \quad \alpha_1 > \alpha_2$
(6) **else if** $fs < \sigma_1$ && $vs \geq \sigma_2$ **then**
(7) $n = \arg_i \left(\max_{i=1,2,\ldots N} \sum_{m=1}^{2} \alpha_m vote_{im} \right), \quad \alpha_2 \geq \alpha_1$
(8) **end if** {n is category judgment result}
(9) **until** the (k+1)-th test sample

Algorithm 1: The MFDB-decision algorithm.

and k represents a total of k test samples. α_1 is the weight of fingerprint recognition system; α_2 is the weight of voiceprint recognition system; σ_m is the quality failure threshold. Here, we set σ_2 to be the average score value of current person with N template and cycling σ_1 from 0.3 to 0.6, step 0.01. In step 5, the algorithm uses (10) to calculate $vote_{im}$. In step 7, because the voiceprint quality is high, the algorithm sets the weight of voiceprint α_2 to be greater than α_1.

4.3.1. Fingerprint Matching Score Calculation. Fingerprint matching algorithm is mainly divided into three kinds of schemes based on correlation, minutiae, and nonminutiae matching. Either of the schemes will firstly form a fingerprint feature vector template, which is authenticated by a template. After extracting the feature vector for two fingerprint images, we can express it as

$$\begin{aligned} A &= \left\{ m_{A_1}, m_{A_2}, \ldots, m_{A_p} \right\}, \\ &\text{where } m_{A_i} = \left\{ x_{A_i}, y_{A_i}, cn, \theta_{A_i} \right\},\ 1 \leq i \leq p \\ B &= \left\{ \mathrm{m}_{B_1}, \mathrm{m}_{B_2}, \ldots, \mathrm{m}_{B_p} \right\}, \\ &\text{where } m_{B_1} = \left\{ x_{B_j}, y_{B_j}, cn, \theta_{B_j} \right\},\ 1 \leq j \leq q. \end{aligned} \tag{12}$$

There are two finite sets of points in space: $A = \{x_1, y_1, cn, \theta_1\}$ and $B = \{x_2, y_2, cn, \theta_2\}$, where x and y represent the coordinates of the detail points, respectively, where cn denotes the type of the detail point, for example, $cn = 3$ for a fork, $cn = 1$ for an endpoint, and so on. θ denotes the direction along the main ridge. As fingerprints are collected during being pressed, it is easy for the collected ones to be offset. Therefore, in the authentication process, the geometric constraints on the details of the matching point are

proposed, including the geometric distance and the angle of detail deviation from the limit as follows:

$$dist_r\left(m_{A_i}, m_{B_j}\right) = \sqrt{\left(x_{A_i} - x_{B_j}\right)^2 + \left(y_{A_i} - y_{B_j}\right)^2} < r_\delta \tag{13}$$

$$dist_\theta\left(m_{A_i}, m_{B_j}\right) = \min\left(\left|\theta_{A_i} - \theta_{B_j}\right|, 360 - \left|\theta_{A_i} - \theta_{B_j}\right|\right) < r_\theta. \tag{14}$$

Following global registration, a local search can be performed [36], where r_δ means a reasonable distance threshold for the offset of the minutia and r_θ is a permissible deviation from the distortion estimate obtained from the ridge pattern. At the same time, two characteristic points satisfying formula (13) and (14) are considered as matching feature points. Two fingerprints with enough matching feature points are considered to be matched fingerprints. The specific score of fingerprint matching can be calculated as follows:

$$sim(A, B) = \frac{n_{match}^2}{n_A n_B} \tag{15}$$

where n_{match}^2 means the number of feature points that match within the threshold in both graphs. n_A and n_B are the number of feature points, respectively, owned by the template vector and the test vector.

4.3.2. Voiceprint Matching Score Calculation. For D-dimensional acoustic feature vector x_k, $k = 1, 2, \ldots, M$, the Gaussian mixture probability density function used to calculate the likelihood is as shown in (1). The whole process of likelihood parameter estimation can be described as updating the model parameter λ^* satisfying $p(X \mid \lambda^*) \geq p(X \mid \lambda)$ iteratively until convergence. According to Jensen inequality, the problem of parameter solving can be transformed into the problem of maximized $Q(\lambda, \lambda^*)$, and $Q(\lambda, \lambda^*)$ can be solved as follows due to $p(x_k, i \mid \lambda) = \omega_i p(x_k \mid \lambda, i)$:

$$Q(\lambda, \lambda^*) = \sum_{k=1}^{m}\sum_{i=1}^{M}\frac{\omega_i p(x_k \mid \lambda, i)}{p(x_k \mid \lambda)}\left[\lg \omega_i^* + \lg p(x_k \mid \lambda^*, i)\right] \tag{16}$$

Calculate the partial derivatives of mean value, weight, and covariance, respectively, and let the result be zero, and then an updated formula of the model parameters ω_i, μ_i, and Σ_i will be obtained. Given the trained feature vectors set of speaker I, the model parameters are usually obtained by the EM iterative algorithm, but the computational complexity is large. Therefore, this paper adopts the adaptive method proposed by Reynolds to solve the model parameters. Using the set of observed feature vectors of speaker I to fit the predictive speaker model through the maximum posteriori probability (MAP), the problem is actually transformed into an optimization problem. Similar to the E step in the EM algorithm, the sufficient statistics n_i, $E_i(x)$, $E_i(xx')$, and $p(i \mid x_k)$ of each Gaussian component of the UBM are first calculated, the difference being that the weight of the speaker model, the mean value, and covariance of the update process at the M step are as follows:

$$\omega_i^* = \left[\frac{\alpha_i^\omega n_i}{m} + \left(1 - \alpha_i^\omega\right)\omega_i\right]\zeta \tag{17}$$

$$\mu_i^* = \alpha_i^\omega E_i(x) + \left(1 - \alpha_i^\omega\right)\mu_i \tag{18}$$

$$\left(\sigma_i^*\right)^2 = \alpha_i^\upsilon E_i\left(xx'\right) + \left(1 - \alpha_i^\upsilon\right)\left(\sigma_i^2 + \mu_i\mu_i'\right) - \left(\mu_i^*\right)^2. \tag{19}$$

In the formula, $\alpha_i^\rho = n_i/(n_i + r^\rho)$, $\rho \in \{\omega, m, \upsilon\}$ is an adaptive parameter that controls the change between the old and new coefficients, where $r^\rho = 16$ is a fixed correlation factor. ζ makes sure the weight rollup is always 1. Usually, only update the mean value, that is, $\alpha_i^\omega = \alpha_i^\upsilon = 0$. If the test speech feature vectors set of speaker J (declared as the I-th speaker λ_1, abbreviated as λ) is χ, the common background model is λ_u, and the system logarithmic likelihood score is

$$\Lambda = \lg p(\chi \mid \lambda) - \lg p(\chi \mid \lambda_u). \tag{20}$$

4.4. Theoretical Support for the MFDB-Decision Algorithm. Algorithm 1 leads to the following lemma.

Lemma 1. *The MFDB-decision algorithm can be generalized to* L *classifiers when each one is independent.*

Proof. Assuming there are x_i $(i = 1, \ldots, m)$ samples to be identified and entered into L different classifiers, the output of a certain sample to be recognized after passing L classifiers is R_j $(j = 1, 2, \ldots, L)$, and x_i must be able to be identified as one of the N classes. According to the Bayes decision theory for minimizing the risk of loss, the fusing sample to be identified will be recognized as the highest posterior probability in N modal classes, and (11) can be written as

$$n = \arg_i\left(\max_{i=1,2,\ldots,N} P(x_i \mid R_1, R_2, \ldots, R_L)\right). \tag{21}$$

We assume that L classifier is independent. So (6) can be analyzed as follows:

$$P(x_i \mid R_1, R_2, \ldots, R_L) = \prod_{l=1}^{L} P(x_i \mid R_l). \tag{22}$$

We derive the following from bringing (22) into (21):

$$n = \arg_i\left(\max_{i=1,2,\ldots,N}\prod_{l=1}^{L} P(x_i \mid R_l)\right). \tag{23}$$

If it is assumed that the posterior probability $P(x_i \mid R_l)$ fluctuates above and below the prior probability $P(x_i)$ and is not large, just shown as

$$P(x_i \mid R_l) = P(x_i)(1 + \delta_{il}) \quad \delta_{il} \ll 1, \tag{24}$$

take formula (24) into (22) as follows:

$$\begin{aligned}\prod_{l=1}^{L} P(x_i \mid R_l) &= P(x_i) \prod_{l=1}^{L} (1 + \delta_{il}) \\ &\approx P(x_i) + P(x_i) \sum_{l=1}^{L} \delta_{il} \\ &= P(x_i) + P(x_i) \sum_{l=1}^{L} (P(x_i \mid R_l) - P(x_i)) \\ &= P(x_i) \left(1 - LP(x_i) + \sum_{l=1}^{L} P(x_i \mid R_l)\right).\end{aligned} \tag{25}$$

Approximately equal sign uses the Taylor series expansion, and we get a general formalized multiclassifier fusion strategy which holds for each independent feature classifier as follows:

$$n = \arg_i \left(\max_{i=1,2,\ldots,N} \sum_{l=1}^{L} P(x_i \mid R_l) \right). \tag{26}$$

Consider fusing the concept of minimum loss expressed by (11) and (26) and if each classifier can find the maximum posterior probability, we can find the best posterior probability comprehensively:

$$\begin{aligned} vote_{Kl} &= \begin{cases} 1, & K = \arg_i \left(\max_{i=1,2,\ldots,N} (P(x_i \mid R_l)) \right) \\ 0, & otherwise \end{cases} \\ n &= \arg_k \left(\max_{k=1,2,\ldots,N} \sum_{l=1}^{L} vote_{kl} \right). \end{aligned} \tag{27}$$

Under the premise that all L classifiers are correct, the fusion classification result of (27) is guaranteed. However, the abovementioned formula does not consider the influence of the classifier posterior probability $P(x_i \mid R_l)$ on the classification result. If some of the preceding classifications are wrong, the error probability will be accumulated and transmitted backwards, resulting in low robustness. In order to reduce the influence of $P(x_i \mid R_l)$ on fusion classification results, fractional layer information was added to assist in judgment which means the weight α_m needs to be dynamically adjusted in each fusing round and mainly rely on scores that the current user obtained at the matching layer. When the maximum posteriori is within the qualified threshold, then the quality is judged to be good and a larger weight a is assigned. If it is outside the qualified threshold, then the quality is judged to be poor and a smaller weight b is assigned. When a suitable threshold is found as the decision key, our algorithm using matching score to assist in making decision can be modified from formula (27) as

$$\begin{aligned} \alpha_l &= \begin{cases} a & \max_{k=1,2,\ldots,N} (P(x_k \mid R_l)) > \sigma_i \\ b & otherwise,\ b < a; \end{cases} \\ n &= \arg_k \left(\max_{k=1,2,\ldots,N} \sum_{l=1}^{L} \alpha_l vote_{kl} \right), \end{aligned} \tag{28}$$

where α_m is a dynamic weight based on the overall performance of each single-mode which can be judged from the matching layer. a and b are positive numbers that satisfy $a + b = 1$ and $b < a$. σ_i is the quality threshold in which we usually set the average score value of current person with N templates. The setting of details in this paper can be seen in Section 4.3. In this way, when an error is accumulated, the quality of the poor quality feature can be reduced to influence the final result of the voting, thereby improving the robustness of the algorithm. According to formula (28), the lemma is proved. □

Corollary 2. *If the L classifiers are independent and the number of categories is fixed, then the error rate of the MFDB-decision algorithm will be infinitely tending to zero when the number of classifiers tends to infinity.*

Proof. Assuming that L classifiers are independent and our MFDB-decision algorithm achieved the minimum error rate in Lemma 1, then the probability of classification error for each classifier is less than the randomly selected error rate for N categories, i.e., $(N-1)/N$. So it can be expressed as follows:

$$\min er \le \left(\frac{N-1}{N} \right), \tag{29}$$

where er represents the error rate of each classifier. When the number of categories N is fixed and the number of classifiers L increases to infinity, the classification error rate can be expressed as

$$\begin{aligned} 0 &\le \lim_{L \longrightarrow +\infty} (\min er)^L \le \lim_{L \longrightarrow +\infty} \left(\frac{N-1}{N} \right)^L \\ &= \lim_{L \longrightarrow +\infty} \left(1 - \frac{1}{N} \right)^L = 0. \end{aligned} \tag{30}$$

Since N is a fixed value, $1 - 1/N$ is a constant less than 1, so the abovementioned equation $\lim_{L \longrightarrow +\infty} (\min er)^L$ equals 0 according to Sandwich Theorem. Therefore, as the number of classifiers L increases, the classification error will be reduced to 0. □

This section not only proves the feasibility of MFDB-decision theoretically in this paper but also lists a situation showing that the classification error rate will decrease with the increase of classifiers. Therefore, the proposed algorithm is suitable for generalization of multimodal recognition beyond bimodality.

5. Experimental Results and Analysis

In order to test the effectiveness and practicability of the MFDB-decision algorithm, we used 3 common databases and 1 self-extracting database to conduct experimental tests. The three common databases were FVC2002 DB1 database (fingerprint), MIT Media lab Speech Dataset (speech), and TIMIT corpus (speech). The self-extracting database was hdu2016_40 database (short speech). FVC2002 DB1 database was a standard difficult fingerprint dataset with 100 fingers

and eight samples for each finger, which was provided by the National Institute of Standards and Technology (NIST). The recognition rate of difficult fingerprints by general algorithms was not high. Generally, if there was no optimization, the recognition rate would be lower than 95%. In the fingerprint recognition competition, the participants could increase the recognition rate to over 99% by specific optimization, but the versatility of such algorithms was not strong. Since this test mainly investigates the effects of the two types of biometric fusion, the high accuracy of single-mode feature recognition was not conducive to the test results. Therefore, all the following test procedures used the universal fingerprint recognition algorithm and did not specialize optimization for the FVC database. The MIT Media lab Speech Dataset consisted of 48 registrars (22 females and 26 males) and 40 attackers (17 females and 23 males). Recorded separately in the handheld microphone and external headphones, the recording environment included quiet indoor, slightly interfering laboratories and noisy intersections; each tester randomly read 108 words or sentences in 6 environments. The TIMIT corpus was designed by the Defense Advanced Research Projects Agency. The number of registrations in TIMIT was 630, and each person read 10 sentences with 6300 sentences. 630 people were made up of 8 regions, including 438 men and 192 women. Each person read two designated text phonetics (SA) in dialect, five phonetically compact sentences (SX), and three phonetically diverse sentences (SI). The hdu2016_40 was a corpus including 40 people, each person had 25 paragraphs of different short utterance, each paragraph would be recorded ten times, and each record lasted 2~3 seconds. The MIT Speech Dataset and the TIMIT corpus were both English datasets, and the hdu2016_40 database was short speech Chinese datasets. Although, in the field of long speech voiceprint recognition, the recognition accuracy rate had reached 98% in low noise environment, in the short speech voiceprint recognition environment (voice length less than 5s), even if the ambient noise was low, the recognition accuracy was still not ideal, less than 95%. This article tested for short speech voiceprint recognition. Because the common fingerprint and voiceprint from the same tester database were relatively rare, the experiment used the abovementioned four groups of database combination for testing. In order to made the experimental results more reliable, the samples in the different databases were randomly selected during the experiment for combination testing, and multiple random sampling was performed under the condition that the combinations were not repeated. To evaluate the performance, false rejection rate (FRR) and false acceptance rate (FAR) are used as the main indicators.

5.1. Single Modal and Multimodal Comparison. In order to investigate the effectiveness of the MFDB-decision algorithm, we firstly examine the improvement of the accuracy of the MFDB-decision algorithm when the single-mode algorithm has difficulty in achieving high accuracy. The abovementioned four databases constitute two sets of datasets, with one being the combination of FVC2002 DB1 database, the MIT speech database, and the TIMIT corpus and the other being the combination of the FVC2002 DB1 database and the hdu2016_40 short speech database. The TIMIT corpus was used to train the English speech Universal Background Model (UBM), and the Chinese UBM was trained with the network random grasp of Chinese speech set. In the experiment, the fingerprint recognition algorithm used a general feature-based matching algorithm. The voiceprint recognition algorithm used the GMM model. Fingerprint and voiceprint recognition algorithms had no additional targeted optimization measures, so the difficulties of fingerprint recognition rate and the short speech voiceprint recognition rate were not high, as shown in Table 1.

Table 1: Difference between single biometric and multimodal biometric (%).

Trait	FAR	FRR	Accuracy
Fingerprint	4.8	5.0	94.60
Voiceprint(#1)	7.0	7.0	93.12
Voiceprint(#2)	11.5	11.7	88.38
MFDB-decision(#1)	1.0	1.2	99.06
MFDB-decision(#2)	1.6	1.8	98.35

Table 1 shows the experimental differences between single-mode and multimodal states. The '#1' and '#2', respectively, represent the collection of two types of voiceprint test databases in Chinese and English. Since the Chinese database is a self-acquisition voice database, the voice quality is better, so the recognition accuracy is relatively high. The result shows the superiority of our algorithm compared to single-mode recognition. We use the method proposed in Section 4.3.1 in the process of the fingerprint matching, using formulae (13) and (14) calculating the geometric distance and the angle of detail deviation; here, we set $r_\delta = 10$ and $r_\theta = 9$ and get similarity score calculated by (15). We performed the voiceprint process using the method proposed in Section 4.3.2, and 128 mixtures are used by GMM model, and the likelihood score is calculated by using formula (20). From the results, the MFDB-decision algorithm merged two single modes and achieved more stable and accurate results.

5.2. Robustness of the MFDB-Decision Algorithm. The fingerprints of 100 people (FVC2002 DB1, totaling 100 training pieces and 700 testing pieces) were divided into 60 groups, with each group consisting of 1 40 people, 2~41 people, 3~42 people, etc. There were 40 individuals in each group, each with 1 training fingerprint, 7 testing fingerprints, 25×5 training voices, and 25×5 test voices, for a total of 40×7×25×5=35,000 tests. Each fingerprint and each voiceprint were paired and the average of 35,000 tests was taken as the test result of this group. We plotted the recognition rate (Genuine Acceptance Rate (GAR)) of each group after using the MFDB-decision algorithm in Figure 5, where "1~10" in the legend indicated the recognition rate obtained after the first to tenth groups of voiceprints and fingerprints were fused.

It can be seen from Figures 5 and 6 that the fusion model shows high stability as the recognition rate concentrated in 97% to 100%. And the fusion recognition rate is increased by

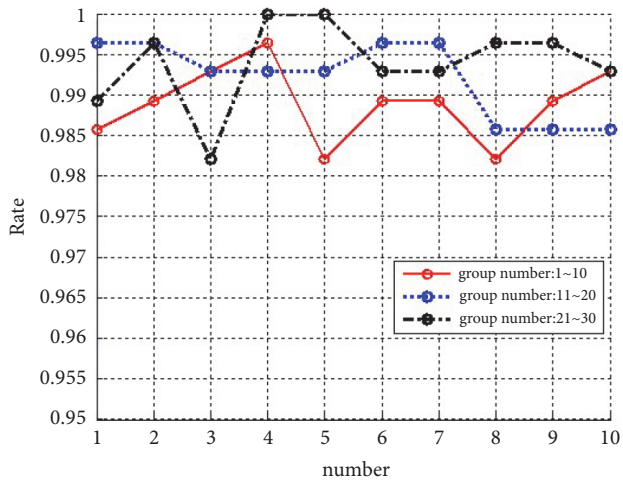


Figure 5: The recognition rate (GAR) of voiceprint fusing with fingerprint Group (a).

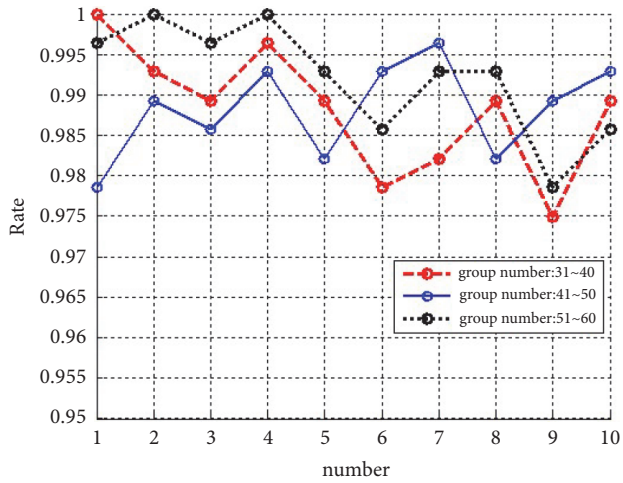


Figure 6: The recognition rate (GAR) of voiceprint fusing with fingerprint Group (b).

4.46% and 5.96% compared with single-model of fingerprint and voiceprint, respectively.

5.3. Effectiveness of the MFDB-Decision Algorithm. We test the effectiveness of the MFDB-decision algorithm by comparing the recognition rate of the MFDB-decision algorithm with several general fusion algorithms. The experimental process used the same grouping method in Section 5.2. The fusion recognition rate of each group was calculated by different algorithms, and the average recognition rate of each group was used as the final recognition rate of the fusion algorithm. The averaging in the abovementioned process was advantageous to avoid contingency. We randomly plotted 10 sets of recognition rates and compared them with the other two methods (AND as well as fixed weight voting method [37]). As shown in Figure 7, it can be seen that the MFDB-decision algorithm not only achieves a high recognition rate but also obtains good stability. The recognition rate of the

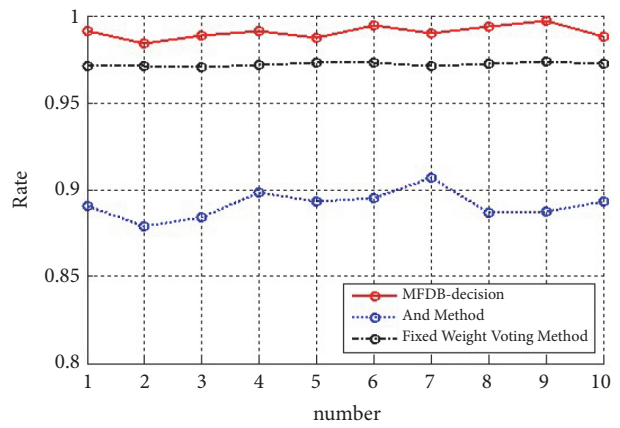


Figure 7: The recognition rates (GAR) of the three methods.

Table 2: Recognition rate of methods (%).

Method	Accuracy
And	89.10
Fixed weight Voting Method	97.21
MFDB-decision	99.06

MFDB-decision algorithm was higher than other algorithms. The fixed weight method was more stable than the MFDB-decision algorithm, but its recognition rate was not high enough. In many cases, it is worthwhile to sacrifice some stability and get a better recognition rate. Table 2 lists the average accuracy of the three multimodal methods.

MFDB-decision-making algorithm uses the score information of matching layer to assist decision-making recognition, which is helpful to recover the lost data in the decision-making process. Figure 8 shows the DET curves for various fusion methods. The PCA method uses the principal component analysis method mentioned in [38]. Since the unprocessed voiceprint MFCC sequence was not specific in the voiceprint recognition process, in this experiment, the accuracy of the PCA method was not high. The fuzzy rule method used fuzzy logic in the decision-making layer in [25] and achieved significant performance improvement. The trend of all curves is similar and decreases with the increase of FRR. The results show that all kinds of fusion methods are effective in the fusion of fingerprint and voiceprint, but our algorithm has achieved better results than other algorithms. We find that each curve intersects with the diagonal, indicating FRR=FAR, which is the equal error rate point EER. Generally, the lower the EER, the better the performance of the algorithm.

6. Conclusions

In this paper, we proposed a multimodal biometric recognition algorithm (named MFDB-decision) and demonstrated its effectiveness. We solved the problem that the fixed weight value could not be adaptively assigned in multimodal recognition and it would result in poor fusion performance. We compared the result of fusion with the result of single-modal

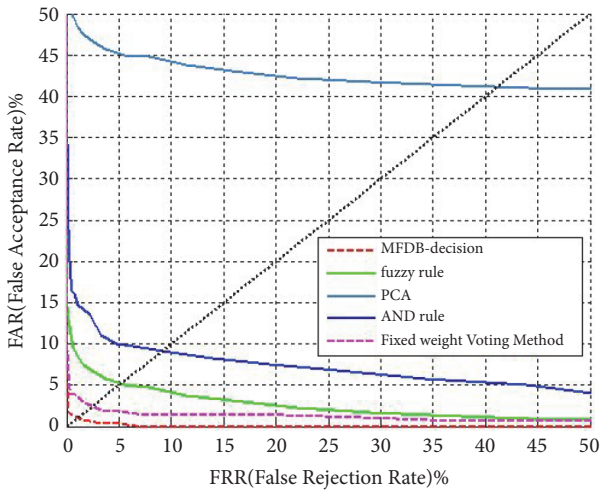


Figure 8: The DET curve for the methods.

recognition as well as the other methods and found that the method improved the recognition rate by an average of 5.0% or more. The multimodal fusion methods we developed are also greatly useful in the fusion recognition of other patterns. Future work will focus on multimodal biometric key extraction, ubiquitous identity authentication, and encryption technologies.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by National Key R&D Program of China (no. 2016YFB0800201), National Natural Science Foundation of China (no. 61772162), joint fund of National Natural Science Fund of China (no. U1709220), and Zhejiang Natural Science Foundation of China (no. LY16F020016).

References

[1] P. Wild, P. Radu, L. Chen, and J. Ferryman, "Robust multimodal face and fingerprint fusion in the presence of spoofing attacks," *Pattern Recognition*, vol. 50, pp. 17–25, 2016.

[2] V. Conti, C. Militello, F. Sorbello, and S. Vitabile, "A frequency-based approach for features fusion in fingerprint and iris multimodal biometric identification systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 4, pp. 384–395, 2010.

[3] A. Nagar, K. Nandakumar, and A. K. Jain, "Multibiometric cryptosystems based on feature-level fusion," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 255–268, 2012.

[4] R. Snelick, U. Uludag, A. Mink, M. Indovina, and A. Jain, "Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 450–455, 2005.

[5] A. Muthukumar, C. Kasthuri, and S. Kannan, "Multimodal biometric authentication using particle swarm optimization algorithm with fingerprint and iris," *ICTACT Journal on Image and Video Processing*, vol. 02, no. 03, pp. 369–374, 2012.

[6] S. Shekhar, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Joint sparse representation for robust multimodal biometrics recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 113–126, 2014.

[7] Z. Wu, B. Liang, L. You, Z. Jian, and J. Li, "High-dimension space projection-based biometric encryption for fingerprint with fuzzy minutia," *Soft Computing*, vol. 20, no. 12, pp. 4907–4918, 2016.

[8] Z. Wu, L. Tian, P. Li, T. Wu, M. Jiang, and C. Wu, "Generating stable biometric keys for flexible cloud computing authentication using finger vein," *Information Sciences*, vol. 433-434, pp. 1339–1351, 2018.

[9] G. Bajwa and R. Dantu, "Neurokey: Towards a new paradigm of cancelable biometrics-based key generation using electroencephalograms," *Computers & Security*, vol. 62, pp. 95–113, 2016.

[10] L. Jin, L. Sun, Q. Yan et al., "Significant permission identification for machine learning based android malware detection," *IEEE Transactions on Industrial Informatics*, p. 12, 2018.

[11] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and Traceable Group Data Sharing in Cloud Computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 912–925, 2018.

[12] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *Journal of Network and Computer Applications*, vol. 106, pp. 117–123, 2018.

[13] J. Shen, C. Wang, T. Li, X. Chen, X. Huang, and Z.-H. Zhan, "Secure data uploading scheme for a smart home system," *Information Sciences*, vol. 453, pp. 186–197, 2018.

[14] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, 2018.

[15] C. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, "Privacy-preserving Naive Bayes classifiers secure against the substitution-then-comparison attack," *Information Sciences*, vol. 444, pp. 72–88, 2018.

[16] Y. Li, G. Wang, L. Nie, Q. Wang, and W. Tan, "Distance metric optimization driven convolutional neural network for age invariant face recognition," *Pattern Recognition*, vol. 75, pp. 51–62, 2018.

[17] X. Zhang, Y. Tan, C. Liang, Y. Li, and J. Li, "A Covert Channel Over VoLTE via Adjusting Silence Periods," *IEEE Access*, vol. 6, pp. 9292–9302, 2018.

[18] J. Li, Q. Lin, C. Yu, X. Ren, and P. Li, "A QDCT- and SVD-based color image watermarking scheme using an optimized encrypted binary computer-generated hologram," *Soft Computing*, pp. 1–19, 2016.

[19] M. Gomez-Barrero, C. Rathgeb, J. Galbally, C. Busch, and J. Fierrez, "Unlinkable and irreversible biometric template protection based on bloom filters," *Information Sciences*, vol. 370/371, pp. 18–32, 2016.

[20] Y. Liu, J. Ling, Z. Liu, J. Shen, and C. Gao, "Finger vein secure biometric template generation based on deep learning," *Soft Computing*, vol. 21, no. 1, pp. 1–9, 2017.

[21] C. Yuan, X. Li, Q. M. J. Wu, J. Li, and X. Sun, "Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis," *Computers, Materials and Continua*, vol. 53, no. 4, pp. 357–371, 2017.

[22] W. Holden, "Securing public faith in biometrics," *Biometric Technology Today*, vol. 2016, no. 9, pp. 7–9, 2016.

[23] M. He, S.-J. Horng, P. Fan et al., "Performance evaluation of score level fusion in multimodal biometric systems," *Pattern Recognition*, vol. 43, no. 5, pp. 1789–1800, 2010.

[24] H. Mehrotra, R. Singh, M. Vatsa, and B. Majhi, "Incremental granular relevance vector machine: A case study in multimodal biometrics," *Pattern Recognition*, vol. 56, pp. 63–76, 2016.

[25] M. Abdolahi, M. Mohamadi, and M. Jafari, "Multimodal biometric system fusion using fingerprint and iris with fuzzy logic," in *International Journal of Soft Computing and Engineering*, vol. 2, pp. 504–510, 2013.

[26] D. Miao, M. Zhang, Z. Sun, T. Tan, and Z. He, "Bin-based classifier fusion of iris and face biometrics," *Neurocomputing*, vol. 224, pp. 105–118, 2017.

[27] Y. Chen, J. Yang, C. Wang, and N. Liu, "Multimodal biometrics recognition based on local fusion visual features and variational Bayesian extreme learning machine," *Expert Systems with Applications*, vol. 64, pp. 93–103, 2016.

[28] S. Khellat-Kihel, R. Abrishambaf, J. L. Monteiro, and M. Benyettou, "Multimodal fusion of the finger vein, fingerprint and the finger-knuckle-print using Kernel Fisher analysis," *Applied Soft Computing*, vol. 42, pp. 439–447, 2016.

[29] G. Mai, M.-H. Lim, and P. C. Yuen, "Binary feature fusion for discriminative and secure multi-biometric cryptosystems," *Image and Vision Computing*, vol. 58, pp. 254–265, 2017.

[30] Z. Liu, Z. Wu, T. Li, J. Li, and C. Shen, "GMM and CNN Hybrid Method for Short Utterance Speaker Recognition," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1-1, 2018.

[31] M. Gomez-Barrero, E. Maiorana, J. Galbally, P. Campisi, and J. Fierrez, "Multi-biometric template protection based on Homomorphic Encryption," *Pattern Recognition*, vol. 67, pp. 149–163, 2017.

[32] R. Gurusamy and V. Subramaniam, "A machine learning approach for MRI brain tumor classification," *Computers, Materials and Continua*, vol. 53, no. 2, pp. 91–109, 2017.

[33] R. Meng, G. S, J. Wang, and X. Sun, "A machine learning approach for mri brain tumor classification," *Computers Materials and Continua*, vol. 55, no. 1, p. 16, 2018.

[34] N. Ueffing and H. Ney, "Bayes decision rules and confidence measures for statistical machine translation," in *Advances in Natural Language Processing*, vol. 3230 of *Lecture Notes in Computer Science*, pp. 70–81, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[35] J.-T. Chien, C.-H. Huang, K. Shinoda, and S. Furui, "Towards optimal bayes decision for speech recognition," in *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006*, pp. I45–I48, France, May 2006.

[36] J. Abraham, P. Kwan, and G. Junbin, *Fingerprint Matching using A Hybrid Shape and Orientation Descriptor*, 2011.

[37] S. Lei and M. Qi, "Multimodal Recognition Method based on Ear and Profile Face Feature Fusion," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 1, pp. 33–42, 2016.

[38] C. Chibelushi, "Feature-level data fusion for bimodal person recognition," in *Proceedings of the 6th International Conference on Image Processing and its Applications*, pp. 399–403, Dublin, Ireland.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 2329891, 12 pages
https://doi.org/10.1155/2018/2329891

Research Article

An Improved Permission Management Scheme of Android Application Based on Machine Learning

Shaozhang Niu,[1] **Ruqiang Huang**,[1] **Wenbo Chen**,[1] **and Yiming Xue**[2]

[1]*Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China*

Correspondence should be addressed to Shaozhang Niu; szniu@bupt.edu.cn

Received 26 June 2018; Revised 25 September 2018; Accepted 2 October 2018; Published 18 October 2018

Guest Editor: Zhaoqing Pan

The Android permission mechanism prevents malicious application from accessing the mobile multimedia data and invoking the sensitive API. However, there are still lots of deficiencies in the current permission management, which results in the permission mechanism being unable to protect users' private data properly. In this paper, a dynamic management scheme of Android permission based on machine learning is proposed to solve the problem of the existing permission mechanism. In order to accomplish the dynamic management, the proposed scheme maintains a dynamic permission management database which records the state of permissions for each application. Only the permission which is granted state in the database can be used in this application. In the whole process, the scheme first classifies the application by means of machine learning, then retrieves the corresponding permission information from databases, and issues the dangerous permission warning to users. Finally, the scheme updates the dynamic management database according to the users' decisions. Through this scheme, users can prevent malicious behaviour of accessing private data and invoking sensitive API in time. The solution increases the flexibility of permission management and improves the security and reliability of multimedia data in Android devices.

1. Introduction

While smart devices bring us a lot of convenience, they also become the attractive targets of cyberattacks [1]. Multimedia data in mobile device features as large storage and speedy transmit of high-definition data, which increases its popularity among big data environment [2]. However, its security and privacy becomes a growing serious problem. Android ensures that all applications get access to the privacy data (such as contact list, photo album, and other private data) in a reasonable situation by means of requiring that applications should declare permissions in the configuration file. Thus, permission management is the straightest and most fundamental method to protect user privacy data. Permission mechanism plays a critical role in controlling access to the resources in the device.

However, there are still a number of problems in the permission mechanism: Android does not support dynamic configuration and customized management of the permissions. In addition, most developers of Apps do not follow "Least Privilege Principle" when they create function for their products. Felt AP et al. analysed nearly 1, 000 applications and found that more than 1/3 of all applications have declared unnecessary permissions [3]. In other word, more than 300 applications may use these unnecessary permissions to extract private data. Thus, it is so difficult to implement the security of Android system with current permission mechanism.

Furthermore, with the constant appearances of Android system vulnerabilities, there arise a large amount of attack behaviours and malwares. Many researchers pay attention to the malware detection, such as Jaewoo S who proposed an approach to distinguish the malware in Android Unity [4] and Huanran W who introduced object reference graph in malware detection [5]. Many attackers are used to exploiting the deficiencies of permission security mechanism to implement their illegal purpose. This illegal behaviour produces serious threat to user's privacy data such as obtaining user's privacy data and running Trojan program.

Because of the openness of Android system, the research on its security has never been interrupted. These researches help us to implement the effective protection to the private data. Furthermore, there are various researches on the mechanism of Android permission. Zhang Y et al. declared one dynamic analysis platform, named VetDroid [6]. This tool constructs the correct permission usage specification to prevent the malicious behaviour in malware. Pearce P et al. introduced a privilege separation framework upon the mobile advertisements [7]. This framework creates a new advertisement API, which separates the application and the advertisement. However, all these researches do not give user the ability to decide which permission can be granted. It is so limited in implementing the permission management.

Aiming at the drawback of Android permission mechanism, this paper puts forward a kind of permission management scheme based on machine learning. This research makes use of the application category and permission-related data to implement the application classification training and testing. The application will adopt specific permission white list corresponding to the classification result. This scheme helps users to manage permissions more actively, and furthermore, improves the flexibility of original permission mechanism and the security of Android OS. User is allowed to manage the permission properly to prevent risk behaviors such as accessing private data.

2. Android System Permission Mechanism

This section analyses the existing permission mechanism of Android and summarizes its defects. In order to refine the design and implementation of the permission management scheme, this section analyses the usage situation of the Android application permission.

2.1. Drawback of Existing Permission Mechanism. Android system adopts permission checking mechanism which protects Android devices and user's private data effectively. Meanwhile, there are also many defects shown in the following aspects.

Firstly, when user installs the application, they either grant the required permissions or reject them all. If you reject it, the installation process will be failed, which named "ALL-or-NONE" pattern [8]. It is obvious that this pattern makes an impact on the security of private data. If user sticks to install the application, he has no choice except to grant all permissions, although it is risky operation.

The second drawback is that as long as the application installation is complete, the granted permissions can no longer be modified. Sometimes, it is too difficult to be aware of the malicious behavior in the background for users. When they find out the problem, they have no idea to prevent this illegal behavior by modifying the permission list. In other word, if the multimedia data is accessed illegally, users cannot prevent this event happened unless uninstalling the application.

Thirdly, the permission mechanism lacks the sufficient protection, especially in the native layer, For example, when malware use JNI method to operate the Native layer of the system and obtain the system services, which would lead to worse situation.

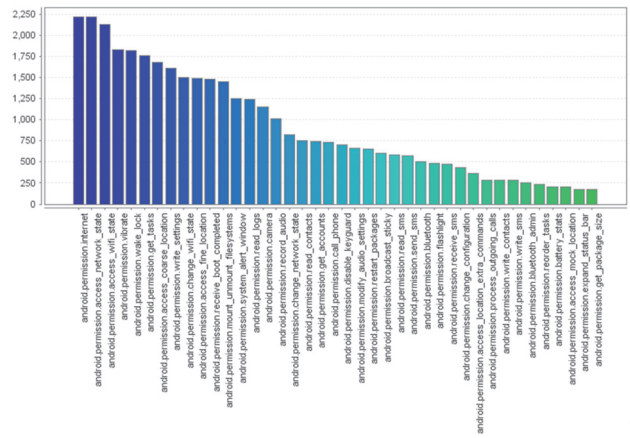


FIGURE 1: The permission statistic result. The column chart shows the first forty permissions. Nearly 2240 applications have declared the permission of accessing Internet. The fortieth (nearly 210 times) is obtaining the package size of application which belongs to private data.

These defects bring a number of problems to the Android system and have an impact on the security and practicability of the multimedia data on the device. Excessive permissions are requested in the process of application development, which may be exploited by malicious software. User cannot predominate over the management work of the permission mechanism, when they wonder to modify the configuration. Thus, the existing permission mechanism still cannot satisfy the requirement from users.

2.2. Statistical Analysis of Permission Usage in Android system. As for Android applications, different applications have different functions. If the application needs to make use of the system resources, the application has to declare the corresponding permissions in the *AndroidManifest* file. Thus, different types of application have different permission declarations. This paper summarizes the permission sets according to the different application categories. And next, this section analyzes the differences of permission sets in different categories and the relevance of different permissions in the same category.

More than 2200 applications were downloaded in this research, and the samples applications were divided into 21 categories according to the method provided by one popular app market software [9]. The permission statistic result is shown in Figure 1. It is easy to conclude that 99% of the applications in the statistical sample declare the permission named *android.permission.internet*. While *android.permission.access_network_state* and *android.permission.access_wifi_state* these two permissions are the second and third most permission declared in application.

The average amount of permissions declared in each group is found to be different, it is concluded that the

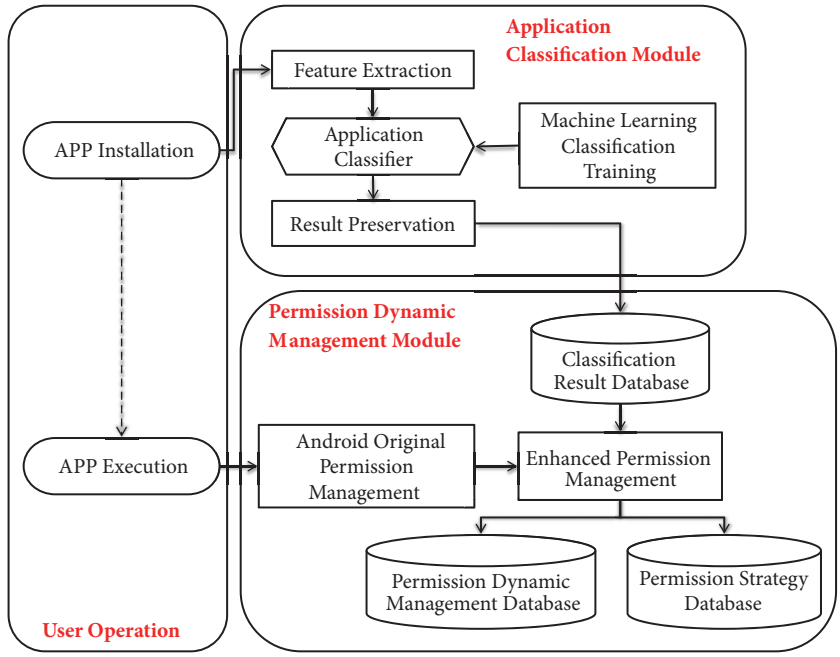


Figure 2: The architecture of the permission management scheme.

applications of communication's category have more granted permissions than other categories generally. The applications of children's category have the least declared permissions, since they are less functional and simpler to implement.

Based on the result of the above analysis, it is easy to know that the permissions between different application categories is very different. Therefore, the category that the application belongs to should be taken into consideration in the permissions management scheme.

3. Design of Permission Management Scheme

This section mainly describes the overall design of the permission management scheme. It accomplishes dynamic permission management according to the analysis result of the permission usage situation of different application categories. The scheme is divided into application classification module and dynamic permission management module.

3.1. The Architecture of Scheme. This paper aims to design an improved permission dynamic management scheme. After the researching on Android permission management technology, the permission management scheme of Android application software is designed and implemented and the architecture is shown in Figure 2. The whole scheme consists of application classification module and permission dynamic management module.

The application classification module will firstly extract the permissions and the sensitive API information by decompiled the APK file. These two records will be considered as the feature value of application classification.

The permission dynamic management module employs the classifier mentioned above to determine which one category the application belongs to and then puts forward the permission warning to users according to the permission whitelist of the corresponding category. Finally, the module asks users to decide whether to grant the permissions.

3.2. Application Classification Module. The accurate and efficient work of classifier provides the whole scheme guarantee of the effective execution. Actually, classification technology is employed to solve problem in a large scale of industries, even in medical business [10]. Most machine learning algorithms provide a convenient way to represent sequential observations and tend to cluster between different components [11]. Therefore, this section mainly introduces the design of the application classifier model generation, and then explains each step in the module.

3.2.1. Training of Application Classifier. The application classifier is obtained by training and testing a large number of APK applications on PC or server, which is the most critical and complex facet of the permission management scheme. The process of machine learning classification training includes the collection of APK data set, data preprocessing, data modelling, feature extraction and processing, training, and testing (see Figure 3). These steps are described below.

Data Collection and Preprocessing. The APK data sets are selected from YingYongBao which is the most popular app store software in Chinese market. The data sets will be preprocessed and saved in different categories. Eventually, the collection obtained a total of 2240 APK data sets and all these data sets were stored in different categories. There are 21 categories in the application classification, which includes Security, Office, Navigation, Children, Tools, Shopping,

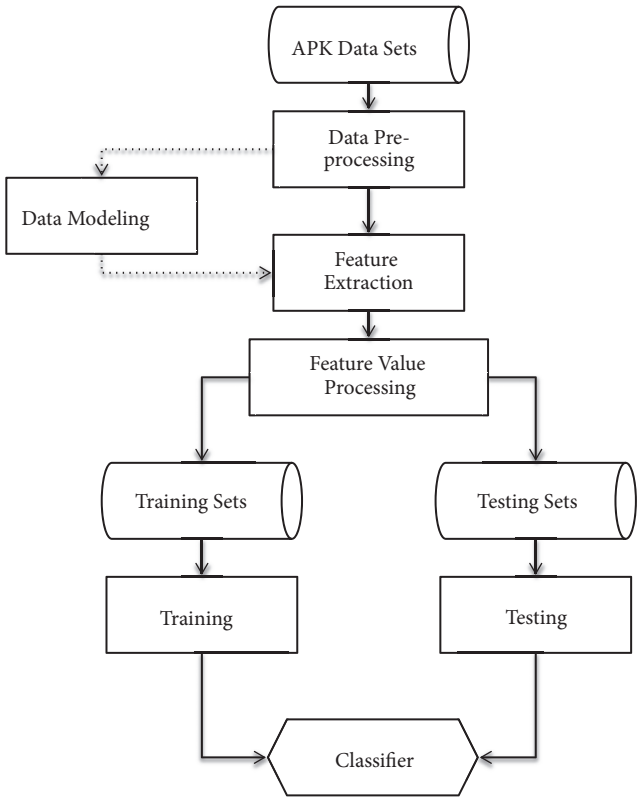


Figure 3: Flowchart of application classification training.

Health, Education, Financial, Traveling, Prettification, Social, Photography, Life, Video, Communication, System, News, Music, Entertainment, and Reading.

Data Modeling. Because the same category of Android applications implements similar functionalities, it is easy to find that they apply similar system APIs and permissions. Thus, this paper adapts the combination of permissions and sensitive APIs as the feature value in the process of data modelling.

This paper adopts the vector space model [12] and transfers the APK file into a multidimensional vector according to the importance of the feature items, and each feature matches one dimensional in the vector. The weights of feature items are calculated with Binary method. It is assumed here that the collected data sets are expressed as

$$(P_1, A_1, C_1), \ldots (P_i, A_i, C_i), \ldots, (P_n, A_n, C_n) \quad (1)$$

where P_i represents the permission usage situation of No.i APP, for example, $P_1 = (0, 1, 0, 1, 1)$ means APP1 employs the second, fourth, and fifth permission, A_i represents the usage situation of sensitive API in the No.i APP, and $A_1 = (1, 0, 1, 0, 0)$ means that APP1 uses both the first and third sensitive APIs. In addition, $C_i \epsilon \{21$ *application categories*$\}$ for example, C_1 = Education which represents APP1 belonging to education category. According to the method above, the APK data can be represented in vectors completely.

Extraction and Processing of Feature Values. Feature value extraction and processing is the most time-consuming and complex process in the application classification. The permission information in the APK file can be extracted from the *AndroidManifest.xml* file by decompiling the APK file. As for the sensitive API information, firstly use *Apktool* [13] to decompile the APK file, analyze the Smali file to obtain the API used in the APK, and after that take the API data to make the String comparison with the sensitive API library [14]. When the API is coincided with the record in database, it will be included in the feature value of this APK file.

Training and Testing. In the classification algorithm of machine learning, the whole data set will be divided into training set and test set. The training set is used to train the model according to a certain classification algorithm in order to produce the classifier [15]. The test set is used to test the classification ability of the classifier we obtained. This paper uses open source machine learning algorithm to train and test the data set, like Logistic regression algorithm [16].

3.2.2. Implementation of Application Classification Module. The application classification module employs the generated classifier to classify the application when it is installed and then saves the result in the classification result database. The detailed processes of the application classification module are shown in Figure 4.

Feature Information Extraction. The feature information extracted from APK applications includes Permission and API, which will be used in classification. Therefore, it is necessary to transfer the above feature information to the feature value that the classifier can recognize.

Classify Prediction. The first step of classify prediction process is to put the permission and sensitive API information into the classifier. The classifier judges and predicts which category this application belongs to and then outputs the classification result.

Saving Classification Results. This step is mainly to store the classification results in the classification results database. There is one classification result table in the database, which is used to store the classification results predicted by the application classifier for all the applications installed in the system. It mainly contains the UID of the application, the name of the application, and the category to which the applications belong.

3.3. Permission Dynamic Management Module. The permission dynamic management module enhances the original permission mechanism, which makes it become more fine-grained. The module can dynamically grant the permission and withdraw the authorization in the execution of the program and even notice users about the details of current permissions. This section includes the design and implementation of database, the design, and implementation of dynamic permission management.

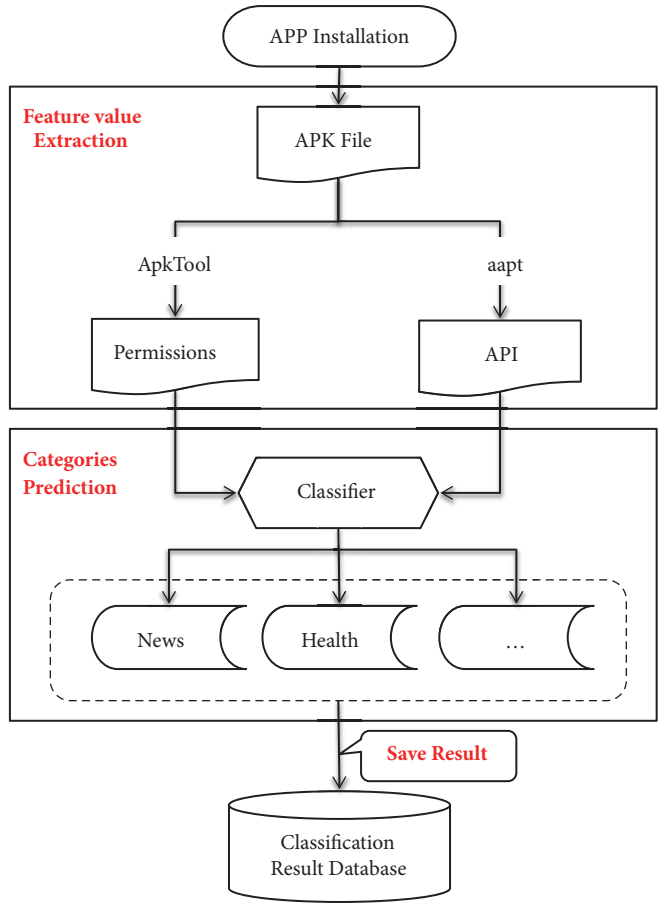


Figure 4: The flowchart of application classification module.

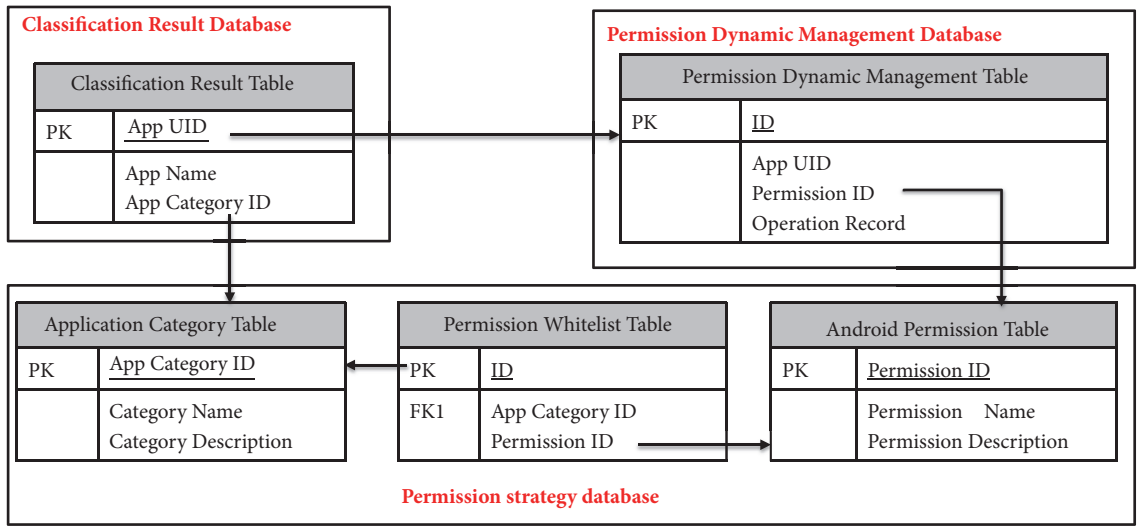


Figure 5: Database module.

3.3.1. Database. Enhanced permission management is based on three databases: they are the classification result database, application permission dynamic management database, and permission policy database. Figure 5 shows the logical model of all three databases.

Classification Result Database. The classification result database contains only one table, that is, the classification result table. The attribute named App UID is used to query the category of the application in the dynamic permissions management.

Permission Dynamic Management Database. This database contains one table named application permission dynamic management table. The table is empty at the very first. It will be updated with the installation and usage of the application. When an application first uses the permission, the dynamic management table will add a record indicating that this application had used this permission before and the operation is recorded as grant.

Permission Strategy Database. The permissions strategy database contains three tables. The application category table stores the information of application categories. Each category contains similar application functions and the same purposes. This table contains 21 categories mentioned in the Section 3.2.1. The Android permission table records all the permission information. The Permission whitelist table stores 10 permissions at most for each application category, which regarded as the whitelist permissions of the application category. If the application requests permissions which are not included in the whitelist table, the permission dynamic management module will inform user and then lets the user decide whether to grant the unknown permission.

3.3.2. Permission Dynamic Management. The enhanced permission management model makes an improvement on the original permission management model. In other word, when application make use of system resources, assuming that it have obtained authorization from the Android original permissions checking, the enhanced permission management still needs to further check. The enhanced one asks users to decide whether to authorize and update the database records at the same time. The detailed process of the enhanced permission management is shown in Figure 6.

Step 1 (query the permissions dynamic management database). The permissions dynamic management database is used to save the grant state of all permissions for this application. The state is either granted or rejected. When the database is initialized, the application does not have any granted or blocked permissions. Thus, this table will be empty when the application is installed. While the application is being executed, the application will be authorized dynamically. Additionally, users can also dynamically manage the database through a specific graphical interface.

After passing through the system permission check, the permission dynamic management module will query the permissions dynamic management database.

(a) If this application has been granted this permission, it will be passed and allowed to operate;

(b) If the permission has been rejected, the operation will be forbidden and terminated;

(c) If you do not have operation record for this permission, then go to *Step 2*.

Step 2 (identify application categories). The classification result database is used to obtain the category of the application. The classification result is predicted by using the classifier and saved in the database when the application is installed. Next go to Step 3.

Step 3 (query the permissions strategy database). Query the permission strategy database to confirm whether the permission is in the whitelist according to the category to which the application belongs.

(a) If it is in the whitelist, which means that the permission is common used and reliable in this application category, then go to the *Step 5*.

(b) If not, go to *Step 4*.

Step 4 (inform user about the permission request). Permission is not on the whitelist, which represents that the permission may be a risky permission for this application. The scheme needs to remind the function of the permission and the risk of granting permission to the user. User can determine whether or not to grant this permission.

(a) If the user chooses to grant the permission, go to *Step 5*;

(b) If the user chooses to block the permission request, go to *Step 6*.

Step 5 (pass the permission checking and update the dynamic management database). At this step, the request of new permission has already passed the checking of dynamic management module. User can access the resource in a safe environment. On the other hand, the record of this granted permission in the permissions dynamic management database needs to be updated. The state of this permission should be marked as granted. Finally, the process ends here.

Step 6 (reject the request and update the permissions dynamic management library). At this step, the request of new permission did not pass the checking of dynamic management module. It will be prohibited to access the requested resources. Meanwhile, the record of this rejected permission in the permissions dynamic management database needs to update. The state of this permission should be marked as rejected. Finally, the process ends here.

4. Experimental Verification and Analysis

This section introduces the verification and analysis of the scheme's feasibility. The first step is the training process of classifier, and then the classifier is applied to test environment to verify the security and reliability of the dynamic permission management system.

4.1. Testing Environment. This paper implements the scheme in the development environment shown in Table 1.

4.2. Machine Learning Algorithm. In order to obtain the most accurate application classifier, we need to select the appropriate machine learning algorithm to train and test the extracted data sets. This paper makes use of open source machine learning algorithms to train the classifier.

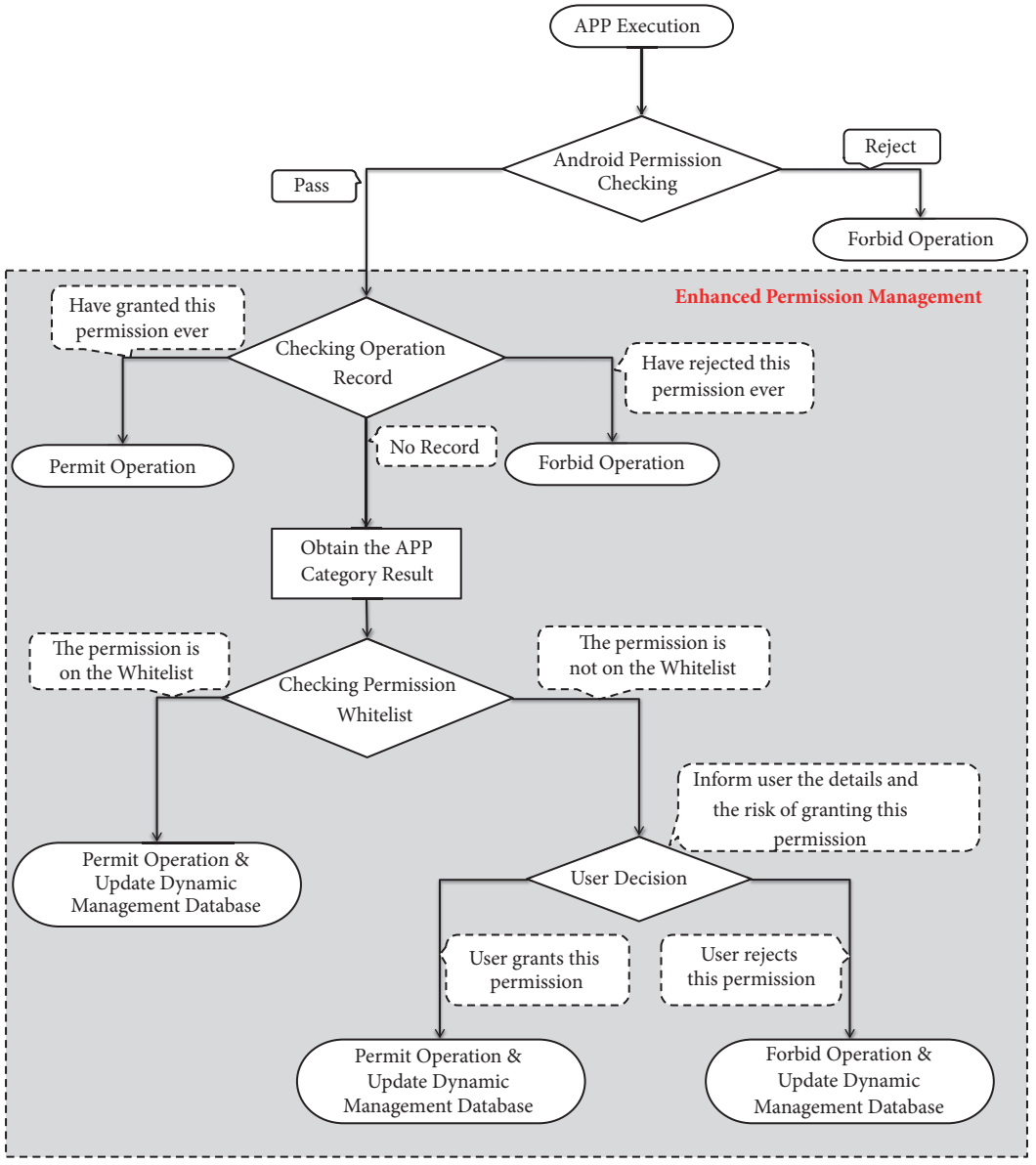


Figure 6: Enhanced permission management mechanism.

Table 1: Developing environment.

Device Version	Nexus 5
Android Version	4.4.2
Baseband Version	M8974A-1.0.25.0.23
Kernel Version	3.4.0-g0315133android-build@wpiy2.hot.corp.google.com
OS Version	aosp_hammerhead-userdebug 4.4.2 LMY48M

The widely used machine learning algorithms include Naive Bayes, Logical Regression, Decision, Tree and SVM.

In the scheme, the classification features are related to the app functions and attributes. However, the functions and attributes will be iterated and changed constantly with the development of application, so it is necessary to incorporate new training data to update the application classifier in time. In addition, the correlation between the application features has little impact to the accuracy of the classifier in our scheme.

In the research to the machine learning algorithms, we found that Logical regression algorithm has many methods of regularization model, so it does not need to consider the correlation of features. Comparing with the decision tree and SVM, logical regression algorithm is easier to update

Table 2: Results of different machine learning algorithm comparison.

Machine learning algorithm	Auc score	Accuracy	FPR	TPR
SMO	0.90	0.94	0.06	0.83
Logical Regression	0.91	0.94	0.07	0.80
Native Bayes	0.80	0.87	0.21	0.90
Decision Tree	0.84	0.89	0.15	0.95

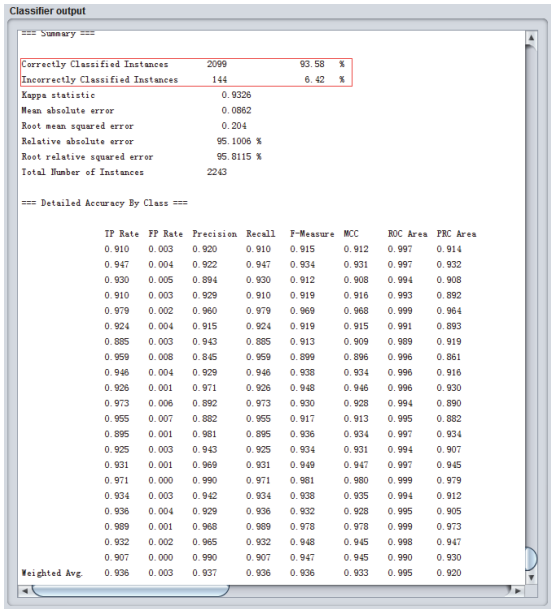


Figure 7: The classifier output module. There are 21 rows in the detailed accuracy by class, each row representing one category. The last row represents the weighted average of all application.

the model and incorporate new features data. As for the application classifier, it will be always updated with the generation of new application categories. Thus, Logical regression algorithm is the most suitable classification algorithm for the application classifier.

4.3. Classifier Training. This paper uses Weka 3.8.0 tools to implement the application classifier. Weka integrates a large number of machine learning algorithms that are able to undertake the task of data mining. The feature datasets of application permissions and sensitive API extracted from the previous steps are used in Weka as training and test datasets. Weka integrates machine learning algorithms such as SMO, logical regression, Bayesian, and decision tree.

The Classifier output module of Weka gives out the result of training and testing in the form of text, which is shown in Figure 7. The "Correctly Classified Instances" filed means that the classification accuracy is up to 93.58% by means of selecting permission and API to be the feature value for machine learning.

We also compare above four machine learning algorithms and select four metrics to evaluate their efficiency. The result is shown in Table 2. It is obvious concluding that the logic regression algorithm shows the best results both in auc score

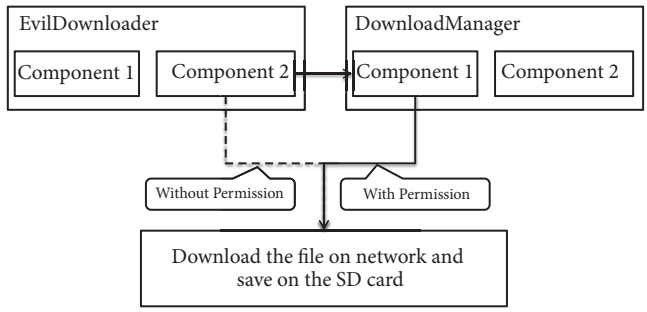


Figure 8: The Principle of this privilege escalation attack. EvilDownloader invokes the exposed component (broadcast receiver) in DownloadManager to achieve the goal of downloading file from internet.

and accuracy. All the accuracy scores are more than 0.8, which means it is feasible to use permissions and sensitive APIs in application classification.

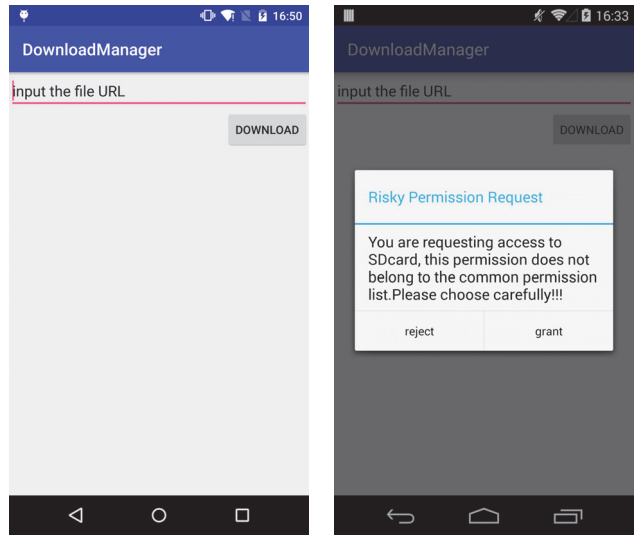
4.4. Function Testing. This paper designed a privilege escalation attack [17] to test the function of the system. The attack model program contains two applications: *EvilDownloader* (attack program, without privilege to access the network and SDcard) and *DownloadManager* (download management program, user had already granted this program with the permission to access the network and SDcard). The principle of this attack is that *EvilDownloader* can access the restricted resources through *DownloadManager*, which achieves the purpose of escalating *EvilDownloader*'s privilege. The principle is shown in Figure 8.

DownloadManager actually simulates a download program, enters the URL of the file, clicks the download button, and starts downloading. The downloaded file is saved to the SD card, and the program is shown in Figure 9(a).

DownloadManager has the permission to download files from internet and save them to the SD card. It downloads files by receiving a request broadcast and then accessing the URL address to download the file. The permission declaration and the broadcast receiver information is defined in the AndroidManifest.xml, which shown in Figure 10.

EvilDownloader behaves as an attack program without any permission. It is worth mentioning that it can also send a broadcast request to download the URL file through Intent, as shown in Figure 11.

From privilege escalation attack code shown in Figure 11, it represents that the *EvilDownloader* will download URL file "http://i4.piimg.com/11340/7f638e192b9079e6.jpg" and save the file, which the file name is "jb.png". In other word,



(a) (b)

Figure 9: The interface of DownloadManager. In figure (a), there are an EditText and button on the UI. User need to input proper URL and click the button to download multimedia data. In figure (b), the improved permission management system pops out the alert dialog to inform user the risk operation and user can choose "reject" to protect multimedia data from being accessed illegally.

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="DownloadManager"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".MainActivity"
        android:label="DownloadManager" >
        <intent-filter...>
    </activity>
    <receiver android:name=".DownloadReceiver">
        <intent-filter>
            <action android:name="com.droider.download"></action>
        </intent-filter>
    </receiver>
</application>
```

Permission

BroadcastReceiver

Figure 10: The AndroidManifest.xml code segment of DownloadManager.

```
setTitle("EvilDownloader");
button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent();
        intent.setAction("com.droider.download");
        intent.putExtra( name: "url", value: "http://i4.piimg.com/11340/7f638e192b9079e6.jpg");
        String filename = "jb.png";
        intent.putExtra( name: "filename",filename);
        sendBroadcast(intent);
    }
});
```

Figure 11: Code segment of EvilDownloader.

EvilDownloader downloads the file and stores it in SD card successfully, which implements the privilege escalation attack on the device. This type of attack poses a giant threat to multimedia data and can easily lead to the disclosure and damage of privacy data.

```
27538-27538/? I/Xposed: [Application Classification]: new application is installing
27538-27538/? I/Xposed: [Application Classification]: PacakgeName = com.admin.downloadmsnager; UID = 100085
27538-27538/? I/Xposed: [Application Classification]: Features = [0,0,0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,
27538-27538/? I/Xposed: [Application Classification]: Classifier Prediction.....
27538-27538/? I/Xposed: [Application Classification]: Category Result = [tools]
27538-27538/? I/Xposed: [Application Classification]: Saving Result.....
27538-27538/? I/Xposed: [Application Classification]: classification work finished
```

Figure 12: The installing log of Downloadmanager. The system checks the application and predicts the classification result of the application and, finally, saving the result in the database.

In the experiment, the attack is respectively implemented in the normal Android environment and the environment in which our scheme is deployed.

4.4.1. Classifier Experiment. The experiment installed these two applications by the way of ADB. Figure 12 is the installing log of *DownloadManager*. The figure shows that the application classification module monitors the installing process of the application and extracts the feature value. And next, the classifier predicts the category of the installing application. The result shown in the Figure 12 is that *DownloadManger* belongs to "Tools" category.

Therefore, in the improved permission management mechanism, *DownloadManger* is executed with the permission whitelist of "Tools" category. If there is any new permission need to be granted, user will receive the alert about the suspected permission request.

4.4.2. Normal Android Environment

The Execution Result. EvilDownloader can successfully exploit the components of *DownloadManager* to download the specified file and save it in SDcard without user's awareness.

Table 3: CPU occupancy rate comparison for five applications.

CPU occupancy	APP1	APP2	APP3	APP4	APP5
without Scheme Deployed	33%	27%	41%	60%	57%
with Scheme Deployed	41%	30%	47%	67%	66%

Table 4: Memory occupancy comparison for five applications (M).

Memory occupancy	APP1	APP2	APP3	APP4	APP5
without Scheme Deployed	17	29	54	44	113
with Scheme Deployed	31	34	77	83	147

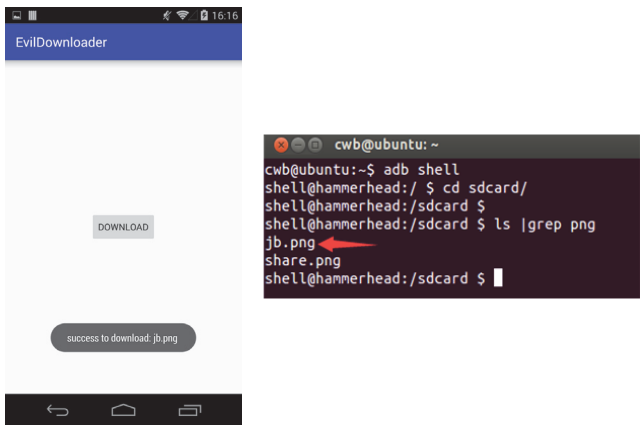


Figure 13: Experiment result in Normal Android Environment: there is a toast shown on the UI to inform the file is downloaded successfully. Checking the file folder of sdcard path, the file "jb,npg" is shown in the list.

The experimental results are shown in Figure 13. The file named "jb.png" was successfully downloaded and saved. Thus, *EvilDownloader* implemented privilege escalation attack. The attack program finished the unauthorized operation without requesting the permissions about accessing to the network and SDcard.

4.4.3. Environment with Scheme Deployed

The Execution Result. After clicking *download* button, the toast "*successful download jb.png*" did not pop up. Instead, the dialog appears on the phone, which is shown in Figure 9(b).

User has to choose whether to grant this permission or not. When clicking the "grant" button, the program will continue to work: download the file and save it on the SD card. However, if you click the "reject" button to reject the request of permission, there will appear exception to block the download operation. The SecurityException is shown in Figure 14.

The main reason is that the required permission is not in the permission whitelist. Furthermore, after the request of permission is rejected, Android throws an exception to figure the unauthorized operation out. The permission management scheme in this paper will detect the risky request of

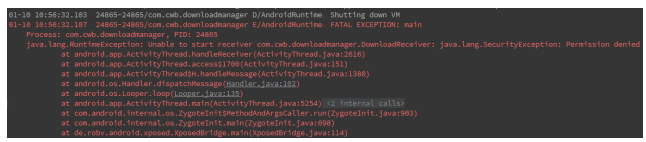


Figure 14: Experiment result in Environment with Scheme Deployed: the application throws an exception after the user choose "reject" to grant the permission. The program is interrupted by the improved permission management mechanism.

permissions and report it to the user in time. Once user rejects the authorization, system will throw a SecurityException exception to prevent malicious behavior effectively.

Therefore, the permission management scheme in this paper intercepts the malicious operation successfully, which protects the multimedia data from being accessed by attack program. This scheme accomplishes the security protection to the private data in Android devices and improves the security of current Android permission management system.

4.5. Performance Evaluation. This section conducts the performance evaluation to verify the efficiency of the scheme. It is obvious to find that the scheme will always interrupt the system operation when new request of permission appears on the phone. It is necessary to check the consumption of memory and CPU in the Android OS, and the time cost with the scheme running.

4.5.1. Device Consumption Evaluation. We executed five applications in the same testing device to compare the performance of the improved scheme. Table 3 records the CPU occupancy rate for the five applications in the two situations. The scheme obviously produce an impact to the CPU occupancy, but all the increase part are no more than 10%, which can be ignored.

Table 4 records the max value of memory occupancy when these five applications running in the devices. In the scheme, the process of querying the database is the major consumption point. According to the result, the average increase memory is no more than 25 M, which also can be ignored.

The result shows that the scheme increases the consumption of device, but it can be ignored in the running process.

Table 5: Installing Time comparison for five applications (ms).

Installing Time	APP1	APP2	APP3	APP4	APP5
without Scheme Deployed	3144	2779	4421	5843	5339
with Scheme Deployed	3797	4029	6711	6674	7936

4.5.2. Time Consuming Evaluation. The scheme resolves the APK file when the application is installed on the phone. This period is the most time-consuming part during the whole running process of the scheme. Table 5 records the time cost in the installing process. The amount of increase time depends on the size of APK file and the number of declared permissions. Generally, the larger the APK file is, the more time that the application costs in the scheme.

According to the results of performance evaluation, the improved permission management scheme produces the extra performance cost, but the amount is so small that can be ignored in the running process of the scheme.

5. Conclusions

This paper gives a research on permission management technology of Android application based on machine learning. The proposed scheme combines machine learning with permission management to enhance the original permission management mechanism. The improved scheme prohibits the illegal operation, like extracting multimedia data on the internet.

However, the implementation of the proposed scheme is based on *Xposed*. Because *Xposed* framework and relies on rooting Android devices, this requirement produces a great discount on the safety to Android system. It is necessary to migrate this scheme to the real Android system in the future work.

Data Availability

The data used to support the findings of this study are free and publicly available on Internet. The database data and classification training data used to support the findings of this study have been deposited in https://pan.baidu.com/s/1talTsZrqKzY_0paf7FrqWg. The password is "6dbv."

Conflicts of Interest

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

Acknowledgments

This work is supported by National Natural Science Foundation of China (nos. U1536121, 61370195). It is worth mentioning that the previous version of this work had been presented in the form of abstract poster on the ICCCS 2018, which was held on June 8, 2018, in Haikou.

References

[1] C. Jinhua, Z. Yuanyuan, C. Zhiping et al., "Securing Display Path for Security-Sensitive Applications on Mobile Devices Computers," *Computers, Material & Continua*, vol. 55, no. 1, pp. 017–035, 2018.

[2] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[3] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS '11)*, pp. 627–638, ACM, Chicago, Ill, USA, October 2011.

[4] S. Jaewoo, L. Kyeonghwan, C. Seong-je et al., "Static and Dynamic Analysis of Android Malware and Goodware Written with Unity Framework," *Security & Communication Networks*, vol. 2018, Article ID 6280768, 12 pages, 2018.

[5] W. Huanran, H. Hui, Z. Weizhe et al., "Demadroid: Object Reference Graph-Based Malware Detection," *Security and Communication Networks*, vol. 2018, Article ID 7064131, 16 pages, 2018.

[6] Y. Zhang, M. Yang, B. Xu et al., "Vetting undesirable behaviors in Android apps with permission use analysis," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*, pp. 611–622, ACM, Berlin, Germany, November 2013.

[7] P. Pearce, A. P. Felt, G. Nunez, and D. Wagner, "AdDroid: privilege separation for applications and advertisers in Android," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS '12)*, pp. 71-72, Seoul, Republic of Korea, May 2012.

[8] L. Lei and H. Yong, "The Research on Android Application of authority detection technology," *Journal of Information Security Research*, vol. 02, pp. 139–144, 2017.

[9] Classifcation of the popular APP, edited by 2017-6 http://sj.qq.com/myapp/.

[10] R. Gurusamy and V. Subramaniam, "A machine learning approach for MRI brain tumor classification," *Computers, Materials and Continua*, vol. 53, no. 2, pp. 91–109, 2017.

[11] Y. Zheng, B. Jeon, L. Sun, J. Zhang, and H. Zhang, "Student's t-hidden markov model for unsupervised learning using localized feature selection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[12] G. Sidorov, A. Gelbukh, H. Gómez-Adorno, and D. Pinto, "Soft similarity and soft cosine measure: Similarity of features in vector space model," *Computacion y Sistemas*, vol. 18, no. 3, pp. 491–504, 2014.

[13] Ma. Z, "Android application install-time permission validation and run-time malicious pattern detection," *Personal & Ubiquitous Computing*, vol. 18, no. 8, pp. 1963–1976, 2014.

[14] K. W. Y. Au, Y. F. Zhou, Z. Huang, and D. Lie, "PScout: analyzing the Android permission specification," in *Proceedings of the*

ACM Conference on Computer and Communications Security (CCS '12), pp. 217–228, ACM, October 2012.

[15] W. Huang, T. Zhao, and J. Wang, "An enhance excavation equipments classification algorithm based on acoustic spectrum dynamic feature," *Multidimensional Systems & Signal Processing*, vol. 28, no. 3, pp. 1–23, 2017.

[16] H. Hou, "Research on the correlation of learning behavior characteristics of mooc platform based on logistic regression algorithm," *Journal of Shangrao Normal University*, 2017.

[17] L. Davi, A. Dmitrienko, A. R. Sadeghi et al., "Privilege Escalation Attacks on Android Information Security," in *Proceedings of the ISC 2010 International Conference*, vol. 6531, pp. 346–360, 2011.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 5037892, 10 pages
https://doi.org/10.1155/2018/5037892

Research Article

A Secure Multimedia Data Sharing Scheme for Wireless Network

Liming Fang,[1,2] Liang Liu,[1,2] Jinyue Xia,[3] and Maosheng Sun[4]

[1]*Nanjing University of Aeronautics and Astronautics, 29 Yudao Street, Nanjing, Jiangsu 210016, China*
[2]*Key Laboratory of Computer Network Technology of Jiangsu Province, China*
[3]*IBM, 3039 E Cornwallis Rd, Research Triangle Park, NC 27709, USA*
[4]*Yangzhou University, 88 South University Ave., Yangzhou, Jiangsu 225009, USA*

Correspondence should be addressed to Liang Liu; flm12311231@hotmail.com

Received 12 June 2018; Accepted 3 September 2018; Published 18 October 2018

Guest Editor: Zhaoqing Pan

A large number of wireless devices like WiFi cameras and 4G robots have been deployed in the rapidly growing wireless network such as Internet of Things. All of the devices (sensors) are collecting and analyzing multimedia data all the time while they are actively working, and it is also required to share data among these the sensors. Typically, the wireless data is transmitted through the network gateway or the cloud platforms. In such a wireless environment, if there is no appropriate protection to the data, it is easy to cause potential data leakage. In reality, the owner of the sensor might only want to share the multimedia data stored in the sensor with a trusted third party (e.g., a family member or a coworker) through an internet gateway or the cloud platform. Ideally, the gateway or the cloud platform in the wireless network should transform one user's encrypted data (wireless multimedia data) directly into another ciphertext under a set of new users (e.g., a trusted third party) without accessing the user's plaintext data. In this work, a new secure notion called fuzzy-conditional proxy broadcast re-encryption (FC-PBRE) is presented to address the concern. In a FC-PBRE scheme, the proxy (the gateway or cloud server) uses a broadcast re-encryption key to re-encrypt the encrypted wireless multimedia data which can be decrypted by a set of delegatees if and only if the broadcast key's conditional set W is close to the conditional set W' of the ciphertext. With the FC-PBRE scheme, the wireless multimedia data is not disclosed and cannot be learnt by the proxy (the gateway or cloud server). In this paper, we first present the definition of security against chosen-ciphertext attacks for FC-PBRE. Second, we propose an efficient fuzzy-conditional proxy broadcast re-encryption scheme. Third, we prove that our FC-PBRE scheme is CCA-secure in the random oracle model based on the Decisional nBDHE assumption.

1. Introduction

Generally, a user (can be an attacker) has the access to the data stored in the wireless devices via (1) a direct connection with the devices, (2) the gateway of the network, and (3) a cloud platform. Because many wireless devices need to localize the configuration before they start working, they usually have a backend configuration interface (e.g., WEB, PC) open. Moreover the devices sometimes go offline even when they are actively working. As a result, the attacker can simply hack the login system of the configuration interface and obtain the multimedia data. In this case, even though the device and the cloud have been authenticated and encrypted (e.g., using SSL), the attack cannot be prevented. So, ensuring the multimedia data to be stored and shared securely has become extremely important [1–8]. To share the data partially, it would be good if all the multimedia data are encrypted via the data owner's public key. For example, the device owner can give the data permission of the wireless camera configuration to the maintenance engineer, but the maintenance is not allowed to access to the video data. Obviously, such a scheme cannot allow sharing the private key directly with the maintenance engineer because sharing the private key means exposing the data to the engineer.

One approach to ensure data confidentiality in the wireless network is that user data can be encrypted before it is updated into the server. The encryption technology is an effective way to protect user data; however it does have some drawbacks. For example, when a cloud user Alice shares her data with another user Bob, her encrypted data cannot be as the plain data and cloud servers should not be directly transmitted to the shared user's ciphertext, because users

(including Bob) cannot decrypt the received data. Of course, one easy way for Alice to fix this issue is to download her own encrypted data that is saved in the cloud, then decrypt and upload the decrypted data to the cloud again, and finally send it to Bob who Alice wants to share the data. However, using this method, the user's data will be obtained by the untrusted third-party cloud, which cannot guarantee the data confidentiality. Therefore, in the cloud storage environment, a security mechanism is needed to allow the cloud server to transform the encrypted data of users directly into another shared user's encrypted data without accessing the user's plaintext data.

Since the data security problem of users in the nontrusted third-party cloud server is increasingly prominent and traditional encryption technology has been unable to meet these application needs, the cloud server really should be able to convert a user's ciphertext to a second's user ciphertext on the basis of not involving decryption. There have been quite a few researches done in this area to address this concern. For example, proxy re-encryption [9] can directly transfer encrypted data of a user Alice, stored in a nontrusted third-party cloud under the authorization of Alice, into user Bob's ciphertext, so as to achieve data sharing of Alice and Bob. Because of this feature, proxy re-encryption has been applied in IoT, cloud computing, email forwarding systems [9], and distributed file systems [10]. In proxy re-encryption, the third party can convert all Alice's ciphertexts into Bob's ciphertext, but in many applications, Alice hopes that the third party can only transform some specific conditions of ciphertext instead of all ciphertext. For example, the owners of wireless devices might only want to share part of the encrypted multimedia data instead of sharing all the data with a group of other users. To achieve that goal, conditional proxy re-encryption was proposed to provide such a mechanism that allows Alice to freely determine which encrypted data needs to be shared with Bob [11]. The third party in the conditional proxy re-encryption scheme has the ability to convert a ciphertext only if it meets certain specific conditions.

For applications like group photo sharing, however, conditional proxy re-encryption scheme becomes a problem if one person's multimedia needs to be shared with a group of users via a cloud platform. For example, in a scenario of a picture's owner (Alice) who wishes to share the encrypted photo data with the family members, the cloud cannot directly forward the owner Alice's encrypted data of camera to a group of family members, since only Alice has the private key to decrypt after forwarding. Although conditional proxy re-encryption can convert Alice's ciphertext into a different ciphertext, only the one person can decrypt the ciphertext, not a group of people to decrypt. Thus it cannot be adapted to the situation of group data sharing. Recently, the conditional proxy broadcast re-encryption(C-PBRE) was presented [12] to resolve the issue. In a C-PBRE scheme, a user's ciphertext can be transformed to another ciphertext for a group of users by a third party proxy. Moreover, the third-party proxy can only convert ciphertext with specific conditions.

Although the C-PBRE has addressed some concerns, still there is a flaw, which is that the C-PBRE scheme cannot support fuzzy condition matching. Here, we give an example of an online medical service system to show the importance of fuzzy condition matching and the fuzzy-conditional proxy broadcast re-encryption (FC-PBRE). Usually a multimedia electronic medical record is a system that tracks the electronic medical record of the patients. It integrates image, video, audio, and text and everything can be stored in the cloud at the same time. Doing so enables the medical staff to look up all medical records related to the patient such as an MRI or X-ray image from a PC or a smart phone. In order to protect patient privacy of multimedia electronic medical records, we need to encrypt and protect relevant data. But how to implement the sharing of multimedia electronic medical records after encryption is a difficult problem and FC-PBRE comes to solve this problem. More specifically, in an online medical system, patients are more likely to find a doctor who meets the following requirements for the treatment of a cold. We can simply denote the requirements as follows: $\mathsf{R}_1 = (\text{“Cold”} \wedge \text{“fever”} \wedge \text{“runny nose”} \wedge \text{“sore throat”})$. With R_1 a patient can encrypt her health record before she uploads it to the medical system. However, the medical system cannot directly access the disease record in this case, because it does not necessarily have a matching secret key under R_1. What the system can do is re-encrypt the ciphertext so that other doctors may be able to see the case as long as these doctors meet at least $d(d \leq |\mathsf{R}_1|)$ conditions of R_1. By adopting FC-PBRE, a doctor sets up a different access policy R_2 and sends a re-encryption key rk_{R_1} to the proxy. When a doctor is away, the proxy can re-encrypt the ciphertext if and only if $|\mathsf{R}_1 \cap \mathsf{R}_2| \geq d$. However, in many situations Alice may want to cooperate with a set of colleagues satisfying R_2 to consultation on the patient's condition. In traditional FC-PRE, if Alice wants to consulate with N colleagues, the proxy needs to perform N re-encryption operations. The problem, though, is that the proxy's computation is linear with N; this may not be desirable in terms of the complexity. In contrast, in FC-PBRE, the proxy can re-encrypte Alice's ciphertext to a group of users at one time. As a result, an FC-PBRE scheme is likely to resolve this complexity problem.

In the multimedia data sharing environment, it is needed to have a secure mechanism that allows the cloud server to transform one user's encrypted data directly into another ciphertext under a set of new users without accessing the user's plaintext data. In this paper, fuzzy-conditional proxy broadcast re-encryption (FC-PBRE) is proposed to address the concern. In FC-PBRE, the proxy uses a broadcast re-encryption key to re-encrypt a ciphertext which can be decrypted by a set of delegatees if and only if the broadcast key's conditional set W is close to the conditional set W' of the ciphertext. With the FC-PBRE scheme, the plaintext data is not disclosed and cannot be learnt by the proxy. The paper is structured as follows. First we introduce the security definition against chosen-ciphertext attacks for FC-PBRE. Second, our efficient fuzzy-conditional proxy broadcast re-encryption scheme is presented. Finally, we prove that our FC-PBRE scheme is CCA-secure in the random oracle model based on the Decisional nBDHE assumption.

2. Related Work

Proxy re-encryption [9] can directly transfer encrypted data of user Alice stored in nontrusted third-party cloud under the authorization of Alice into user Bob ciphertext, so as to achieve data sharing of Alice and Bob. But the third party can transform all Alice's ciphertext into Bobs ciphertext in PRE scheme, but in many applications, Alice hopes that the third party can only transform some specific conditions of ciphertext instead of all ciphertext. In a C-PRE scheme [11], only the one ciphertext from Alice that meets certain specific conditions can be converted by third parties into a Bob's ciphertext. Therefore, with conditional proxy re-encryption, Alice can determine which ciphertext for third parties to re-encrypt; thus a flexible control of the ciphertext can be observed. Weng et al. [11] proposed conditional proxy re-encryption scheme and they proved that it is chosen-ciphertext attack secure in the random oracle model. Similarly, Tang [13] proposed a type based proxy re-encryption scheme, a secure proxy re-encryption with keyword search scheme, and it is proven secure in the random oracle model [14]. On top of the keyword search scheme, an anonymous conditional proxy re-encryption with keyword search scheme was proposed by Fang et al. [15]. On the paper, they also proved that their scheme is chosen-ciphertext secure. Fang et al. [16] presented a fuzzy-conditional proxy re-encryption scheme and proved the security in the random oracle model. The scheme can support fuzzy matching between multiple keywords; that is, only some key words in the re-encryption key satisfy the matching, and the third party can complete the re-encryption.

Without the random oracle, a conditional proxy re-encryption scheme [17] was proposed to support a more fine-grained access strategy under the standard model. Reference [18] designed an identity based proxy re-encryption mechanism for multiple hop (Multihop) under the standard model. In this scheme, the re-encrypt ciphertext can still be re-encrypted repeatedly, and the length of ciphertext does not increase with the number of re-encryptions. So the length of the ciphertext is constant.

With regard to anonymity, Ateniese et al. [19] came out with the idea of anonymous proxy re-encryption and proved that their scheme is chosen plaintext attack security under the standard model. With anonymous proxy re-encryption, an attacker cannot obtain user's identity from the key. Later, a new scheme was proposed to achieve the security of chosen-ciphertext attack with the random oracle [20]. Subsequently, Shao et al. improved the scheme by using the standard model for the security proof [21]. Followed by Shao et al.'s work, a security model is enhanced for anonymous proxy re-encryption [22]. In the security model of [22], it allows attackers to get re-encrypted queries directly, instead of obtaining re-encrypted query by acquiring re-encryption key query. Shao et al. [23] proposed an anonymous ID based proxy re-encryption to extend anonymous proxy re-encryption to identity based anonymous proxy re-encryption. Their scheme is proven secure in the random oracle model. An anonymous identity based multiuser identity based proxy re-encryption scheme was proven CCA-secure in the standard model [18]. The same literature also analyzed its application in privacy protection and data sharing in big data storage system. The above schemes are about proxy anonymous encryption with user identity anonymity. A keyword anonymity conditional proxy re-encryption scheme [24] was demonstrated to achieve the anonymity of conditions.

3. Preliminaries

3.1. Bilinear Map. G and G_T denote two multiplicative cyclic groups with the same prime order p. g is a generator of group G. A bilinear pairing is a bilinear map $e : G \times G \longrightarrow G_T$ with the following properties:

(1) $e(U^a, V^b) = e(U,V)^{ab}$ for all $a, b \overset{R}{\leftarrow} Z_p^*$ and $U, V \in G$.

(2) $e(U,U) \neq 1$.

(3) $e(U,V)$ can be computed in polynomial time for all $U, V \in G$.

3.2. The n-BDHE Assumption. If p is a prime, let Z_p denote the set $\{0, 1, \ldots, p-1\}$ and Z_p^* denote the set $\{1, 2, \ldots, p-1\}$. Let $e : G \times G \longrightarrow G_T$ be a bilinear map. Given $2n+1$ elements

$$\left(\mu, \nu, \nu^{\gamma}, \nu^{\gamma^2}, \ldots, \nu^{\gamma^n}, \nu^{\gamma^{n+2}}, \ldots, \nu^{\gamma^{2n}}\right) \in G^{2n+1} \tag{1}$$

and an element $T \in G_T$, the adversary's task is to decide if $T \overset{?}{=} e(\mu, \nu)^{\gamma^{n+1}}$.

Denote ν^{γ^i} as ν_i, and define the advantage of an adversary $\mathcal{A}$ as

$$Adv_{G,\mathcal{A}}^{n-BDHE} = \left| \begin{matrix} Pr\left[\mathcal{A}\left(\mu, \nu, \nu_1, \ldots, \nu_n, \nu_{n+2}, \ldots, \nu_{2n}, e\left(\nu_{n+1}, \mu\right)\right) = 1\right] \\ -Pr\left[\mathcal{A}\left(\mu, \nu, \nu_1, \ldots, \nu_n, \nu_{n+2}, \ldots, \nu_{2n}, T\right) = 1\right] \end{matrix} \right| \tag{2}$$

where $\nu, \mu \in G$, $\gamma \in Z_p^*$, and $T \in G_T$ are randomly chosen. We conclude that the n-BDHE assumption relative to (G, G_T) holds [25], if $Adv_{G,\mathcal{A}}^{n-BDHE}$ is negligible for all probability polynomial time (PPT) adversary $\mathcal{A}$.

4. FC-PBRE Model and Security Notion

Two security definitions for fuzzy-conditional proxy broadcast re-encryption as well as its model are introduced in this section.

Definition 1 (FC-PBRE). A (single-use) proxy broadcast re-encryption scheme runs the following algorithms:

(i) $Setup(\lambda, n, d)$: in the setup step, for the input, λ is the security parameter, n is the maximum allowed number of users, and d is the threshold. The system at this step generates a public key PK and a master secret key MK.

(ii) $KenGen(PK, MK, i)$: given the public key PK, the master key MK, and a user i, the system generates secret key sk_i for the user i.

(iii) $Encrypt(PK, S, m, W)$: the $Encrypt()$ algorithm takes the public key PK, a user sets $S \subseteq \{1, 2, \ldots, n\}$, a message m, and a set of keywords W as the input, it then outputs the ciphertext C for the user set S with condition set W.

(iv) $ReKGen(PK, sk_i, S', W')$: on inputting PK, the user's private key sk_i, a set of users ($S' \subseteq \{1, 2, \ldots, n\}$) and a keywords set W', outputs the re-encryption key $rkey_{i,S',W'}$.

(v) $REnc(PK, rkey_{i,S',W'}, i, S, S', C)$: with the public key PK, a re-encryption key $rkey_{i,S',W'}$, the user i, two different user sets S and S', and the original ciphertext C, this algorithm computes the re-encrypted ciphertext C_R. This step requires $|W \cap W'| \geq d$, where d is the threshold and if the condition cannot be met, an error symbol $\perp$ will be output instead.

(vi) $DecryptL2(PK, sk_i, i, S, C)$: $DecryptL2$ represents the decryption algorithm for the original ciphertext. It takes the public key PK, user i, the private key sk_i of the user i, a set of users S, and ciphertext C for user set S, and it outputs the plaintext m.

(vii) $DecryptL1(PK, sk_j, i, j, S, S', C_R)$: $DecryptL1$ represents the re-encrypted ciphertext decryption algorithm. It outputs the plaintext m from the input of a public key, a user's private key sk_j, users i, j, user sets S, S', and a re-encrypted C_R. If the algorithm aborts, it outputs $\perp$.

Correctness. The definition of the correctness of a FC-PBRE scheme is as follows; given two user sets S, S', condition sets W, W', $C = Encrypt(PK, S, m, W)$, and $rkey_{i,S',W'} = ReKGen(PK, sk_i, S', W')$, then $C_R = REnc(PK, rkey_{i,S',W'}, i, S, S', C)$, if $|W \cap W'| \geq d$

$Pr[DeccyptL2(PK, sk_i, i, S, C) = m] = 1$, if $i \in S$;

$Pr[DecryptL1(PK, sk_j, i, j, S, S', C_R) = m] = 1$, if $j \in S'$.

The game-based model is used to define the security for the FC-PBRE scheme. Similar to a security model from [25], the CCA security of the FC-PBRE scheme is considered in the selective-set model. In such a model, the adversary is supposed to commit ahead the challenge user set S^* and the condition set W^*.

Definition 2 (IND-set-CCA game). The following lists two games between one adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Game 1 is to consider the security of the original ciphertexts. We define the IND-OR-CCA game as follows:

(1) Init. In the initial phase, the adversary $\mathcal{A}$ selects a target users set $S^* \subseteq \{1, 2, \ldots, n\}$ and the condition set $W^* = \{\omega_1^*, \omega_2^*, \ldots, \omega_n^*\}$.

(2) Setup. At this stage of the game, the challenger $\mathcal{C}$ runs $Setup(n)$, and as a result, he can generate the public key PK and the master key MK. Then he gives PK to $\mathcal{A}$.

(3) Query phase 1. $\mathcal{A}$ makes the following queries:

(a) $Extract(i)$: the challenger runs $sk_i = KeyGen(PK, MK, i)$ and returns sk_i to $\mathcal{A}$.

(b) $ReKGen(i, S', W')$: $\mathcal{C}$ runs $rkey_{i,S',W'} = ReKGen(PK, sk_i, S', W')$ and $sk_i = KeyGen(PK, MK, i)$ and returns $rkey_{i,S',W'}$ to $\mathcal{A}$.

(c) $REnc(i, S, S', C)$: $\mathcal{C}$ runs $REnc(PK, rkey_{i,S',W'}, i, S, S', C)$, where $rkey_{i,S',W'} = ReKGen(PK, sk_i, S', W')$ and $sk_i = KeyGen(PK, MK, i)$, and returns the output to $\mathcal{A}$.

(d) $DecryptL2(i, S, C)$: $\mathcal{C}$ runs $DecryptL2(PK, sk_i, i, S, C)$, where $sk_i = KeyGen(PK, MK, i)$, and returns the output to $\mathcal{A}$.

(e) $DecryptL1(i, j, S, S', C_R)$: $\mathcal{C}$ runs $DecryptL1(PK, sk_j, i, j, S, S', C_R)$, where $sk_j = KeyGen(PK, MK, j)$, and returns the output to $\mathcal{A}$.

$\mathcal{A}$ has to follow the restrictions in this phase. First, $\mathcal{A}$ cannot make $Extract(i)$ for any $i \in S^*$; second $\mathcal{A}$ cannot make $ReKGen(i, S', W')$ and $Extract(j)$, if $i \in S^*$, $j \in S'$, and $|W' \cap W^*| \geq d$.

(4) Challenge. As soon as $\mathcal{A}$ finishes Query phase 1, $\mathcal{A}$ computes two equal length messages (m_0, m_1). $\mathcal{C}$ selects a bit $b \in \{0, 1\}$ and sets the challenge ciphertext to be $C^* = Encrypt(PK, m_b, S^*, W^*)$. C^* is then sent to $\mathcal{A}$.

(5) Query phase 2. In addition to the restrictions in phase 1, $\mathcal{A}$ continues making queries with the following extra restrictions: $\mathcal{A}$ is not allowed to run $REnc(i, S^*, S', C^*)$ and $Extract(j)$ if $i \in S^*$ and $j \in S'$;

(a) $\mathcal{A}$ is not allowed to run $DecryptL2(i, S^*, C^*)$ for any $i \in S^*$;

(b) $\mathcal{A}$ is not allowed to run $DecryptL1(i, j, S^*, S', C_R)$ if $i \in S^*$, $j \in S'$ and $C_R = REnc(i, S^*, S', C^*)$.

(6) Guess. $\mathcal{A}$ makes the guess for b'. The adversary wins the game if $b' = b$.

The above adversary $\mathcal{A}$ is referred to as an IND-OR-CCA adversary. The advantage is defined as

$$Adv_{\mathcal{A},n}^{Game_1} = \left| Pr\left[b' = b\right] - \frac{1}{2} \right|. \tag{3}$$

Game 2. IND-Re-CCA game is used for the indistinguishability of the re-encrypted ciphertext.

(1) Init. To begin the game, an adversary $\mathcal{A}$ selects a target users set $S^* \subseteq \{1, 2, \ldots, n\}$ and a condition set $W^* = \{\omega_1^*, \omega_2^*, \ldots, \omega_n^*\}$.

(2) Setup. The challenger $\mathcal{C}$ generates the public key PK and master key MK by running runs $Setup(n)$. Then $\mathcal{C}$ gives PK to $\mathcal{A}$.

(3) Query phase 1. The adversary $\mathcal{A}$ makes the following queries:

(a) $Extract(i)$: $\mathscr{C}$ gets sk_i from $KeyGen(PK, MK, i)$. Then it returns it to $\mathscr{A}$. Note that $\mathscr{A}$ cannot make $Extract(i)$ for any user $i \in S^*$;

(b) $ReKGen(i, S', W')$: $\mathscr{C}$ runs $rkey_{i,S',W'} = ReKGen(PK, sk_i, S', W')$, where $sk_i = KeyGen(PK, MK, i)$, and returns $rkey_{i,S',W'}$ to $\mathscr{A}$.

(c) $DecryptL1(i, j, S, S', C_R)$: $\mathscr{C}$ runs $DecryptL1(PK, sk_j, i, j, S, S', C_R)$, where $sk_j = KeyGen(PK, msk, j)$, and returns the result to $\mathscr{A}$.

(4) Challenge. Once $\mathscr{A}$ finishes Query phase 1, $\mathscr{A}$ outputs two equal length messages (m_0, m_1). Challenger $\mathscr{C}$ chooses a bit $b \in \{0, 1\}$ and sets the challenge ciphertext to be $C^* = REnc(PK, rkey_{i,S^*,W^*}, i, S, S^*, C)$, where $i \in S$, $i \notin S^*$ and $C = Encrypt(PK, m_b, S, W^*)$. C^* is returned to the adversary $\mathscr{A}$.

(5) Query phase 2. $\mathscr{A}$ continues making queries but is subject to the following restrictions:

(a) $\mathscr{A}$ cannot make $Extract(i)$ for any $i \in S^*$;

(b) $\mathscr{A}$ cannot make $DecryptL1(i, j, S, S^*, C^*)$, if $i \in S$ and $j \in S^*$.

(6) Guess. $\mathscr{A}$ makes the guess b'. The adversary wins the game if $b' = b$.

The above adversary $\mathscr{A}$ is referred as an IND-Re-CCA adversary. Its advantage is defined as

$$Adv_{\mathscr{A},n}^{Game_2} = \left| Pr\left[b' = b\right] - \frac{1}{2} \right|. \tag{4}$$

A fuzzy-conditional proxy broadcast re-encryption scheme is called IND-Set-CCA-secure if for all PPT adversary $\mathscr{A}$, $Adv_{\mathscr{A},n}^{Game_1}$, and $Adv_{\mathscr{A},n}^{Game_2}$ are negligible.

5. The Proposed FC-PBRE Scheme

5.1. Our Construction. A Lagrange coefficient $\Delta_{\omega,S}(x)$ is defined for $\omega \in Z_p$ and a set S, of elements in Z_p:

$$\Delta_{\omega,S}(x) = \prod_{i \in S, i \neq \omega} \frac{x - i}{\omega - i} \tag{5}$$

Our FC-PBRE scheme contains the following algorithms:

(i) $Setup(\lambda, n, d)$: Let $M = \{0, 1\}^k$ denote the message space. The algorithm randomly selects $\alpha, \gamma \in Z_p$ and $Z \in G$ and computes $\nu_i = g^{\alpha^i}$ for $i = 1, 2, \ldots, n, n+2, \ldots, 2n$. Let $Hash_1 : \{0, 1\}^k \times G_T \longrightarrow Z_p^*$, $Hash_2 : G_T \longrightarrow \{0, 1\}^k$, $Hash_3 : G_T \times G \times G \times G \times \{0, 1\}^k \longrightarrow G$, $Hash_4 : Z_p^* \longrightarrow G$, and $Hash_5 : \{0, 1\}^k \longrightarrow Z_p^*$ be collusion-resistant hash functions. It computes $\upsilon = g^\gamma$. $Setup(\lambda, n, d)$ outputs the public key PK and the master key MK for the cloud server as follows:

$$\begin{aligned} PK &= \left(\nu, \nu_1, \ldots, \nu_n, \nu_{n+2}, \nu_{2n}, \upsilon, Z, Hash_1, Hash_2, Hash_3, Hash_4\right), \\ MK &= \gamma. \end{aligned} \tag{6}$$

(ii) $KeyGen(PK, MK, i)$: The secret key for a cloud user i is generated in this phase, where

$$sk_i = \nu_i^\gamma. \tag{7}$$

(iii) $Encrypt(PK, S, m, W)$: To encrypt a plaintext $m \in M$ under the set $S \subseteq \{1, 2, \ldots, n\}$ with condition W, it randomly picks a $R \in G_T$, computes $t = Hash_1(m, R)$, and outputs the results of the ciphertext: $C = (C_1, C_2, C_3, C_4, C_5, C_6)$.

$$\begin{aligned} C_1 &= R \cdot e\left(\nu_1, \nu_n\right)^t, \\ C_2 &= \nu^t, \\ C_3 &= \left(\upsilon \cdot \prod_{j \in S} \nu_{n+1-j}\right)^t, \\ C_4 &= \left(C_\omega = Hash_4\left(\omega\right)^t\right)_{\omega \in W}, \\ C_5 &= m \oplus Hash_2\left(R\right), \\ C_6 &= Hash_3\left(C_1, C_2, C_3, C_4, C_5\right)^t. \end{aligned} \tag{8}$$

(iv) $ReKGen(PK, sk_i, S', W')$: On inputting $sk_i = \nu_i^\gamma$, $S' \in \{1, 2, \ldots, n\}$ and W', it randomly selects $\sigma \in \{0, 1\}^k$ and a polynomial $f(x)$ with $d-1$ degree, where $f(0) = Hash_5(\sigma)$. For each $\omega \in W'$, it selects a random value r_ω and computes

$$\begin{aligned} a &= \left(a_\omega = sk_i \cdot Z^{q_{(\omega)}} Hash_4\left(\omega\right)^{r_\omega}\right)_{\omega \in W'}, \\ b &= \left(b_\omega = \nu^{r_\omega}\right)_{\omega \in W'}. \end{aligned} \tag{9}$$

It chooses random value $s \in Z_p^*$, $R' \in G_2$, computes $t' = Hash_1(\sigma, R')$, and sets

$$\begin{aligned} rkey_1 &= R' \cdot e\left(\nu_1, \nu_n\right)^{t'}, \\ rkey_2 &= \nu^{t'}, \\ rkey_3 &= \left(\upsilon \cdot \prod_{j \in S'} \nu_{n+1-j}\right)^{t'}, \\ rkey_4 &= \sigma \oplus Hash_2\left(R'\right), \\ rkey_5 &= Hash_3\left(rkey_1, rkey_2, rkey_3, rkey_4\right)^{t'}. \end{aligned} \tag{10}$$

It outputs the re-encryption key $rk_{i,S',W'} = (a, b, rkey_1, rkey_2, rkey_3, rkey_4, rkey_5)$.

(v) $REnc(PK, rkey_{i,S',W'}, i, S, S', C)$: On inputting a re-encryption key $rkey_{i,S',W'} = (a, b, rkey_1, rkey_2, rkey_3, rkey_4, rkey_5)$ and a ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6)$, it verifies whether the following hold:

$$e\left(C_2, \upsilon \cdot \prod_{j\in S} \nu_{n+1-j}\right) \stackrel{?}{=} e\left(\nu, C_3\right), \tag{11}$$

$$e\left(C_2, Hash_4(\omega)\right)_{\omega\in S} \stackrel{?}{=} e\left(\nu, C_\omega\right)_{\omega\in S}, \tag{12}$$

$$e\left(C_2, Hash_3\left(C_1, C_2, C_3, C_4, C_5\right)\right) \stackrel{?}{=} e\left(\nu, C_6\right). \tag{13}$$

If not, it outputs $\perp$. Otherwise, it randomly selects a d element set $F \subseteq W \cap W'$. If such a set F cannot be found, the algorithm outputs $\perp$, or else it computes

$$\widetilde{C_1} = C_1 \cdot \prod_{\omega\in F} \left(\frac{e\left(rkey_0 \cdot \prod_{j\in S, j\neq i} \nu_{n+1-j+i}, C_2\right)}{e\left(\nu_i, C_3\right) \cdot e\left(b_\omega, C_\omega\right)}\right)^{\Delta_{\omega,F}(0)} \tag{14}$$

Then the re-encrypted ciphertext is $C_R = (\widetilde{C_1}, C_2, C_5, rkey_1, rkey_2, rkey_3, rkey_4, rkey_5)$.

(vi) $DecryptL2(PK, sk_i, i, S, C)$: It takes a secret key sk_i and a ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6)$ as the inputs, it

(1) verifies if the three equations (11)–(13) hold. If not, it aborts and outputs the error $\perp$ or, else,

(2) computes $R = C_1 \cdot e(sk_i \cdot \prod_{j\in S, j\neq i} \nu_{n+1-j+i}, C_2)/e(\nu_i, C_3)$, $m = C_5 \oplus Hash_2(R)$, and $t = Hash_1(m, R)$ and verifies whether

$$C_2 = \nu^t, \tag{15}$$

$$C_3 = \left(\upsilon \cdot \prod_{j\in S} \nu_{n+1-j}\right)^t, \tag{16}$$

$$C_6 = Hash_3\left(C_1, C_2, C_3, C_4, C_5\right)^t \tag{17}$$

hold. It returns m or else returns $\perp$.

(vii) $DecryptL1(PK, sk_j, i, j, S, S', C_R)$: On inputting a secret key sk_j and a re-encrypted ciphertext $C_R = (\widetilde{C_1}, C_2, C_5, rkey_1, rkey_2, rkey_3, rkey_4, rkey_5)$, it proceeds as follows:

First verifying whether all of the equations hold:

$$e\left(rkey_2, \upsilon \cdot \prod_{j\in S} \nu_{n+1-j}\right) \stackrel{?}{=} e\left(\nu, rkey_3\right), \tag{18}$$

$$e\left(rkey_2, Hash_3\left(rkey_1, \ldots, rkey_4\right)\right) \stackrel{?}{=} e\left(\nu, rkey_5\right). \tag{19}$$

If they do not hold, it outputs $\perp$. Otherwise, it computes

$$\begin{aligned} R' &= rkey_1 \cdot \frac{e\left(sk_j \cdot \prod_{l\in S', l\neq j} \nu_{n+1-l+j}, rkey_2\right)}{e\left(g_j, rk_3\right)}, \\ \sigma &= rkey_5 \oplus Hash_2\left(R'\right), \\ g^s &= \frac{rkey_4}{Hash_4(\sigma)}, \\ t' &= Hash_1\left(\sigma, R'\right). \end{aligned} \tag{20}$$

It verifies whether

$$\begin{aligned} rkey_2 &= \nu^{t'}, \\ rkey_3 &= \left(\upsilon \cdot \prod_{l\in S'} \nu_{n+1-l}\right)^{t'}, \\ rkey_5 &= Hash_3\left(rkey_1, rkey_2, rkey_3, rkey_4\right)^{t'} \end{aligned} \tag{21}$$

hold. If not, it returns $\perp$, or else

it computes $R = \widetilde{C_1}/e(C_2, Z^{Hash_5(\sigma)})$, $m = C_5 \oplus Hash_2(R)$, and $t = Hash_1(m, R)$. It verifies whether $C_2 = \nu^t$ holds. If it holds, it returns m or else it returns $\perp$.

5.2. Security Proof. In section, we prove that our scheme is IND-Set-CCA-secure in the ROM.

Theorem 3. *If $Hash_1, Hash_2, Hash_3, Hash_4, Hash_5$ are target collision-resistant hash functions and the Decisional nBDHE assumption holds, the above scheme then is IND-Set-CCA-secure in the ROM.*

Lemma 4. *If an IND-OR-CCA is attacker $\mathcal{A}$ that can successfully attack FC-PBRE, then we are able to construct a simulator $\mathcal{S}$ that can solve Decisional nBDHE assumption.*

Proof. The simulator $\mathcal{S}$ is provided a Decisional nBDHE instance $(\mu, \nu, \nu_1, \ldots, \nu_n, \nu_{n+2}, \ldots, \nu_{2n}, T)$ and has to decide whether $T = e(\nu_{n+1}, \mu)$ from a random value. The simulator $\mathcal{S}$ controls the random oracles (RO) $Hash_1, Hash_2, Hash_3, Hash_4, Hash_5$ as follows:

$\mathcal{S}$ searches $Hash_1$ for (m, R, t). If $Hash_1$ list already exists, then $\mathcal{S}$ sends t to $\mathcal{A}$. Else, $\mathcal{S}$ randomly selects $t \in Z_p^*$, sends s to $\mathcal{A}$, and puts (m, R, t) in $Hash_1$ list.

$\mathcal{S}$ searches $Hash_2$ for (R, κ). If $Hash_2$ list already exists, then $\mathcal{S}$ sends κ to $\mathcal{A}$. Else, $\mathcal{S}$ randomly selects $\kappa \in \{0,1\}^k$, sends κ to $\mathcal{A}$, and puts (R, κ) to $Hash_2$.

$\mathcal{S}$ searches $Hash_3$ for $(C_1, C_2, C_3, C_4, C_5, \phi, \varphi)$. If $Hash_3$ list already exists, then $\mathcal{S}$ sends ψ to $\mathcal{A}$. Else, $\mathcal{S}$ randomly selects $\phi \in Z_p^*$, generates $\psi = \nu^\phi$, sends ψ to $\mathcal{A}$, and puts $(C_1, C_2, C_3, C_4, C_5, \phi, \varphi)$ to $Hash_3$ list.

$\mathcal{S}$ searches $Hash_4$ for (ω, τ, ϖ). If $Hash_4$ list already exists, then $\mathcal{S}$ sends ϖ to $\mathcal{A}$. Else, $\mathcal{S}$ randomly selects $\tau \in Z_p^*$ and sets $\varpi = \nu^\tau$, sends ϖ to $\mathcal{A}$, and puts (ω, τ, ϖ) to $Hash_4$ list.

$\mathcal{S}$ searches $Hash_5$ for (σ, ϱ). If $Hash_5$ list already exists, then $\mathcal{S}$ sends ϱ to $\mathcal{A}$. Else, $\mathcal{S}$ randomly selects $\varrho \in Z_p^*$, sends ϱ to $\mathcal{A}$, and adds (σ, ϱ) to $Hash_5$ list.

$\mathcal{S}$ then keeps the lists as follows:

(i) Key^{List} stores the tuples (β, i, sk_i);

(ii) $ReKey^{List}$ stores the tuples $(\beta_1, i, S', W', rkey_{i,S',W'}, \sigma, R, FlagOne)$, which maintains the results of query $ReKGen(sk_i, S', W')$. If $FlagOne = 1$, then it is a valid re-encryption key, or else it is random.

(iii) $REnc^{List}$ stores the tuples $(i, S, S', C, C_R, FlagTwo)$, which maintains query $REnc(i, S, S', C)$. Note that if $FlagTwo = 1$, then a re-encryption key which is used to generate the re-encrypted ciphertext should be valid, or else the re-encrypted ciphertext is invalid and randomly generated.

(1) **Init.** The attacker selects a challenge user set $S^* \subseteq \{1, 2, \ldots, n\}$ and challenge conditional set $W^* = \{\omega_1^*, \omega_2^*, \ldots, \omega_n^*\}$.

(2) **Setup.** The simulator $\mathcal{S}$ randomly selects values $\varpi, \in Z_p$ and $Z \in G$ and sets

$$\upsilon = \nu^{\varpi} \cdot \left(\prod_{j \in S^*} \nu_{n+1-j} \right)^{-1} \triangleq \nu^{\gamma}. \tag{22}$$

$\mathcal{S}$ sets the public key as

$$PK = \left(\upsilon, \nu, \nu_1, \ldots, \nu_n, \nu_{n+2}, \ldots, \nu_{2n}, Z, Hash_1, Hash_2, Hash_3, Hash_4, Hash_5\right) \tag{23}$$

and $sk = \gamma$. $\mathcal{S}$ sends PK to the attacker $\mathcal{A}$.

(3) **Query phase 1.** $\mathcal{A}$ does some of the columns of the query, and the response of $\mathcal{S}$ is as follows:

(i) $Extract(i)$: If $i \in S^*$, then $\mathcal{S}$ aborts. Else the simulator $\mathcal{S}$ first searches Key^{List}, and if (β, i, sk_i) are in Key^{List}, then it outputs sk_i. Otherwise, $\mathcal{S}$ throws a biased coin β ($Pr[\beta = 1] = \delta$ for some δ).

(a) When $\beta = 0$, $\mathcal{S}$ aborts and outputs a random bit.

(b) When $\beta = 1$, $\mathcal{S}$ calculates $sk_i = \nu_i^{\mu} \cdot (\prod_{j \in S^*} \nu_{n+1-j+i})^{-1}$. Note that we have

$$sk_i = \nu_i^{\varpi} \cdot \left(\prod_{j \in S^*} \nu_{n+1-j+i} \right)^{-1} = \left(\nu^{\varpi} \cdot \left(\prod_{j \in S^*} \nu_{n+1-j} \right)^{-1} \right)^{\alpha^i} = \upsilon^{\alpha^i} = \nu_i^{\gamma}. \tag{24}$$

(ii) $ReKGen(i, S', W')$: the simulator $\mathcal{S}$ verifies whether tuple $(*, j, sk_j)$ is in Key^{List} ($i \in S^*$, $j \in S'$, and $|W' \cap W^*| \geq d$). If not, then $\mathcal{S}$ aborts. Otherwise $\mathcal{S}$ searches whether there is a tuple $(*, i, S', W', rkey_{i,S',W'}, \sigma, R, *)$ in $ReKey^{List}$. If yes, $\mathcal{S}$ outputs $rkey_{i,S',W'}$. Otherwise, the simulator $\mathcal{S}$ proceeds:

(a) If $(1, i, sk_i)$ exists in Key^{List}, as in the real algorithm of the scheme, the simulator $\mathcal{S}$ generates the re-encryption key $rkey_{i,S',W'}$ from $ReKGen$ by using the secret key sk_i. $\mathcal{S}$ returns $rkey_{i,S',W'}$ to $\mathcal{A}$. Then it puts $(*, i, S', W', rkey_{i,S',W'}, \sigma, R, 1)$ to $ReKey^{List}$, where σ, R are randomly selected.

(b) Otherwise, $\mathcal{S}$ throws a biased coin β. If $\beta = 1$, $\mathcal{S}$ queries the $Extract(i)$ oracle to get sk_i and, then, generates $rkey_{i,S',W'}$ from $ReKGen$. $\mathcal{S}$ returns the re-encryption key $rkey_{i,S',W'}$ to $\mathcal{A}$. Then it adds $(1, i, sk_i)$ and $(*, i, S', W', rkey_{i,S',W'}, \sigma, R, 1)$ to Key^{List} and $ReKey^{List}$, respectively. If $\beta = 0$, $\mathcal{S}$ sets $\{a = (a_\omega = \rho_\omega), b = (b_\omega = \rho'_\omega) : \omega \in W'\}$ for randomly chosen $\rho_\omega, \rho'_\omega \in G$. Then $\mathcal{S}$ constructs $rkey_1, rkey_2, rkey_3, rkey_4, rkey_5$ to encrypted random σ, R as the real environment. The simulator $\mathcal{S}$ sends the re-encryption key to $\mathcal{A}$. Then it adds $(*, i, S', W', rkey_{i,S',W'}, \sigma, R, 0)$ to $ReKey^{List}$.

(iii) $REnc(i, S, S', C)$: $\mathcal{S}$ searches $(i, S, S', C, C_R, *)$ in $REnc^{List}$. If yes, $\mathcal{S}$ returns C_R. Else, it is dealt with in the below:

(a) If $(*, i, S', rkey_{i,S',W'}, \sigma, R, *)$ in $ReKey^{List}$, as the real environment, $\mathcal{S}$ uses $rkey_{i,S',W'}$ to construct C_R by $REnc$ and sends C_R to $\mathcal{A}$. Then it adds $(i, S, S', C, C_R, *)$ to $REnc^{List}$. Here we need $C = Encrypt(PK, S, m, W)$ and $|W \cap W'| \geq d$.

(b) Otherwise, $\mathcal{S}$ first makes $ReKGen(i, S')$ query to retrieve the re-encryption key $rkey_{i,S',W'}$. Next, $\mathcal{S}$ constructs C_R. Then it puts $(i, S, S', C, C_R, *)$ to $REnc^{List}$.

(iv) $DecryptL2(i, S, C)$: $\mathcal{S}$ tests (11)-(13). If the test does not pass, it aborts with an error $\perp$. Else

(a) If $(1, i, sk_i)$ already exists in Key^{List}, $\mathcal{S}$ recovers m by using sk_i.

(b) Otherwise, $\mathcal{S}$ queries $Extract(i)$ to obtain sk_i and $\mathcal{S}$ recovers m by using sk_i.

(v) $DecryptL1(i, j, S, S', C_R)$: $\mathcal{S}$ checks (18)-(19). If one of the equations cannot be validated, it aborts and outputs $\perp$. Otherwise, $\mathcal{S}$ proceeds:

(a) If $(1, j, sk_j)$ already exists in Key^{List}, $\mathcal{S}$ recovers m by using sk_j.

(b) Otherwise, $\mathcal{S}$ queries $Extract(j)$ to get sk_j and $\mathcal{S}$ recovers m by using sk_j.

(4) **Challenge.** Once the attacker $\mathcal{A}$ decides that Phase 1 is finished, it outputs two messages (m_0, m_1). And $\mathcal{S}$

randomly selects $b \in \{0,1\}, R^* \in G_T$. Let $\mu = g^{t^*}$ for some randomly chosen t^*. $\mathcal{S}$ computes

$$
\begin{aligned}
C_1^* &= R^* \cdot T,\\
C_2^* &= \mu = \nu^{t^*},\\
C_3^* &= \mu^{\omega} = \nu^{\omega t^*}\\
&= \left(\nu^{\omega} \cdot \left(\prod_{j\in S^*} \nu_{n+1-j}\right)^{-1} \left(\prod_{j\in S^*} \nu_{n+1-j}\right)\right)^{t^*}\\
&= \left(\upsilon \prod_{j\in S^*} \nu_{n+1-j}\right)^{t^*},\\
C_4^* &= \left(C_\omega = \mu^{\tau} = \left(\nu^{\tau}\right)^{t^*} = Hash_4(\omega)^{t^*}\right)_{\omega\in W^*},\\
C_5^* &= m_b \oplus Hash_2\left(R^*\right),\\
\varphi^* &= Hash_3\left(C_1^*, C_3^*, C_3^*, C_4^*, C_5^*\right),\\
C_6^* &= \mu^{\phi^*} = Hash_3\left(C_1^*, C_3^*, C_3^*, C_4^*, C_5^*\right)^{t^*}
\end{aligned}
\tag{25}
$$

where $\mathcal{S}$ queries $(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$ to RO $Hash_3$ for entry $(C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, \phi^*, \varphi^*)$. $\mathcal{S}$ sends $C^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*)$ to $\mathcal{A}$. If $T = e(\nu_{n+1}, \mu)$, we have $C_1^* = R^* \cdot T = R^* \cdot e(\nu, \nu_{n+1})^{t^*}$. Hence the challenge ciphertext C^* is a valid one. Otherwise, C^* is independent of b if T is a random value, in the attacker's view.

(5) **Query phase 2.** $\mathcal{A}$ makes queries continuously.

(6) **Guess.** $\mathcal{A}$ generates the guess bit b', if $b' = b$, and then outputs 1, which means that $T = e(\nu_{n+1}, \mu)$; else outputs 0. In this case, T is a random value in G_2.

□

Probability analysis. When $\mathcal{S}'s$ aborting happens, we define the game as *Abort*. In the case $\mathcal{S}$ does not abort, $\mathcal{A}$ does not know the different between the game and the actual scheme. Let q denote the total number of all queries; we have $Pr[\neg Abort] \geq \delta^q \cdot ((p-1)/p)^q \triangleq \xi^q \geq \xi^q(1-\xi)$, which is maximized at $\delta_{opt} = q/(1+q)$. Using δ_{opt}, the probability $Pr[\neg Abort]$ is at least $1/\dot{e}(1+q)$, where $\dot{e}$ is the base of the nature logarithm. Therefore, we have

$$
\begin{aligned}
\epsilon' \geqslant \frac{\epsilon}{\dot{e}(1+q)} &- Adv^{TCR}_{Hash_1,\mathcal{A}} - Adv^{TCR}_{Hash_2,\mathcal{A}}\\
&- Adv^{TCR}_{Hash_3,\mathcal{A}} - Adv^{TCR}_{Hash_4,\mathcal{A}} - Adv^{TCR}_{Hash_5,\mathcal{A}}.
\end{aligned}
\tag{26}
$$

With this we complete the proof of Lemma 4.

Lemma 5. *If an IND-Re-CCA attacker can successfully attack our scheme, then a simulator $\mathcal{S}$ that can be constructed to solve the Decisional nBDHE assumption.*

Proof. We use the same game construction from the proof of Lemma 4 and modify the challenge phase to prove Lemma 5. To construct the challenge re-encrypt ciphertext, $\mathcal{S}$ randomly selects $b \in \{0,1\}, R^* \in G_T$. Let $h = g^{t^*}$ for the randomly selected t^*. $\mathcal{S}$ then computes

$$
\begin{aligned}
\widetilde{C_1}^* &= R^* \cdot e(\mu, Z)^{Hash_5(\sigma)},\\
C_2^* &= \mu = \nu^{t^*},\\
C_5^* &= m_b \oplus Hash_2\left(R^*\right),
\end{aligned}
\tag{27}
$$

$\mathcal{S}$ then constructs $rkey_1^*, rkey_2^*, rkey_3^*, rkey_4^*, rkey_5^*$ in the same way as in game 1.

The proofs of Lemmas 4 and 5 conclude that we have proven Theorem 3. □

6. Comparison

We use schemes from [24, 26] as baselines as [26] achieves the same security and [24] supports the same fuzzy property with our scheme. In the implementation, we selected two most efficient schemes to do experimental comparisons. Our experimental environment is as follows: Core i7 Processor (6M Cache, 3.40 GHz) with a Linux operating system. In the implementation, the proxy re-encrypted the ciphertext of the delegator to 20 different ciphertext for the delegate. The average value of the execution time of 50 experiments is used to eliminate the errors. Table 1 lists the comparison of the performance of the schemes. Although *ReKGen*, *Enc*, *REnc*, *DecryptL2*, and *DecryptL1* time is a little greater than [24, 26] as our scheme needs to support the property of broadcast, the time overhead of the re-encryption algorithm is much slower than our scheme. This is due to the fact that the proxy is only required to run the re-encryption algorithm once.

7. Application

With the proposed FC-PBRE scheme, we illustrate an example of the scheme's possible application and show how FC-PBRE scheme protects the confidentiality and privacy of multimedia data in the Internet of Things.

7.1. Application of Security in Internet of Things. With the development of wireless networks and the Internet of Things, a large number of wireless devices (WiFi cameras, wireless sensors, etc.) are deployed; they are collecting and analyzing multimedia data at all times. Many applications require these wireless devices to share data, and insecure data sharing before wireless devices can easily lead to data leakage. Usually a normal user accesses a wireless device in three ways: first, through direct connection with the wireless device; second, through the gateway; third, through the cloud platform. Because many wireless devices need to be localized before they start to work, they usually have backend configuration interfaces (WEB, PC) and are offline at some point during their work, so it is more convenient for an attacker to access multimedia data directly

Table 1: Comparison with Weng et al. [26] and Fang et al. [24].

Scheme	ReKGen(ms)	Enc(ms)	REnc(ms)	DecryptL2(ms)	DecryptL1(ms)
Weng et al.	9.22	5.64	40.32	5.09	6.27
Fang et al.	11.46	6.03	36.62	6.31	7.13
Our scheme	12.42	7.01	28.08	7.54	8.36

by cracking the login password of the backend configuration interface. In this case, even if the device and cloud have been authenticated and encrypted (such as using SSL) they cannot prevent the attack. How to ensure the safe storage and sharing of wireless multimedia data becomes particularly important. In this environment, we can deploy FC-PBRE to ensure the security of the system. The data of each device (including multimedia data, control data) is encrypted by the device owner's public key ($C_A = Encrypt(PK_A, S, m, W)$). Obviously, no user can decrypt the data except the data owner A. When the device owner A needs to grant data privileges to the maintenance engineer for the configurations of wireless devices (such as WiFi cameras), the maintenance engineer B has to request a re-encryption key $rk_{A,S',W'} = ReKGen(PK, sk_A, S', W')$, where $BinS'$ from the device owner A. The maintenance engineer B can use the re-encrypted key to perform the re-encryption algorithm $C_R = REnc(PK_A, rk_{i,S',W'}, i, S, S', C_A)$ to convert the device owner's ciphertext C_A into their own ciphertext C_R, and then B uses his own public key to decrypt the ciphertext by performing the re-encrypted ciphertext decryption algorithm $m = DecryptL1(PK_A, sk_B, A, B, S, S', C_R)$. Similarly, if the gateway or cloud platform wishes to share encrypted multimedia data to authorized third-party users (i.e., $E = E_1, E_2, \ldots, E_n$) without decryption, the device owner A only needs to produce a re-encrypted key $rk_{A,E,W'} = ReKGen(PK_A, sk_A, E, W')$, which will be used to re-encrypt the message from the device owner A to the third-party user sets $E = E_1, E_2, \ldots, E_n$. After the key $rk_{A,E,W'}$ is generated, it will be sent to the cloud platform. Therefore, the cloud platform can run the re-encryption algorithm $C_R = REnc(PK_A, rk_{i,E,W'}, A, S, S', C_A)$ to convert the device owner's ciphertext into C_R, which can be decrypted with the third party's own private key.

8. Conclusions

In the paper, a new security notion called fuzzy-conditional proxy broadcast re-encryption (FC-PBRE) is presented. In a FC-PBRE, the proxy uses a broadcast re-encryption key to re-encrypt a ciphertext which can be decrypted by a set of delegatees if and only if the broadcast key's conditional set W is close to the conditional set W' of the ciphertext. Moreover, the proxy learns nothing about the plaintext from entire process. Second, we define the security notion against chosen-ciphertext attacks for FC-PBRE and propose an efficient fuzzy-conditional proxy broadcast re-encryption scheme. Finally, we prove that our FC-PBRE scheme is the chosen-ciphertext attack secure in the random oracle model under the Decisional nBDHE assumption.

Given our contributions, further research might explore constructing a CCA-secure FC-PBRE scheme in the standard model. It also can focus on the construction of the fuzzy-conditional proxy broadcast re-encryption schemes without pairings.

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

This paper is an extended version of the oral report [27] the authors made at the International Conference on Cloud Computing and Security (ICCCS 2018). However, this paper has not been published by the conference ICCCS 2018.

Conflicts of Interest

The authors declare that the funding in the Acknowledgments did not lead to any conflicts of interest regarding the publication of this manuscript. Also, there is no conflicts of interest in the manuscript.

Acknowledgments

Liming Fang is supported by the National Natural Science Foundation of China (nos. 61872181, 61702236, and 61300236), the National Natural Science Foundation of Jiangsu (under Grant no. BK20130809), the National Science Foundation for Postdoctoral Scientists of China (no. 2013M530254), the National Science Foundation for Postdoctoral Scientists of Jiangsu (no. 1302137C), China Postdoctoral Science Special Foundation (no. 2014T70518), and Changzhou Sci&Tech Program (no. CJ20179027). Jinyue Xia is partially supported by the National Natural Science Foundation of China (no. 6127208361300236).

References

[1] Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," *Journal of Internet Technology*, vol. 16, no. 2, pp. 317–323, 2015.

[2] Y. Ren, J. Shen, D. Liu, J. Wang, and J.-U. Kim, "Evidential quality preserving of electronic record in cloud storage," *Journal of Internet Technology*, vol. 17, no. 6, pp. 1125–1132, 2016.

[3] J. Kaur and K. Kaur, "A fuzzy approach for an IoT-based automated employee performance appraisal," *Computers, Materials and Continua*, vol. 53, no. 1, pp. 24–38, 2017.

[4] A. Pradeep, S. Mridula, and P. Mohanan, "High security identity tags using spiral resonators," *Computers, Materials and Continua*, vol. 52, no. 3, pp. 185–195, 2016.

[5] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[6] J. G. Li, W. Yao, Y. C. Zhang, H. L. Qian, and J. G. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 785–796, 2017.

[7] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User Collusion Avoidance CP-ABE With Efficient Attribute Revocation for Cloud Storage," *IEEE Systems Journal*, 2017.

[8] J. G. Li, X. N. Lin, Y. C. Zhang, and J. G. Han, "KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 715–725, 2017.

[9] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology—EUROCRYPT '98 (Espoo)*, vol. 1403 of *Lecture Notes in Computer Science*, pp. 127–144, Springer, Berlin, Germany, 1998.

[10] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[11] J. Weng, R. H. Deng, X. Ding, C. Chu, and J. Lai, "Conditional proxy re-encryption secure against chosen-ciphertext attack," in *Proceedings of the the 4th International Symposium*, p. 322, Sydney, Australia, March 2009.

[12] C.-K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, "Conditional proxy broadcast re-encryption," in *Information Security and Privacy*, C. Boyd and J. G. Nieto, Eds., vol. 5594 of *Lecture Notes in Computer Science*, pp. 327–342, Springer, Berlin, Germany, 2009.

[13] Q. Tang, "Type-based proxy re-encryption and its construction," in *Progress in cryptology—INDOCRYPT 2008*, vol. 5365 of *Lecture Notes in Comput. Sci.*, pp. 130–144, Springer, Berlin, 2008.

[14] J. Shao, Z. Cao, X. Liang, and H. Lin, "Proxy re-encryption with keyword search," *Information Sciences*, vol. 180, no. 13, pp. 2576–2587, 2010.

[15] L. Fang, W. Susilo, C. Ge, and J. Wang, "Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search," *Theoretical Computer Science*, vol. 462, pp. 39–58, 2012.

[16] L. Fang, W. Susilo, and J. Wang, "Anonymous Conditional Proxy Re-encryption without Random Oracle," in *Provable Security*, vol. 5848 of *Lecture Notes in Computer Science*, pp. 47–60, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[17] C. Ge, W. Susilo, J. Wang, Z. Huang, L. Fang, and Y. Ren, "A key-policy attribute-based proxy re-encryption without random oracles," *The Computer Journal*, vol. 59, no. 7, pp. 970–982, 2016.

[18] K. Liang, W. Susilo, and J. K. Liu, "Privacy-preserving ciphertext multi-sharing control for big data storage," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1578–1589, 2015.

[19] G. Ateniese, K. Benson, and S. Hohenberger, "Key-private proxy re-encryption," in *Topics in cryptology—CT-RSA 2009*, vol. 5473 of *Lecture Notes in Comput. Sci.*, pp. 279–294, Springer, Berlin, 2009.

[20] J. Shao, P. Liu, G. Wei, and Y. Ling, "Anonymous proxy re-encryption," *Security and Communication Networks*, vol. 5, no. 5, pp. 439–449, 2012.

[21] J. Shao, P. Liu, and Y. Zhou, "Achieving key privacy without losing CCA security in proxy re-encryption," *The Journal of Systems and Software*, vol. 85, no. 3, pp. 655–665, 2012.

[22] Q. Zheng, W. Zhu, J. Zhu, and X. Zhang, "Improved anonymous proxy re-encryption with CCA security," in *Proceedings of the the 9th ACM symposium*, pp. 249–258, Kyoto, Japan, June 2014.

[23] J. Shao, "Anonymous ID-Based Proxy Re-Encryption," in *Information Security and Privacy*, vol. 7372 of *Lecture Notes in Computer Science*, pp. 364–375, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[24] L. Fang, J. Wang, C. Ge, and Y. Ren, "Fuzzy conditional proxy re-encryption," *Science China Information Sciences*, vol. 56, no. 5, 052116, 13 pages, 2013.

[25] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Proceedings of the International Conference on Theory and Applications of Cryptographic Techniques*, pp. 440–456, 2005.

[26] J. Weng, M. Chen, Y. Yang, R. Deng, K. Chen, and F. Bao, "CCA-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles," *Science China Information Sciences*, vol. 53, no. 3, pp. 593–606, 2010.

[27] L. Fang, L. Liu, J. Xia, and M. Sun, "A secure multimedia data sharing scheme for wireless network," *Oral report at ICCCS 2018*, 2018.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 6395362, 8 pages
https://doi.org/10.1155/2018/6395362

Research Article

A Source Hiding Identity-Based Proxy Reencryption Scheme for Wireless Sensor Network

Chunpeng Ge,[1] **Jinyue Xia**,[2] **Aaron Wu**,[3] **Hongwei Li**,[4] **and Yao Wang**[4]

[1]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China*
[2]*IBM, 3039 E Cornwallis Rd., Research Triangle Park, NC 27709, USA*
[3]*College of Computer Sciences, University of Illinois at Urbana Champaign, 1205 W. Nevada St. MC-137, Urbana, USA*
[4]*College of Computer Engineering, Jiangsu University of Technology, Changzhou, Jiangsu 213000, China*

Correspondence should be addressed to Chunpeng Ge; gecp@jsut.edu.cn

Received 31 May 2018; Accepted 2 October 2018; Published 17 October 2018

Guest Editor: Zhaoqing Pan

Wireless sensor network (WSN), which extends the typical Internet environment to Internet of Things, has been deployed in various environments such as safety monitoring, intelligent transportation, and smart home. In a WSN, encryption is typically used to protect data that are stored in wireless devices. However some features like data sharing can be affected if the traditional encryption is used. A secure mechanism should support a gateway of the network to directly convert a user's encrypted data (encrypted pollution data) to a new user's encryption without exposing the underlying plaintext data during the whole sharing phase. In this work, a new source hiding identity-based proxy reencryption scheme (SHIB-PRE) is proposed to deal with the issue. The proposed SHIB-PRE scheme supports a proxy (gateway or cloud server) to transform a user's encrypted date to a new user's ciphertext as long as the proxy has the proxy reencryption key. In SHIB-PRE, the encrypted pollution data is kept secure from the proxy and the relationship between a source ciphertext and a reencrypted ciphertext is concealed from the outside eavesdropper. In this paper, we give an introduction to the definition of a source hiding identity-based proxy reencryption and its chosen plaintext security model. Further, a concrete construction will be presented and proven chosen plaintext secure under the $q - DDHE$ assumption in the standard model.

1. Introduction

With the growth of wireless sensor devices, people are facing a formidable problem of huge sensor data management and maintenance [1, 2]. One cost-effective and convenient approach to resolve this issue is to deploy the sensor data on the cloud, for example, IBM cloud [3] and Amazon AWS [4]. People can adopt data encryption as an intuitive defense to ensure data confidentiality on the cloud [5]. By encrypting the sensor data and saving on the cloud, however, sharing sensor data within the wireless sensor network is limited. As a result, traditional public key encryption only guarantees the confidentiality of wireless sensor data, yet it is frustrating with the data sharing functionality.

Considering the following scenario, we will need a secure mechanism that supports a gateway of the network to directly convert a user's encrypted data (encrypted pollution data) to a new user's encrypted data without revealing the underlying plaintext data. Suppose many wireless sensor nodes are deployed in a wireless pollution sensor network to monitor the campus air quality. All sensor nodes send their monitoring data to the sink node and then send to the cloud through the gateway. For the purpose of confidentiality, we could encrypt the monitoring data before sending it to the sink node. In some situations, the campus administrator Alice may want to cooperate with the government institute researcher Bob to analyze the environment. As the data is encrypted by Alice's public key, Bob cannot decrypt the encryption to get the underlying plaintext due to the fact that he does not access to Alice's private key. What we can do in this case is that let the campus administrator Alice fetch the secret data off the cloud and then reencrypt the data with Bob's public key. However, it can significantly increase Alice workload and violates the original intention of cloud

computing, leaving heavy workload to the cloud. What is worse is that Alice should be online all time during each sharing phase. Another native solution is that Alice can store the private key in cloud. Thus the cloud can perform the download-decrypt-reencrypt work instead of Alice. But, it may be a disaster if the cloud is disclosed as the attacker can use Alice's private key.

In addition to secure data sharing, another security requirement for above scenario is privacy preservation. If the government system is disclosed, the campus' identity should not be revealed. This privacy-preserving property enables that, even if the government system is assailed by an adversary, the adversary can not know who is sharing the data with the government system. This requires the relationship between the campus and the government system can not be revealed by an attacker.

Therefore, a new public key encryption mechanism is desired to support data sharing and privacy preservation at the same time. Enabling the confidentiality of data and preserving the privacy without losing efficiency [6] are an important problem to be issued. In this work, we focus on solving these elusive problems by presenting a novel notion of source hiding identity-based proxy reencryption. In our proposed source hiding identity-based proxy reencryption scheme, a proxy (gateway or cloud server) with a proxy reencryption key can convert a delegator's (campus) ciphertext to a delegate's (government institute researcher) ciphertext without exposing the plaintext. At the meanwhile, an outsider eavesdropper can not gain the relationship between the original ciphertext and the reencrypted ciphertext.

In related work, proxy reencryption (PRE) was proposed to enable a semitrusted proxy to convert Alice's ciphertext to Bob's ciphertext by a reencryption key [7]. Proxy reencryption has been applied into several places, such as secure email forwarding [7, 8] and cloud computing [9]. Green et al. [10] introduced identity-based proxy reencryption in which a user's public key is viewed as his identity. After their work, a great number of identity-based proxy reencryptions have came out [11–13] to deal with the efficiency and security property. An AB-PRE scheme was presented to apply attribute-based setting to proxy reencryption [14]. Luo, Hu, and Chen [15] revealed another scheme to provide "AND" gates on both positive and negative attributes. Later on, a ciphertext-policy attribute-based proxy reencryption (CPAB-PRE) [16, 17] was presented to support a monotonic access formula in the selective model. Further, they enhanced its security in the adaptive model [18]. Meanwhile, Ge et al. [19, 20] presented two key-policy attribute-based proxy reencryption (KPAB-PRE) schemes in both the selective and adaptive model, respectively. Recently, a DFA-based proxy reencryption scheme [21] allows the access to be described as a DFA. Unfortunately, none of these schemes support the functionality of privacy-preserving keyword search.

To capture the source hiding property, Emura, Miyaji, and Omote [22] introduced the notion of source hiding and they presented the first source hiding IB-PRE scheme in the random oracle model. However, their proof is only a heuristic argument and might lead to the scheme insecure [23]. Furthermore, the previous source hiding scheme [22] is found not collusion resistant. As a result, if a proxy colludes a set of delegates, the delegator's message is revealed as well as the delegator's private key.

1.1. Our Contribution. To address above problems [22], this work presents a CPA secure collusion resistant source hiding identity-based proxy scheme. Additionally, we prove the security without random oracles. More specifically, a proxy and a set of delegates can only collude to reveal the plaintext but not the delegator's private key. The paper organizes as follows: first we describe our scheme, second we prove our scheme secure in the standard model, and finally we show it is collusion resistant.

2. Preliminaries

2.1. Bilinear Map. G and G_T denote two multiplicative cyclic groups with the same prime order p. g is a generator of group G. A bilinear pairing is a bilinear map $e : G \times G \longrightarrow G_T$ with the following properties [24]:

(1) $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $a, b \overset{R}{\leftarrow} Z_p^*$ and $g_1, g_2 \in G$.

(2) $e(g, g) \neq 1$.

(3) There is an efficient algorithm to compute $e(g_1, g_2)$ for all $g_1, g_2 \in G$.

2.2. Complexity Assumption. Our proposed system security relies on the truncated q decisional Diffie-Hellman exponent (q-DDHE) assumption. Here is the assumption: given a vector of $q + 2$ elements

$$\left(g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, T\right) \in G^{q+2} \tag{1}$$

it is difficult to distinguish $T = g^{(\alpha^{q+1})}$ from a random value in G. Formally speaking, for all probability polynomial time adversaries $\mathcal{A}$, the following probability is negligible:

$$\left|\Pr\left[\alpha, r \overset{R}{\leftarrow} Z_p^*;\ T_0 = g^{(\alpha^{q+1})};\ T_1 = g^r;\ z \in \{0,1\}; z' \longleftarrow \mathcal{A}\left(g, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, T_z\right) : z = z'\right] - \frac{1}{2}\right|. \tag{2}$$

2.3. Identity-Based Proxy Reencryption. The encryption level in our paper is defined as follows: A "level 1" ciphertext is a ciphertext generated directly by the Encrypt algorithm. A "level $l + 1$" ciphertext is a reencryption result of a "level l" ciphertext by using the Reencryption algorithm. *MaxLevel* is the highest-possible ciphertext level. It is obvious that, for a single-hop IB-PRE scheme, *MaxLevel* = 2. In this paper, we deal with single-hop IB-PRE scheme, as the max level equals 2. In our scheme, the first and second level ciphertext denote the original and reencrypted ciphertext, respectively.

Definition 1 (identity-based proxy reencryption). The following algorithms describe a single-hop identity-based proxy reencryption scheme [10]:

(i) Setup(λ): the private key generator (PKG) runs setup with a security parameter λ input. This step generates the global public parameters PP and a master secret key msk.

(ii) KeyGen(msk, ID): in this step, KeyGen takes the master secret key msk and an identity ID as the input; it returns a private key sk_{ID} for identity ID.

(iii) Encrypt(ID, m): the input for this algorithm is an identity ID and a message $m \in M$ (M: message space); it generates the ciphertext C_{ID}.

(iv) RKeyGen(sk_{ID_i}, ID_i, ID_j): RKeyGen takes identities ID_i, ID_j and sk_{ID_i} and outputs the reencryption key $rk_{ID_i \longrightarrow ID_j}$.

(v) ReEncrypt($C_{ID_i}, rk_{ID_i \longrightarrow ID_j}$): a reencryption key $rk_{ID_i \longrightarrow ID_j}$ and a ciphertext C_{ID_i} corresponding to identity ID_i are the input; it returns the reencrypted ciphertext C_{ID_j}.

(vi) Decrypt(C_{ID}, sk_{ID}): given a private key sk_{ID} and a ciphertext C_{ID}, it outputs the plaintext m or it aborts with an error symbol $\perp$.

Correctness. Suppose $(PP, msk) \longleftarrow Setup(\lambda)$, $sk_{ID_i} \longleftarrow KeyGen(msk, ID_i)$, $sk_{ID_j} \longleftarrow KeyGen(msk, ID_j)$, and $rk_{ID_i \longrightarrow ID_j} \longleftarrow RKeyGen(sk_{ID_i}, ID_i, ID_j)$. The correctness of IB-PRE means that

$$\begin{aligned} &\Pr\left[Decrypt\left(C_{ID_i}, sk_{ID_i}\right) = m\right] = 1, \\ &\Pr\left[Decrypt\left(sk_{ID_j}, ReEncrypt\left(C_{ID_i}, rk_{ID_i \longrightarrow ID_j}\right)\right)\right. \\ &\quad \left. = m\right] = 1. \end{aligned} \tag{3}$$

2.4. Security Notion for Key-Private IB-PRE. We describe game-based security definition of source hiding IB-PRE in this section. Compared to the work presented in [22], our security model considers the indistinguishability of message against chosen-plaintext attack (IND-CPA) and the source hiding property of IB-PRE against chosen-plaintext attack (IND-SH-CPA).

Definition 2 (IND-CPA). A (single-use) source hiding IB-PRE scheme is IND-CPA secure if no probabilistic polynomial time (PPT) adversary $\mathscr{A}$ can win the game below with nonnegligible advantage. Next in the game, we assume λ is the security parameter and $\mathscr{B}$ is the game challenger.

(1) Setup: $\mathscr{B}$ runs the Setup(λ) algorithm to obtain the (PP, msk) and assigns PP to $\mathscr{A}$.

(2) Query phase 1:

(a) Extract(ID): run the KeyGen(msk, ID) algorithm to get sk_{ID} and return sk_{ID} to $\mathscr{A}$.

(b) RKExtract(ID_i, ID_j): run the RKeyGen(sk_{ID_i}, ID_i, ID_j) algorithm to get $rk_{ID_i \longrightarrow ID_j}$ and return $rk_{ID_i \longrightarrow ID_j}$ to $\mathscr{A}$.

(3) *Challenge.* Once $\mathscr{A}$ decides that phase 1 is finished, it outputs two equal length messages (m_0, m_1) and two challenge identities ID^*. The challenger $\mathscr{C}$ chooses a random bit $b \in \{0, 1\}$ and sends the challenge ciphertext $C^* = Encrypt(ID^*, m_b)$ to $\mathscr{A}$. The restrictions are that $\mathscr{A}$ has never made the following queries:

(i) $Extract(ID^*)$;

(ii) $RKExtract(ID^*, ID_j)$ and $Extract(ID_j)$.

(4) Query phase 2: $\mathscr{A}$ continues making queries. The queries are same as phase 1, except the followings:

(i) $Extract(ID^*)$;

(ii) $RKExtract(ID^*, ID_j)$ and $Extract(ID_j)$;

(5) Guess: $\mathscr{A}$ makes the guess b' and wins the game if $b' = b$.

We claim IB-PRE is IND-CPA secure, if the probability

$$Adv_{\mathscr{A}}^{IND-CPA}(\lambda) = \left|\Pr\left[b' = b\right] - \frac{1}{2}\right| \tag{4}$$

is negligible for all probabilistic polynomial time adversary $\mathscr{A}$.

Next, we present the source hiding property of IB-PRE (IND-SH-CPA) and we follow the security model of [22]. IND-SH-CPA guarantees that even if an adversary knows a mailing-list address and a mailing-list member address included in the mailing-list system, the adversary cannot identify whether a source ciphertext is the source of a destination ciphertext or not. We allow an adversary to select the challenge source identities ID_0^*, ID_1^* and the challenge ciphertext ID^*. An adversary $\mathscr{A}$ is provided the *Extract* and *RKExtract* queries as in the IND-CPA game.

(1) Setup: run the Setup(λ) algorithm to get the (PP, msk) and then assign PP to $\mathscr{A}$.

(2) Query phase 1:

(a) Extract(ID): $\mathscr{A}$ runs the KeyGen(msk, ID) algorithm to get sk_{ID} and obtain sk_{ID}.

(b) RKExtract(ID_i, ID_j): $\mathscr{A}$ runs the RKeyGen(sk_{ID_i}, ID_i, ID_j) algorithm to get $rk_{ID_i \longrightarrow ID_j}$ and obtain $rk_{ID_i \longrightarrow ID_j}$.

(3) Challenge: as soon as $\mathscr{A}$ considers phase 1 is over, it outputs two identities (ID_0, ID_1), a challenge plaintext m^* and a challenge identity ID, ID not in $\{ID_0, ID_1\}$. The challenger $\mathscr{C}$ chooses a random bit $b \in \{0, 1\}$ and computes $C_{ID_b}^* = Encrypt(ID_b^*, m^*)$. Next, $\mathscr{C}$ computes $C_{ID}^* = ReEncrypt(C_{ID_b}^*, rk_{ID_b^* \longrightarrow ID})$ and sends the challenge ciphertext C_{ID}^* to $\mathscr{A}$.

(4) Query phase 2: $\mathscr{A}$ continues making queries as in the query phase 1.

(5) Guess: $\mathscr{A}$ outputs the guess b'. The adversary wins if $b' = b$.

We say that a source hiding IB-PRE scheme is IND-SH-CPA secure, if the following probability is negligible for all probabilistic polynomial time adversary $\mathcal{A}$:

$$Adv_{\mathcal{A}}^{IND-SH-CPA}(\lambda) = \left|\Pr\left[b' = b\right] - \frac{1}{2}\right|. \quad (5)$$

Note that, unlike the IND-CPA security game, in the IND-SH-CPA security game, the adversary $\mathcal{A}$ is allowed to get the private key of the target ciphertext. The IND-SH-CPA guarantees that even if $\mathcal{A}$ can decrypt the challenge ciphertext C_{ID}^*, $\mathcal{A}$ only can obtain the following: (1) C_{ID}^* is encrypted under identity ID; (2) m^* is the plaintext, all of which however have been already known by $\mathcal{A}$.

3. Our Proposed Source Hiding IB-PRE

First, we analyze what conditions IB-PRE scheme should meet such that it has the source hiding property. Second, we describe our source hiding IB-PRE scheme and prove its IND-CPA and IND-SH-CPA security.

3.1. Impossibility Result for Source Hiding IB-PRE. Before presenting our scheme, we introduce several necessary yet not sufficient conditions that are satisfying the source hiding property.

Lemma 3. *As proven in [22], the adversary breaks the IND-SH-CPA security if he can learn to determine if destination ciphertexts are derived from the same source ciphertext or not.*

Lemma 4. *Any IB-PRE scheme, in which the ReEncrypt algorithm is deterministic, cannot satisfy source hiding.*

Proof. Suppose the *ReEncrypt* algorithm is deterministic, an adversary $\mathcal{A}$ can win the IND-SH-CPA game as below. Suppose the source ciphertext is $C_{ID_0}^*$ and $C_{ID_1}^*$ and the challenge ciphertext is C_{ID}^*. The adversary works as follows:

(1) Makes a $RKExtract(ID_1, ID)$ query and get the reencryption key $rk_{ID_1 \longrightarrow ID}$.

(2) Using the reencryption key $rk_{ID_1 \longrightarrow ID}$, run the deterministic algorithm $ReEncrypt(C_{ID_1}^*, rk_{ID_1 \longrightarrow ID}) \longrightarrow C'$.

(3) If $C' = C_{ID}^*$, it outputs 1, else returns 0.

It is not difficult to see that $\mathcal{A}$ can succeed with an overwhelming probability. □

3.2. Our Construction. Let G and G_T be bilinear group of prime order p, and g be a generator of G. Additionally, let $e : G \times G \longrightarrow G_T$ denote the bilinear map. The proposed scheme contains the following steps:

(i) Setup(λ): λ is the security parameter, and (p, g, G, G_T, e) are the bilinear map parameters. The PKG chooses random generators $g, h \in G$, random value $\alpha \in Z_p$, and a collusion resistant hash function $H : G_T \longrightarrow Z_p^*$. It sets $g_1 = g^\alpha \in G$. The PKG keeps h secret and outputs the public parameters PP. So master secrets are set as

$$PP = \left(g, g_1, e\left(g, h\right), H\right) \quad msk = \alpha. \quad (6)$$

(ii) KeyGen(msk, ID): in this step, the PKG picks a random value $r_{ID} \in Z_p$ to compute a private key for $ID \in Z_p$. It calculates $h_{ID} = (hg^{-r_{ID}})^{1/(\alpha - ID)}$ and the private key

$$sk_{ID} = \left(r_{ID}, h_{ID}\right). \quad (7)$$

If $\alpha = ID$, the PKG aborts.

(iii) Encrypt(ID, m): the input are an identity ID and a message $m \in G_T$. In this step, the sender picks a random value $s \in Z_p$ and sets

$$\begin{aligned} C_1 &= g_1^s g^{-s \cdot ID}, \\ C_2 &= g^s, \\ C_3 &= m \cdot e\left(g, h\right)^{-s}. \end{aligned} \quad (8)$$

Outputs the ciphertext $C = (C_1, C_2, C_3)$.

(iv) RKeyGen(sk_{ID_i}, ID_i, ID_j): on input identities ID_i, ID_j and the secret key sk_{ID_i}, the reencryption key $rk_{ID_i \longrightarrow ID_j}$ is generated as follows:

(1) Choose random values $\theta \in G_T$ and $s' \in Z_p$, and compute $rk_1 = (g_1^{s'} g^{-s' \cdot ID_j})$, $rk_2 = g^{s'}$, $rk_3 = \theta \cdot e(g, h)^{-s'}$.

(2) Choose a random value $\rho \in Z_p$, and set $rk_4 = r_{ID_i} \cdot H(\theta) + \rho$, $rk_5 = h_{ID_i}^{H(\theta)}$, $rk_6 = g^\rho$, and $rk_7 = e(g, h)^{H(\theta)}$.

(3) Output the reencryption key $rk_{ID_i \longrightarrow ID_j} = (rk_1, rk_2, rk_3, rk_4, rk_5, rk_6, rk_7)$.

(v) ReEncrypt($C_{ID_i}, rk_{ID_i \longrightarrow ID_j}$): on input a reencryption key $rk_{ID_i \longrightarrow ID_j} = (rk_1, rk_2, rk_3, rk_4, rk_5, rk_6, rk_7)$ and a ciphertext $C_{ID_i} = (C_1, C_2, C_3)$ under identity ID_i, the proxy proceeds as follows:

(1) Compute $C_3' = (e(C_1, rk_5) \cdot e(g, C_2)^{rk_4}) / e(C_2, rk_6)$.

(2) Choose a random value $t \in Z_p$ and compute

$$\begin{aligned} \widetilde{C_3'} &= C_3' \cdot rk_7^t, \\ \widetilde{C_3} &= C_3 \cdot e\left(g, h\right)^{-t}. \end{aligned} \quad (9)$$

(3) Choose a random value $t' \in Z_p$ and compute

$$\begin{aligned} R_1 &= rk_1 \cdot g_1^{t'} g^{-t' \cdot ID_j}, \\ R_2 &= rk_2 \cdot g^{t'}, \\ R_3 &= rk_3 \cdot e\left(g, h\right)^{-t'}. \end{aligned} \quad (10)$$

(4) Output the reencrypted ciphertext $C_{ID_i \longrightarrow DI_j} = (R_1, R_2, R_3, \widetilde{C_3'}, \widetilde{C_3})$.

(vi) Decrypt(C_{ID}, sk_{ID}):

(a) If C_{ID} is an original ciphetext, let $sk_{ID} = (r_{ID}, h_{ID})$ and $C_{ID} = (C_1, C_2, C_3)$. Compute

$$m = C_3 \cdot e\left(C_1, h_{ID}\right) \cdot e\left(g, C_2\right)^{r_{ID}}. \tag{11}$$

(b) If C_{ID} is a reencrypted ciphetext, let $C_{ID} = (R_1, R_2, R_3, \widetilde{C_3'}, \widetilde{C_3})$. Compute

$$\begin{aligned} \theta &= R_3 \cdot e\left(R_1, h_{ID}\right) \cdot e\left(g, R_2\right)^{r_{ID}}, \\ m &= \widetilde{C_3} \cdot \left(\widetilde{C_3'}\right)^{1/H(\theta)}. \end{aligned} \tag{12}$$

Correctness. The correctness of the proposed scheme is defined as follows:

(1) For an original ciphertext $C = (C_1, C_2, C_3)$, we have

$$\begin{aligned} &C_3 \cdot e\left(C_1, h_{ID}\right) \cdot e\left(g, C_2\right)^{r_{ID}} \\ &= C_3 \cdot e\left(g_1^s g^{-s \cdot ID}, \left(h g^{-r_{ID}}\right)^{1/(\alpha - ID)}\right) \cdot e\left(g, g^s\right)^{r_{ID}} \\ &= C_3 \cdot e\left(g^{s(\alpha - ID)}, \left(h g^{-r_{ID}}\right)^{1/(\alpha - ID)}\right) \cdot e\left(g, g\right)^{s r_{ID}} \\ &= C_3 \cdot e\left(g^s, h g^{-r_{ID}}\right) \cdot e\left(g, g\right)^{s r_{ID}} = m. \end{aligned} \tag{13}$$

(2) For a reencrypted ciphertext $C_{id} = (C_3, \widetilde{C_3}, rk^{(4)})$, we have

$$\begin{aligned} R_1 &= rk_1 \cdot g_1^{t'} g^{-t' \cdot ID_j} = g_1^{s'} g^{-s' \cdot ID_j} \cdot g_1^{t'} g^{-t' \cdot ID_j} \\ &= g_1^{s'+t'} g^{-(s'+t') \cdot ID_j} \triangleq g_1^{\Delta t'} g^{-\Delta t' \cdot ID_j}, \\ R_2 &= rk_2 \cdot g^{t'} = g^{s'} \cdot g^{t'} = g^{\Delta t'}, \\ R_3 &= rk_3 \cdot e\left(g, h\right)^{-t'} = \theta \cdot e\left(g, h\right)^{-s'} \cdot e\left(g, h\right)^{-t'} \\ &= \theta \cdot e\left(g, h\right)^{-\Delta t'}, \\ C_3' &= \frac{e\left(C_1, rk_5\right) \cdot e\left(g, C_2\right)^{rk_4}}{e\left(C_2, rk_6\right)} \\ &= \frac{e\left(g^{s(\alpha - ID)}, \left(h g^{-r_{ID}}\right)^{H(\theta)/(\alpha - ID)}\right) \cdot e\left(C_2, rk_6\right)}{e\left(g^s, g^\rho\right)} \\ &= \frac{e\left(g^s, \left(h g^{-r_{ID}}\right)^{H(\theta)}\right) \cdot e\left(g, g^s\right)^{r_{ID} \cdot H(\theta) + \rho}}{e\left(g^s, g^\rho\right)} \\ &= e\left(g, h\right)^{sH(\theta)}, \\ \widetilde{C_3'} &= C_3' \cdot rk_7^{\mathrm{t}} = e\left(g, h\right)^{sH(\theta)} \cdot e\left(g, h\right)^{tH(\theta)} \\ &= e\left(g, h\right)^{(s+t)H(\theta)} \triangleq e\left(g, h\right)^{\Delta t \cdot H(\theta)} \\ \widetilde{C_3} &= C_3 \cdot e\left(g, h\right)^{-t} = m \cdot e\left(g, h\right)^{-s} \cdot e\left(g, h\right)^{-t} \\ &= m \cdot e\left(g, h\right)^{-\Delta t}. \end{aligned} \tag{14}$$

Finally, we have $\widetilde{C_3} \cdot (\widetilde{C_3'})^{1/H(\theta)} = m \cdot e(g,h)^{-\Delta t} \cdot e(g,h)^{\Delta t \cdot H(\theta)/H(\theta)} = m$.

3.3. Security of Our Source Hiding IB-PRE Scheme

Theorem 5. *Our scheme is IND-CPA secure without random oracles under the q-DDHE assumption.*

Proof. Assuming there exists an adversary $\mathscr{A}$ that can break our scheme's IND-CPA security with the probability ε, we can construct an algorithm $\mathscr{B}$ that can solve the q-DDHE problem with probability ε', where

$$\varepsilon' \geq \frac{\varepsilon}{\dot{e}\left(1 + q_e\right)}. \tag{15}$$

$\mathscr{B}$ inputs a q-DDHE instance $(g, A_1 = g^\alpha, A_2 = g^{\alpha^2}, \ldots, A_q = g^{\alpha^q}, T)$ and has to distinguish $T = A_{q+1} = g^{\alpha^{q+1}}$ from a random element in G.

The approach to prove Theorem 5 follows the steps of the security proof of Gentry's scheme [25]. Note $\mathscr{B}$ maintains a list of tables that are empty initialized. Here is the list:

(i) K^{List}: it keeps the secret keys tuples (β, ID, sk_{ID}).

(ii) RK^{List}: it maintains the result of the queries to RKExtract(ID_i, ID_j) which are the tuples $(ID_i, ID_j, rk_{ID_i \longrightarrow ID_j}, flag)$. In the tuples, $flag = 1$ represents the reencryption key which is a valid one, while $flag = 0$ represents the reencryption key which is a random value.

(1) **Setup**: $\mathscr{B}$ generates a random polynomial $f(x) \in Z_p[x]$ of degree q. It sets $h = g^{f(\alpha)}$, computing h from $(g, A_1, \ldots, A_q)$. $\mathscr{B}$ also picks a collusion resistant hash function $H : G_T \longrightarrow Z_p^*$. It sends the public key $(g, A_1, e(g,h), H)$ to $\mathscr{A}$. With this assignment, the master secret key msk is α. This assignment has a distribution identical to that in the actual construction since g, α, $f(x)$, and h are uniformly random.

(2) **Query phase 1**: $\mathscr{A}$ sends a bunch of queries to $\mathscr{B}$, and $\mathscr{B}$ responds as follows:

(a) *Extract*(ID): $\mathscr{B}$ searches K^{List}, if $(1, ID, sk_{ID})$ exists in K^{List}, then $\mathscr{B}$ obtains sk_{ID}. Otherwise, $\mathscr{B}$ generates a biased coin β so that $\Pr[\beta = 1] = \delta$ for some δ that can be determined later.

(i) If $\beta = 0$, $\mathscr{B}$ aborts and returns a random bit.

(ii) If $\beta = 1$, if $ID = \alpha$, we have that $\Pr[ID = \alpha] = 1/p$, $\mathscr{B}$ uses α to solve the q-DDHE problem. Else, let $F_{ID}(x)$ denote the $q-1$ degree polynomial $(f(x) - f(ID))/(x - ID)$. $\mathscr{B}$ returns the

private key $sk_{ID} = (r_{ID}, h_{ID}) = (f(ID), g^{F_{ID}(\alpha)})$ to the adversary and adds $(1, ID, sk_{ID})$ to K^{List}. Note that $g^{F_{ID}(\alpha)} = g^{(f(\alpha)-f(ID))/(\alpha-ID)} = (hg^{-f(ID)})^{1/(\alpha-ID)}$, which is identical to the actual construction.

(b) $RKExtract(ID_i, ID_j)$: $\mathcal{B}$ first searches whether there is a tuple $(ID_i, ID_j, rk_{ID_i \longrightarrow ID_j}, *)$ in RK^{List}. If yes, $\mathcal{B}$ returns $rk_{ID_i \longrightarrow ID_j}$ ($*$ denotes the wildcard). Otherwise, $\mathcal{B}$ proceeds as follows:

(i) If $(1, ID_i, sk_{ID_i})$ exists in K^{List}, $\mathcal{B}$ uses sk_{ID_i} to compute the reencryption key $rk_{ID_i \longrightarrow ID_j}$ by running $RKeyGen$. $\mathcal{B}$ returns $rk_{ID_i \longrightarrow ID_j}$ to $\mathcal{A}$ and adds $(ID_i, ID_j, rk_{ID_i \longrightarrow ID_j}, 1)$ to RK^{List}.

(ii) Otherwise, $\mathcal{B}$ flips a biased coin β. If $\beta = 1$, $\mathcal{B}$ queries the $Extract(ID_i)$ oracle to obtain sk_{ID_i} and then computes $rk_{ID_i \longrightarrow ID_j}$ from $RKeyGen$ algorithm. $\mathcal{B}$ returns $rk_{ID_i \longrightarrow ID_j}$ to $\mathcal{A}$ and adds $(1, ID_i, sk_{ID_i})$ and $(ID_i, ID_j, rk_{ID_i \longrightarrow ID_j}, 1)$ to K^{List} and RK^{List}, respectively. If $\beta = 0$, $\mathcal{B}$ first selects a random $\theta \in G_T$ and computes rk_1, rk_2, rk_3 as the $Encrypt$ algorithm. Next $\mathcal{B}$ computes $rk_4 = \sigma, rk_5 = \phi_1, rk_6 = \phi_2, rk_7 = e(g,h)^{H(\theta)}$ for randomly chosen $\sigma \in Z_p, \phi_1, \phi_2 \in G$. $\mathcal{B}$ forwards the reencryption key to $\mathcal{A}$ and adds $(ID_i, ID_j, rk_{ID_i \longrightarrow ID_j}, 0)$ to RK^{List}.

(3) **Challenge**: once $\mathcal{A}$ has decided that query phase 1 is over, it outputs two equal length plaintexts (m_0, m_1) and a challenge identity ID^*. If $(1, ID^*)$ exists in K^{List}, $\mathcal{B}$ outputs a random bit and aborts. Else if $\alpha = ID^*$, $\mathcal{B}$ uses α to solve the q-DDHE problem. Else $\mathcal{B}$ generates a random bit $b \in \{0,1\}$ and computes a private key (r_{ID^*}, h_{ID^*}) as in phase 1. Let $f_2(x) = x^{q+2}$ and $F_{2,ID^*}(x) = (f_2(x) - f_2(ID^*))/(x - ID^*)$; $\mathcal{B}$ sets

$$\begin{aligned} C_1^* &= g^{f_2(\alpha)-f_2(ID^*)}, \\ C_2^* &= T \cdot \prod_{i=0}^{q} g^{F_{2,ID^*,i} \cdot \alpha^i}, \\ C_3^* &= \frac{m_b}{\left(e\left(C_1^*, h_{ID^*}\right) \cdot e\left(g, C_2^*\right)^{r_{ID^*}}\right)}, \end{aligned} \tag{16}$$

where $F_{2,ID^*,i}$ is the coefficient of x^i in $F_{2,ID^*}(x)$. It sends the challenge ciphertext (C_1^*, C_2^*, C_3^*) to $\mathcal{A}$.

Note that, let $s^* = F_{2,ID^*}(\alpha)$. If $T = A_{q+1} = g^{\alpha^{q+1}}$, we have

$$C_1^* = g^{f_2(\alpha)-f_2(ID^*)} = g^{F_{2,ID^*}(\alpha)\cdot(\alpha-ID^*)} = g_1^{s^*} g^{-s^* \cdot ID^*},$$

$$\begin{aligned} C_2^* &= T \cdot \prod_{i=0}^{q} g^{F_{2,ID^*,i} \cdot \alpha^i} = g^{\alpha^{q+1}} \cdot \prod_{i=0}^{q} g^{F_{2,ID^*,i} \cdot \alpha^i} \\ &= g^{(f_2(\alpha)-f_2(ID^*))/(\alpha-ID^*)} = g^{F_{2,ID^*}(\alpha)} = g^{s^*} \end{aligned}$$

$$\begin{aligned} C_3^* &= \frac{m_b}{\left(e\left(C_1^*, h_{ID^*}\right) \cdot e\left(g, C_2^*\right)^{r_{ID^*}}\right)} \\ &= \frac{m_b}{\left(e\left(g_1^{s^*} g^{-s^* \cdot ID^*}, h_{ID^*}\right) \cdot e\left(g, g^{s^*}\right)^{r_{ID^*}}\right)} \\ &= m_b \cdot e(g,h)^{-s^*}. \end{aligned} \tag{17}$$

Thus, (C_1^*, C_2^*, C_3^*) is a valid ciphertext for (ID^*, m_b).

(4) **Query phase 2**: $\mathcal{A}$ continues querying as in the query phase 1 except for the restrictions described in the IND-CPA game.

(5) **Guess**: $\mathcal{A}$ outputs the guesse $b' \in \{0,1\}$. If $b' = b$, $\mathcal{B}$ outputs 1 meaning $T = g^{\alpha^{q+1}}$; else output 0 meaning T is a random value in G.

Probability Analysis. If $\mathcal{B}$ does not abort, $\mathcal{A}$'s view is identical to the actual scheme. Abort is defined to be the event of $\mathcal{B}'s$ aborting during the simulation of $Extract$ query. Let q_e denote the total number of $Extract$ queries; we have $\Pr[\neg Abort] \geq \delta^{q_e} \cdot ((p-1)/p)^{q_e} \triangleq \xi^{q_e} \geq \xi^{q_e}(1-\xi)$, which is maximized at $\delta_{opt} = q_e/(1+q_e)$. Using δ_{opt}, the probability $\Pr[\neg Abort]$ is at least $1/\dot{e}(1+q_e)$, where $\dot{e}$ is the base of the nature logarithm. Therefor, we have $\varepsilon' \geq \varepsilon/\dot{e}(1+q_e)$.

This completes the proof of Theorem 5. □

Theorem 6. *Our proposed scheme is IND-SH-CPA secure in the information theoretic sense.*

Proof. Since a source identity ID_i is not included in a destination ciphertext, Theorem 6 is clearly satisfied. $(R_1, R_2, R_3, \widetilde{C_3'}, \widetilde{C_3})$ as $R_1 = g_1^{\Delta t'} g^{-\Delta t' \cdot ID_j}$, $R_2 = g^{\Delta t'}$, $R_3 = \theta \cdot e(g,h)^{-\Delta t'}$, $\widetilde{C_3'} = e(g,h)^{\Delta t \cdot H(\theta)}$, and $\widetilde{C_3} = m \cdot e(g,h)^{-\Delta t}$, where ID_j is a destination ciphertext, namely, a part of source ciphertext C_3 is randomized using a random value t. More precisely, for $C_{ID_j} = (R_1, R_2, R_3, \widetilde{C_3'}, \widetilde{C_3})$ and all identity ID, there exists a ciphertext $C_{ID} = (g_1^s g^{-s \cdot ID}, g^s, m \cdot e(g,h)^{-s})$ which can be a source ciphertext of C_{ID_j}.

This completes the proof of Theorem 6. □

4. Performance and Comparison

4.1. Efficiency Theoretical Analysis. To compare the performance of our scheme, we choose the existing source hiding IB-PRE scheme [22] as the base. We make the comparison in the aspect of the public/private key size, reencryption key size, level 1/level 2 ciphertext size, reencryption key generation cost, reencryption cost, and security model. Table 1 illustrates the detailed comparison. To construct a fair comparison, we choose Emura, Miyaji, and Omote's first scheme denotes EMO 1 scheme [22], which is also CPA secure with source hiding. Let c_e, c_p represent the computational cost of an exponentiation and a pairing cost, respectively, $|Z_p|$, $|G|$, $|G_T|$ denote the bit-length of an element in Z_p, G, G_T, respectively, and $|H|$ denotes the size of a hash function.

Table 1: Efficiency and security comparison.

Schemes	EMO scheme [22]	Our IB-PRE scheme
Public/Private key size	$2\|G\| + 2\|H\| /\|Z_p\|$	$2\|G\| + \|G_T\| + \|H\| /\|Z_p\|$
Re-encryption key size	$2\|G\| + \|G_T\|$	$4\|G\| + 2\|G_T\| + \|Z_p\|$
Level 1/Level 2 ciphertext size	$\|G\| + \|G_T\| /2\|G\| + 2\|G_T\|$	$2\|G\| + \|G_T\| /2\|G\| + 3\|G_T\|$
Rekey generation/ Re-encryption cost	$3c_e + c_p /4c_e + 3c_p$	$7c_e /7c_e + 3c_p$
Without RO?	×	√
Collusion resistant	×	√

Table 2: Execute time comparison.

Algorithms	KeyGen (ms)	Enc (ms)	RKenGen (ms)	ReEnc (ms)	Dec(Or) (ms)	Dec(Re) (ms)
scheme [22]	3.279	4.662	6.441	4.971	5.082	6.506
Our scheme	3.795	5.103	6.061	9.017	5.602	7.032

From Table 1, we found that, although the ciphertext size of our scheme is a little larger than the scheme of [22] in terms of the computational cost. However, the computational cost is the same order of magnitude. Most of important, our scheme is collusion resistant and without relying on random oracle.

4.2. Execute Time. Now we compare the proposed scheme with the existing source hiding IB-PRE scheme [22] regarding the execute time. For the scheme implementation, we use the Pairing Based Cryptography Library [26] to calculate the implementation time. Our Hardware is Intel(R) Core(TM) i5-8250U CPU @ 1.60GHZ 8GB RAM. The operation system is Linux Mint 18.1 Serena and programming language is GO 1.9. The elliptic curve $Y^2 = X^3 + X$ and the group order is 160 bits which are selected for the experiment. In our experiment we run each experiment for 20 times to obtain the average execution time.

From Table 2, it is observable that the execution time of *KeyGen*, *Enc*, *RKeyGen*, *ReEnc*, *Dec*(*Or*), and *Dec*(*Or*) of our scheme is a little more than scheme [22]. This coincides with the theoretical analysis.

5. Conclusions

In this paper, we introduced a new source hiding identity-based proxy reencryption scheme (SHIB-PRE) which is proposed to support a gateway of the wireless network to directly convert a user's encrypted data (encrypted pollution data) to a new user's encrypted data without exposing the underlying plaintext data during the whole sharing phase. Additionally, our SHIB-PRE scheme addresses the open problems left by Emura, Miyaji, and Omote [22] by presenting collusion resistant, source hiding, and against chosen ciphertext-plaintext attack secure in the standard model. Still, interesting questions are remained to be resolved and can be our future work, such as the following:

CCA-Secure. Designing a source hiding IB-PRE scheme that is chosen ciphertext secure is necessary. The technique described in [27] might be the potential approach to achieve CCA-secure.

Key-Private IB-PRE. The property of source hiding protects the source identity from a destination ciphertext. It will be challenging to design a key-private IB-PRE, in which a source identity and a destination identity are not disclosed from a reencryption key. The technique presented in [28] could be the potential approach to achieve a key-private IB-PRE scheme.

Data Availability

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

Conflicts of Interest

The authors declare that the funding in Acknowledgments section did not lead to any conflicts of interest regarding the publication of this manuscript. Also, there is no any other conflicts of interest in the manuscript.

Acknowledgments

Chunpeng Ge is supported by the National Natural Science Foundation of China (no. 61702236) and Changzhou Sci&Tech Program (no. CJ20179027), Jinyue Xia is partially supported by the National Natural Science Foundation of China (no. 6127208361300236), and Hongwei Li is partially supported by the National Natural Science Foundation of China (no. 61702216).

References

[1] J. Cui, Y. Zhang, Z. Cai, A. Liu, and Y. Li, "Secring display path for security-sensitive applications on mobile devices," *Computers, Materials & Continua*, vol. 55, no. 1, pp. 17–35, 2018.

[2] A. Pradeep, S. Mridula, and P. Mohanan, "High security identity tags using spiral resonators," *Computers, Materials and Continua*, vol. 52, no. 3, pp. 185–195, 2016.

[3] IBM, "Ibm smart cloud," http://ibm.com/cloudcomputing/.

[4] Amazon, "Amazon web services (aws)," http://aws.amazon.com.

[5] Z. Xiangyang, D. Hua, Y. Xun, Y. Geng, and L. Xiao, "MUSE: An Efficient and Accurate Verifiable Privacy-Preserving Multi-keyword Text Search over Encrypted Cloud Data," *Security and Communication Networks*, vol. 2017, 2017.

[6] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[7] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology—EUROCRYPT '98 (Espoo)*, vol. 1403 of *Lecture Notes in Computer Science*, pp. 127–144, Springer, Berlin, Germany, 1998.

[8] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[9] Y. Ren, J. Shen, D. Liu, J. Wang, and J.-U. Kim, "Evidential quality preserving of electronic record in cloud storage," *Journal of Internet Technology*, vol. 17, no. 6, pp. 1125–1132, 2016.

[10] J. Katz and M. Yung, "Identity-based proxy re-encryption," in *Proceedings of the nternational Conference on Applied Cryptography and Network Security*, pp. 288–306, 2007.

[11] C. K. Chu and W. G. Tzeng, "Identity-based proxy re-encryption without random oracles," in *International Conference on Information Security*, pp. 189–202, 2007.

[12] K. Liang, Z. Liu, X. Tan, D. S. Wong, and C. Tang, "A CCA-Secure Identity-Based Conditional Proxy Re-Encryption without Random Oracles," in *Information Security and Cryptology – ICISC 2012*, vol. 7839 of *Lecture Notes in Computer Science*, pp. 231–246, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[13] C. Ge, W. Susilo, J. Wang, and L. Fang, "Identity-based conditional proxy re-encryption with fine grain policy," *Computer Standards & Interfaces*, vol. 52, pp. 1–9, 2017.

[14] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *Proceedings of the 4th International Symposium on ACM Symposium on Information, Computer and Communications Security (ASIACCS '09)*, pp. 276–286, March 2009.

[15] S. Luo, J. Hu, and Z. Chen, "Ciphertext policy attribute-based proxy re-encryption," in *Information and Communications Security*, M. Soriano, S. Qing, and J. López, Eds., vol. 6476 of *Lecture Notes in Computer Science*, pp. 401–415, Springer, Berlin, Germany, 2010.

[16] K. Liang, L. Fang, W. Susilo, and D. S. Wong, "A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security," in *Proceedings of the 5th IEEE International Conference on Intelligent Networking and Collaborative Systems, INCoS 2013*, pp. 552–559, China, September 2013.

[17] N. Helil and K. Rahman, "CP-ABE access control scheme for sensitive data set constraint with hidden access policy and constraint policy," *Security and Communication Networks*, vol. 2017, 2017.

[18] Z. Liu and D. S. Wong, "Practical ciphertext-policy attribute-based encryption: traitor tracing, revocation, and large universe," in *Applied Cryptography and Network Security*, vol. 9092 of *Lecture Notes in Comput. Sci.*, pp. 127–146, Springer, [Cham], 2015.

[19] C. Ge, W. Susilo, J. Wang, Z. Huang, L. Fang, and Y. Ren, "A key-policy attribute-based proxy re-encryption without random oracles," *The Computer Journal*, vol. 59, no. 7, pp. 970–982, 2016.

[20] C. Ge, W. Susilo, L. Fang, J. Wang, and Y. Shi, "A CCA-secure key-policy attribute-based proxy re-encryption in the adaptive corruption model for dropbox data sharing system," *Designs, Codes and Cryptography. An International Journal*, vol. 86, no. 11, pp. 2587–2603, 2018.

[21] K. Liang, M. H. Au, J. K. Liu et al., "A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1667–1680, 2014.

[22] K. Emura, A. Miyaji, and K. Omote, "An identity-based proxy re-encryption scheme with source hiding property, and its application to a mailing-list system," *European Public Key Infrastructure Workshop*, vol. 6711, pp. 77–92, 2010.

[23] C. Ran, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited (preliminary version)," in *Proceedings of ACM Symposium on Theory of Computing*, pp. 209–218, 1998.

[24] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[25] C. Gentry, "Practical identity-based encryption without random oracles," in *Advances in cryptology—EUROCRYPT*, vol. 4004 of *Lecture Notes in Comput. Sci.*, pp. 445–464, Springer, Berlin, 2006.

[26] "Library P. Pbc library," http://github.com/Nik-U/pbc.

[27] C. Ran, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 207–222, 2004.

[28] J. Shao, P. Liu, and Y. Zhou, "Achieving key privacy without losing CCA security in proxy re-encryption," *The Journal of Systems and Software*, vol. 85, no. 3, pp. 655–665, 2012.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 5264526, 9 pages
https://doi.org/10.1155/2018/5264526

Research Article

An Ensemble Learning Method for Wireless Multimedia Device Identification

Zhen Zhang,[1] Yibing Li,[1] Chao Wang,[1] Meiyu Wang,[1] Ya Tu,[1] and Jin Wang[2]

[1]*College of Information and Communication Engineering, Harbin Engineering University, Harbin, 150001, China*
[2]*School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha 410114, China*

Correspondence should be addressed to Jin Wang; jinwang@csust.edu.cn

Received 27 May 2018; Revised 9 August 2018; Accepted 27 September 2018; Published 15 October 2018

Guest Editor: Weizhi Meng

In the last decade, wireless multimedia device is widely used in many fields, which leads to efficiency improvement, reliability, security, and economic benefits in our daily life. However, with the rapid development of new technologies, the wireless multimedia data transmission security is confronted with a series of new threats and challenges. In physical layer, Radio Frequency Fingerprinting (RFF) is a unique characteristic of IoT devices themselves, which can difficultly be tampered. The wireless multimedia device identification via Radio Frequency Fingerprinting (RFF) extracted from radio signals is a physical-layer method for data transmission security. Just as people's unique fingerprinting, different Internet of Things (IoT) devices exhibit different RFF which can be used for identification and authentication. In this paper, a wireless multimedia device identification system based on Ensemble Learning is proposed. The key technologies such as signal detection, RFF extraction, and classification model are discussed. According to the theoretical modeling and experiment validation, the reliability and the differentiability of the RFFs are evaluated and the classification results are shown under the real wireless multimedia device environments.

1. Introduction

With the rapid development of the wireless multimedia technologies, the security of wireless multimedia data transmission becomes increasingly more and more important. The design of efficient identification and authentication algorithms among different wireless multimedia devices has become an urgent subject.

As is well known, the Internet of Things [1] and the wireless sensor network (WSN) [2–4] are important carriers of multimedia data transmission; they will lead to improved efficiency, reliability, security, and economic benefits in our daily life [5]. At the same time, wireless multimedia technology and application have been widely studied by the researchers all over the world [6, 7]. However, because of the opening of transmitting channels, compared with the traditional wired network, the wireless network is more vulnerable to large-scale malicious attacks. Now, many existing networks are unprotected against a lot of different malicious attacks [8–10]; meanwhile the security of them is confronted with a series of new threats and challenges [11–13]. Traditional methods for protecting the security of wireless network are usually based on bit-level security protocol. However, there are usually loopholes in the actual wireless network security protocols [14]. For example, the IEEE 820.11 Wireless LAN's (WLAN) wired equivalent encryption protocol (WEP) is easily attacked by statistical analysis of [15]; although it has been upgraded to WPA and WPA2, its password can be restored, and there are still a variety of security problems [16]. The Radio Frequency Fingerprinting (RFF) is an inherent characteristic of wireless multimedia devices, which can hardly be tampered. In recent years, RFF extraction and identification methods for wireless multimedia devices have been widely studied [17–20].

The identification and authentication wireless multimedia device based on RFF is an important physical-layer method for wireless multimedia security [21–23], which has been widely used in intrusion detection [24], access control [25], wormhole detection [26], and cloning detection [27]. RFF is extracted from radio signals from wireless multimedia devices, which is a unique characteristic of wireless multimedia devices themselves and can difficultly be tampered. In

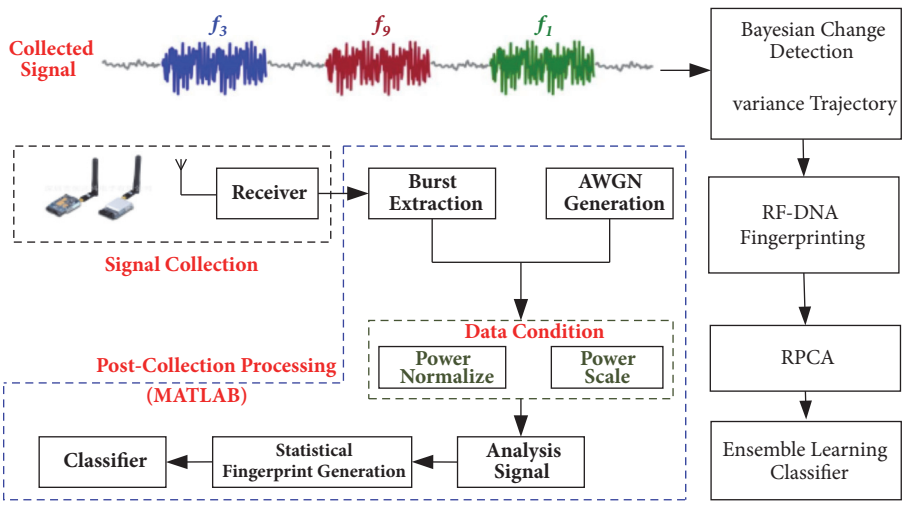


Figure 1: Wireless multimedia device identification process.

physical layer, RFF is just as people's unique fingerprinting; different wireless multimedia devices exhibit different RFF which can be used for identification and authentication. As is well known, RFF is derived hardware imperfection of wireless multimedia device, which can be observed and extracted. With the development of machine learning and the emergence of a large number of new technologies [28], new methods about RFF have been put forward continuously in recent years. Han J et al. [29] propose a physical-layer identification and authentication system for ultrahigh frequency (UHF) passive tags, which is called GenePrint. The classification accuracy of the passive tags is higher than 99.68 %. Furthermore, GenePrint can effectively defend against the replay attack. Huang G et al. [30] propose a novel Specific Emitter Identification (SEI) method based on nonlinear characteristics. The permutation entropy is calculated as the radio signal fingerprint for identifying the unique transmitter. Furthermore, the technology of bispectrum and stray parameter are used for the comparison with the new method, which indicates that the proposed method has a better performance in the classification of the wireless network cards. The PHY-based security based on Time Domain (TD) is also studied extensively in the recent years. Donald R et al. [31] demonstrate the performance of Dimensional Reduction Analysis (DRA) using discrete Gabor-Transform features, which are extracted from the Wi-Fi and WiMAX signals. Jia Y et al. [32] attempt to simultaneously find a low-rank representation matrix of original data and the optimal classifier parameter, which can be used to improve the performance of radiometric identification. Experiments indicate that the new method not only has a higher accurate classification and identification rate, but also has better robustness against noise.

In this paper, the structure of wireless multimedia device identification is proposed. Firstly, the main components of this structure are presented. Secondly, the key technologies such as signal collection, RFF generation, and classification model are discussed. Thirdly, according to the theoretical modeling and experiment validation, the differentiability of the RFFs is extracted. The classification result is shown under the real wireless multimedia device environments. Finally, the advantages and disadvantages of the proposed algorithm and its future prospects are described.

2. General View

In physical layer, the classification and identification of wireless multimedia device include four entities, which are shown in Figure 1: Acquisition Signal Module: acquiring the radio signals from wireless multimedia devices; Burst Extraction Module: detecting and intercepting the start of the turn-on transient; Signal Analysis Module: obtaining identification-relevant information from the radio signals; Fingerprint Generation Module: reducing assist information and generating the RFF; Classifier Module: a classifier for comparing RFF and requesting the identification of the comparison results [33, 34]. Furthermore, in order to better verify the identification performance of wireless multimedia device under different signals to noise ratio (SNR), the Additive White Gaussian Noise (AWGN) module and the data condition module are used in the experiments.

2.1. Data Set Definitions. As shown in Figure 2, the signals in this paper are mainly collected by Agilent oscilloscope, and 10 wireless multimedia devices of the same model and manufacturer are used for this research. In order to dislodge the influence of channel environment, wireless multimedia devices and receiving devices are connected directly with a cable (that is to say, ignoring the influence of multipath, time delay, and clutter in signal transmission process).

The sampling rate of the receiver is 40 MHz; each of the turn-on transients contains 159901 sampling points. For obtaining the complete turn-on transients, the Variance Trajectory (VT) [35] algorithm and Bayesian Change Detection (BCD) [36] are used for transient point detection and interception. The original dataset contains 500 transients from 10 devices, each of which contain 50 samples. According to the ratio of 2:3, the 500 signals are divided into training samples and test samples. At the same time, in order to

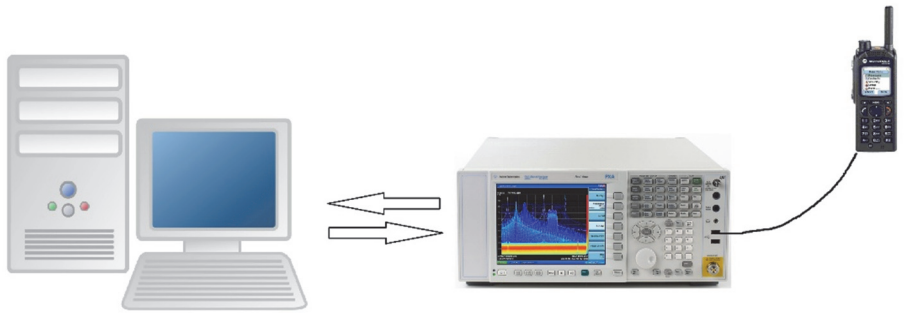


Figure 2: Schematic diagram of the experimental devices.

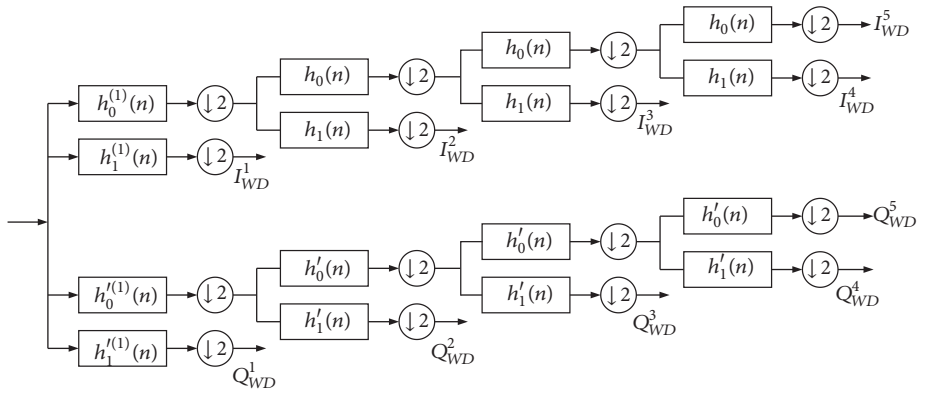


Figure 3: Four-stage (five-level) dual-tree complex wavelet transform (DT-CWT).

compare the performance of identification system under different SNR, the AWGN generation module, in Figure 1, is used in this paper. The noise is added with the simulation software (the range of SNR is 0~35dB, and the step is 1 dB).

2.2. Signal Analysis

2.2.1. Time Domain RF-DNA Fingerprinting. For TD fingerprinting, Radio Frequency-Distinct Native Attribute (RF-DNA) can be generated by the instantaneous amplitude, frequency, and phase response of radio signal's subsequence [37]. The unique features are obtained by the standard deviation (δ), variance (δ^2), skewness (γ), and kurtosis (κ) from $N_R + 1$ subsequence of the original signal, where N_R represents the number of the subsequence. The statistics can be arranged as follows:

$$F_{R_i} = \left[\delta_{R_i}, \delta_{R_i}^2, \gamma_{R_i}, \kappa_{R_i}\right]_{1\times 4} \tag{1}$$

Where $i = 1, 2, \ldots, N_R + 1$. Then, F_{R_i} can be used to generate the final TD fingerprint:

$$F^C = \left[F_{R_1} \;\; F_{R_2} \;\; F_{R_3} \;\; \cdots \;\; F_{R_{N_R+1}}\right]_{1\times 4(N_R+1)} \tag{2}$$

Where C refers to the signal's instantaneous parameter, including $\{a(n)\}$, $\{\varphi(n)\}$ and $\{f(n)\}$.

2.2.2. Wavelet Domain (WD) RF-DNA Fingerprinting. For signal analysis, the Discrete Wavelet Transform (DWT) is a very effective tool. But there are still some disadvantages. One distinct disadvantage of DWT is that it is not shift invariant. The DT-CWT is an improved method of DWT, which is used to overcome the disadvantage of DWT. The DT-CWT is commonly implemented by two real-valued filter banks, which is shown in Figure 3.

In Figure 3, the two filter banks represent two branches of Tree1 and Tree2, respectively, where the filter coefficients $h_1(t)$, $h_0(t)$, $h_1'(t)$, and $h_0'(t)$ are implemented directly as the Analysis Filters (AF) given in [38].

For real-valued input radio signals, the WD coefficients I_{WD}^L and Q_{WD}^L of Tree1 and Tree2 represent the real and imaginary components of complex coefficients [39]:

$$S_{WD}^l(n) = I_{WD}^l(n) + jQ_{WD}^l(n) \tag{3}$$

Then the WD fingerprints can be generated using the similar method [40] in Section 2.2.1.

2.3. Fingerprint Generation. The Robust Principle Component Analysis (RPCA) is an improved algorithm for traditional Principal Component Analysis (PCA). Because of the serious robustness problem of traditional PCA technology, the theoretical framework of RPCA was put forward to solve this problem.

Assuming that the observation matrix $\mathbf{D} = R^{(M*N)}$ is originally a low-rank matrix $\mathbf{A} = R^{(M*N)}$, it is polluted by matrix $\mathbf{E} = R^{(M*N)}$ which has sparse distribution and

arbitrarily large amplitude. The RPCA tries to separate the low-rank part from the sparse part of the observation matrix **D** and obtains the low-rank distribution matrix **A** and the sparse distribution matrix **E**, respectively. By increasing the constraints of low rank and sparsity of matrix **A** and matrix **E**, the sparse matrices can be computed by calculating the following convex optimization [41, 42]:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{E}} \quad & \|\mathbf{A}\| + \lambda \|\mathbf{E}\| \\ s.t. \quad & \mathbf{D} = \mathbf{A} + \mathbf{E} \end{aligned} \tag{4}$$

Where $\|\cdot\|_*$ is the kernel norm of matrix; it can also be understood as the sum of singular values in a matrix. $\lambda > 0$ is the tuning parameter [43] to balance the low-rank matrix and the sparse matrix.

In order to verify the performance, the dataset in Section 2.1 is used for simulation and evaluation. The dimensions are decreased based on RPCA analysis method. According to the contribution rate of main components, the first two principal components are selected for visualization.

3. Designed Classifier

3.1. Adaboost Algorithm. Adaboost algorithm is an iterative algorithm. It does not need to know the prior knowledge of weak classifiers. Instead, it changes the distribution of data and combines weak classifiers to achieve classification. This method has achieved good results in practical applications.

The specific description of the Adaboost algorithm is as follows.

Input: N labeled training samples $\{(1, c(1)), \ldots, (N, c(N))\}$, where $1, \ldots, N$ are serial numbers; $c(N)$ characterizes the different sample categories with a range of (0, 1); the training sample has a D distribution space and uses a weak learning algorithm; the number of iterations is set to T.

Initialization: if the sample $y_i = 0$, then make $w_{1,i} = 1/2m$; if the sample $y_i = 1$, then $w_{1,i} = 1/2l$;

Execution: the number of iterations is T, and each iteration performs the following steps.

(1) Normalized weights:

$$\frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}} \longrightarrow w_{t,i} \tag{5}$$

(2) Calculate the current error of weight:

$$\varepsilon_j = \sum_i w_{t,j} \left| h_j(x_i) - y_i \right| \tag{6}$$

Among them, h_j is a weak classifier generated by the feature j.

(3) Screening: the screening process is to add the smallest weak classifier in the previous step to the strong classifier we need.

(4) Update: regenerate each weight:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i} \tag{7}$$

Where $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$, and if the i^{th} sample x_i is correctly classified, then $e_i = 0$; otherwise $e_i = 1$.

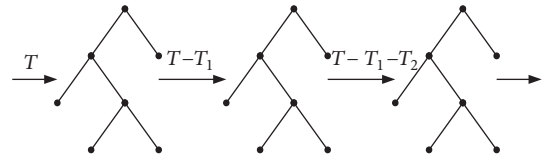


Figure 4: Schematic diagram of GBDT algorithm.

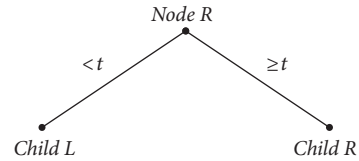


Figure 5: Binary classification.

(5) The resulting strong classifier is

$$h_j(x) = \begin{cases} 1, & \sum_{t=1}^{r} a_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{r} \alpha_t \\ 0, & \text{else} \end{cases} \tag{8}$$

In the above formula, $\alpha_i = \log(1/\beta_i)$.

3.2. Gradient Boosting Decision Tree. Gradient Boosting Decision Tree (GBDT) is a combination of decision tree and Boosting method. It is also one of the integrated learning Boosting algorithms [44]. In the GBDT algorithm, the decision tree training object is not a tree but an error in the classification of the previous decision tree. This is the Boosting concept. The principle of the gradient-based decision tree algorithm is shown in Figure 4.

As can be seen from Figure 4, GBDT is a kind of linear training, so it cannot be trained in parallel. In training, the difference between the training and the actual value is the target of the second tree optimization, as shown in formula T_2:

$$T = T_1 + T_2 + T_3 \tag{9}$$

From the idea of the algorithm shown in Figure 4, we can see that there is a difference between GBDT and traditional boosting. That is, the GBDT iteration is the residual or gradient descent value, and boosting is the sample data. Figure 5 is an example of using GBDT for binary classification.

The node R, which is about to be divided into two categories, can get μ on an average of several different y, and then use it as the output value of the node, as shown in equation ((3)-(4)):

$$\mu = \frac{\sum_{i=1}^{n} y_i}{n} \tag{10}$$

From this, the resulting optimization goal is the node error, which is shown in Equation ((3)-(5)):

$$Error = \sum \left(y_i - \mu \right)^2 \tag{11}$$

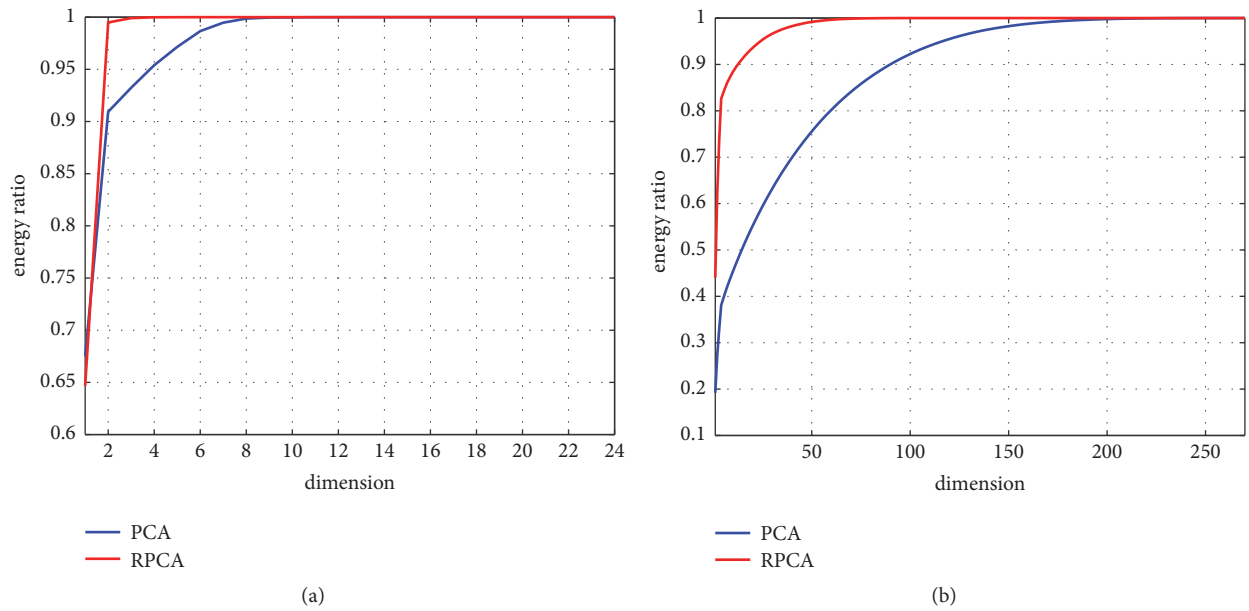


Figure 6: The energy ratio of original feature curves changing with dimensions. (a) Time domain RF-DNA fingerprinting. (b) Wavelet domain RF-DNA fingerprinting.

In the process of binary classification of decision trees, the criteria for classification are to select the attribute that maximizes the classification gain, and the calculation method for split gain is as follows:

$$G = S - S_j \tag{12}$$

In the above equation, to obtain the S_j, the loss function can be replaced by a variance, as follows:

$$S_j = \sum_{m \in L} (y_m - \mu_L)^2 + \sum_{n \in R} (y_n - \mu_R)^2 \tag{13}$$

After S_j is obtained, S and S_j are sequentially expanded, respectively, as in equation (12); then the split gain G can be obtained:

$$G = \left(\frac{Sum_L^2}{\|L\|} + \frac{Sum_R^2}{\|R\|} \right) - \frac{Sum^2}{\|Total\|} \tag{14}$$

In the above equations, Sum_L^2 and Sum_R^2 represent the square sum of samples on the left and right subtrees, while Sum^2 is used to represent the sum of the squares of all the samples. Therefore, to make the classification work best, G gets the maximum in each iteration. It should be noted that, in the GBDT algorithm, the purpose of each decision tree training is to optimize the residual of the previous item.

4. Simulation Result

4.1. Dimension Reduction Algorithm. In general, the features extracted by the feature extraction method have certain redundancy and noise. If the identification is performed directly, it will not only increase the computational complexity of the subsequent process and the requirements for computer memory, but also affect the final classification identification effect. Therefore, it is necessary to reduce the dimensionality of original feature set. Section 2.3 introduces the dimensionality reduction algorithms for PCA and RPCA, where RPCA is an improved algorithm for PCA. In Figure 6, when SNR=20dB, the energy ratio curves of the time domain and wavelet domain characteristics were obtained by using PCA and RPCA methods, respectively.

Table 1: The relationship between the energy ratio and the dimensionality reduction.

Energy ratio	85%	90%	95%
TD	-	-	3
WAV	6	13	24

The energy ratio refers to the ratio of feature vector information after dimension reduction to the feature vector information without dimension reduction. It can be seen from Figure 6 that RPCA has a better dimensionality reduction effect. Under these two feature extraction methods, a higher proportion of energy can be achieved in the same dimension. Considering these advantages of RPCA, the RPCA method is chosen in this paper to reduce the original features.

Table 1 shows the reduced dimension of several typical energy ratios in the use of time and wavelet domain feature extraction methods and the use of RPCA.

4.2. Classification and Identification. Experiments mainly selected the two ensemble learning classifiers described in

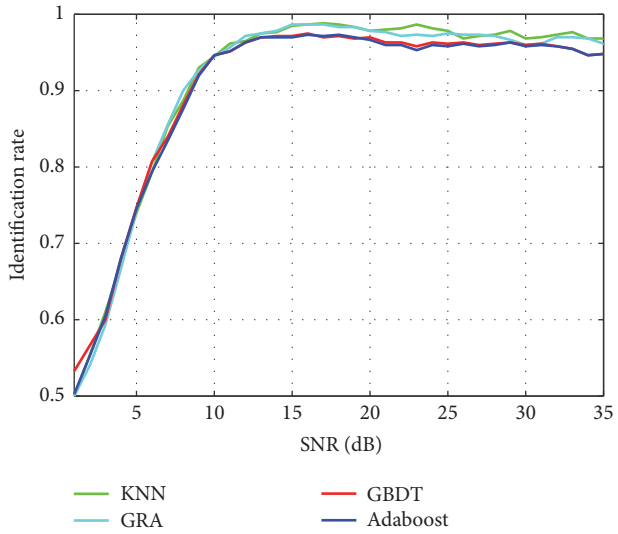


Figure 7: The identification rate of the authorized devices with 3-dimensional features based on (TD) RF-DNA fingerprint.

Section 3, Adaboost and GBDT. In order to verify the performance of classifier, k-Nearest Neighbor (KNN) and Grey relational analysis (GRA) classifier are used for comparison. The ensemble classifier depth is 5, learning rate is 0.3, and number is 50. Before classification and identification, it is necessary to determine the specific feature dimensions. According to the results in Table 1, we selected the dimensions corresponding to the energy proportions of 85%, 90%, and 95% compared with the original features, respectively. Among them, for the time domain feature set, since the energy proportion has exceeded 95% when it falls to three dimensions, the experiment is selected in 3 dimensions; for the feature set of the wavelet domain, 6, 13, 24, and 200 dimensional features are selected for testing.

From Figure 7, in the range of 0~35dB SNR, the identification rate increases with SNR. The four classifiers have the same performance; meanwhile the identification rate of different signals is basically stable at 10 dB; the identification rate is up to 95% in this condition.

We can obtain the same results as above for the classification and identification results of wavelet domain RF-DNA Fingerprinting in high SNR in Figure 8. However, the performance of time domain RF-DNA Fingerprinting is better in low SNR. Meanwhile, the performance of GBDT and Adaboost can achieve a higher identification rate compared with KNN and GRA, especially when the dimension of feature is high. At the same time, compared with the experimental results in Figure 7, it can be seen that time domain RF-DNA fingerprint features have better performance at low SNR. This is mainly because the data set in this paper is formed by the turn-on transient signal of wireless devices. It has a distinct envelope feature, and the time domain RF-DNA fingerprint is mainly based on the instantaneous amplitude phase and frequency characteristics, so it is easier to distinguish different devices. In order to verify the effect of dimension reduction after RPCA dimension reduction on overall identification rate, this article compares the overall identification rate of signals in the range of 0~35dB SNR in different dimensions, as shown in Figure 9.

As can be seen from Figure 9, with the increase of feature dimensions, the average identification rate of the signal shows a trend of increasing first and then decreasing. This is because when the input dimension is too low, the information carried by the feature is too small, and when the input dimension is too high, although there is more comprehensive original feature information, it will also increase noise and redundancy, which will lead to a reduction of identification rate.

5. Conclusion

With the popularity and development of wireless networks, the security of wireless networks has gradually become a research hotspot. Radio frequency fingerprinting technology is a kind of wireless network security technology based on the characteristics of physical layer. It can be used for the identification of most existing wireless multimedia devices, and it has a wide range of application scenarios. RF-DNA Fingerprinting technology is a brand-new RF Fingerprinting method developed in recent years, and it has good device classification identification effect. This paper proposes an RF-DNA Fingerprinting system based on ensemble learning and uses the RF-DNA Fingerprint feature based on time domain and wavelet domain to verify the classification and identification performance. In order to reduce the computational overhead and redundant information of the original features, in this paper, RPCA is introduced to reduce the dimensionality of original features. The experimental results show that it has good classification and identification performance. When the SNR is greater than 10 dB, the GBDT classifier is used to

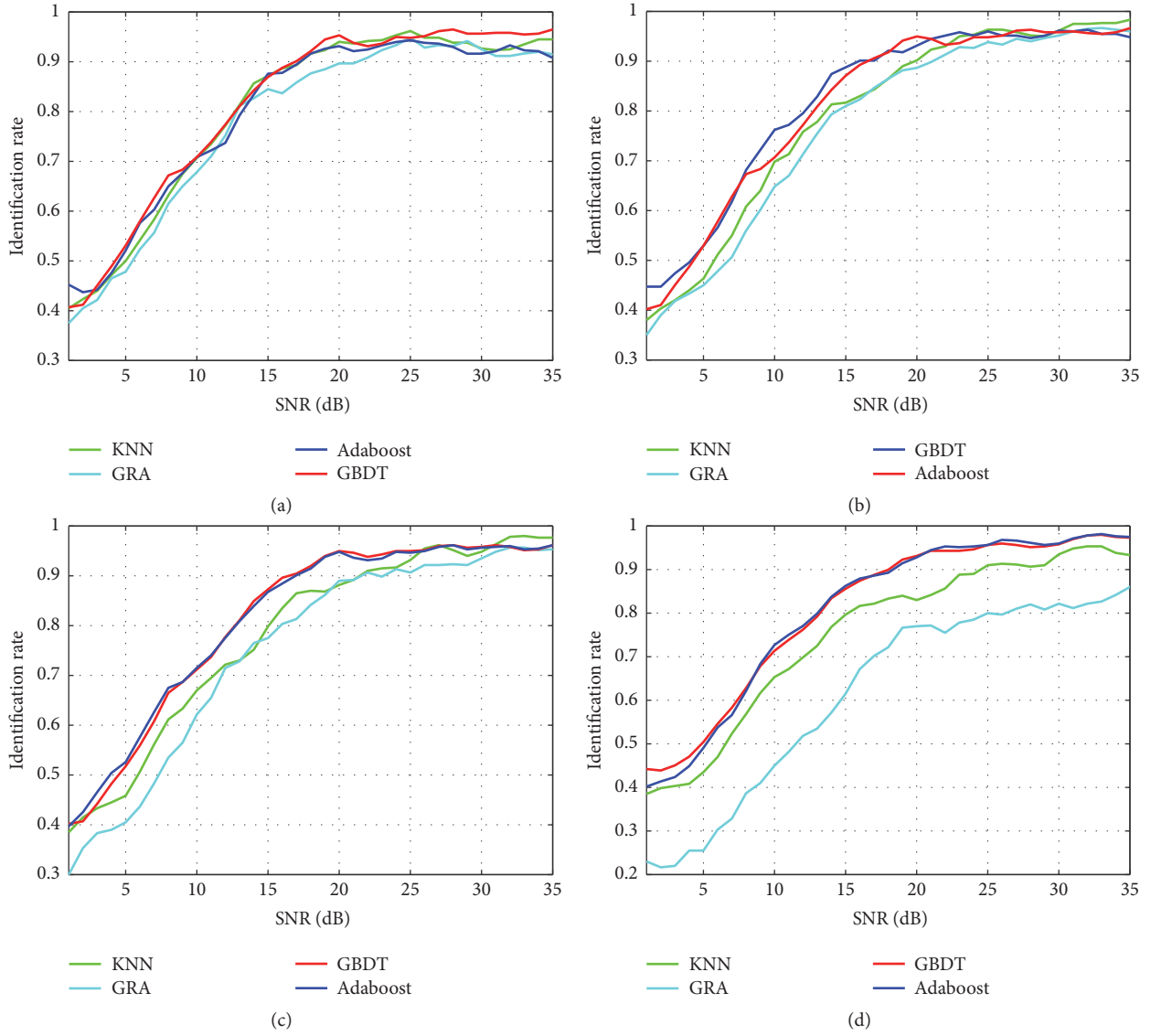


FIGURE 8: The identification rate of authorized devices under different dimensions. (a) 6-dimensional features, (b) 13-dimensional features, (c) 24-dimensional features, (d) 200-dimensional features.

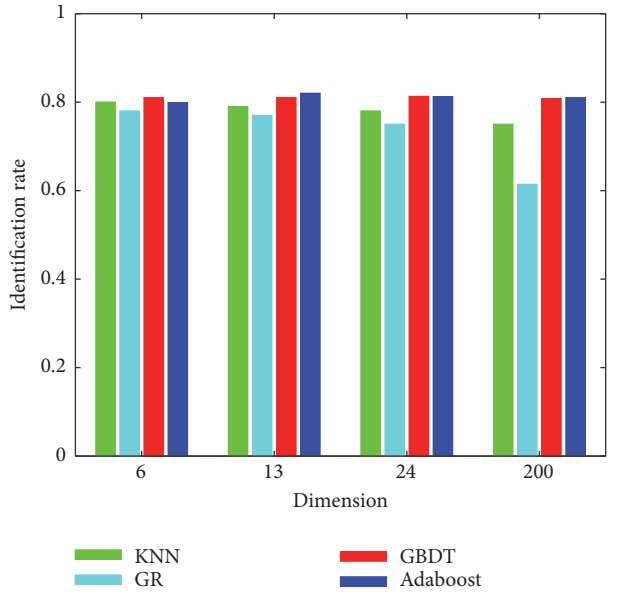


FIGURE 9: The identification results of four classifiers with SNR = 0 dB~20dB.

classify the RF-DNA features in the time domain and wavelet domain, and the identification rate can reach over 95%, which can realize the effective identification of different equipment types.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[2] F. Yunus, N. N. Ismail, S. H. Ariffin, A. A. Shahidan, N. Fisal, and S. K. Syed-Yusof, "Proposed transport protocol for reliable data transfer in wireless sensor network (WSN)," in *Proceedings of the 2011 Fourth International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*, pp. 1–7, Kuala Lumpur, Malaysia, April 2011.

[3] J. Wang, J. Cao, R. S. Sherratt, and J. H. Park, "An improved ant colony optimization-based approach with mobile sink for wireless sensor networks," *The Journal of Supercomputing*, pp. 1–13, 2017.

[4] J. Wang, J. Cao, S. Ji, and J. H. Park, "Energy-efficient cluster-based dynamic routes adjustment approach for wireless sensor networks with mobile sinks," *The Journal of Supercomputing*, vol. 73, no. 7, pp. 3277–3290, 2017.

[5] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[6] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[7] J. Han, R. Quan, D. Zhang, and F. Nie, "Robust object co-segmentation using background prior," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1639–1651, 2018.

[8] H. Wu and W. Wang, "A Game Theory Based Collaborative Security Detection Method for Internet of Things Systems," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1432–1445, 2018.

[9] X. Cao, D. M. Shila, Y. Cheng, Z. Yang, Y. Zhou, and J. Chen, "Ghost-in-ZigBee: Energy Depletion Attack on ZigBee-Based Wireless Networks," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 816–829, 2016.

[10] E. Y. Vasserman and N. Hopper, "Vampire attacks: Draining life from wireless ad Hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 318–332, 2013.

[11] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[12] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56, 2009.

[13] Y. Liu, M. Dong, K. Ota, and A. Liu, "ActiveTrust: secure and trustable routing in wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2013–2027, 2016.

[14] D. Macedonio and M. Merro, "A Semantic Analysis of Wireless Network Security Protocols," in *NASA Formal Methods*, vol. 7226 of *Lecture Notes in Computer Science*, pp. 403–417, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[15] R. Chaabouni, "Break WEP Faster with Statistical Analysis," *Break WEP Faster with Statistical Analysis*, 2006.

[16] I. P. Mavridis, A.-I. E. Androulakis, A. B. Halkias, and P. Mylonas, "Real-life paradigms of wireless network security attacks," in *Proceedings of the 15th Panhellenic Conference on Informatics, PCI 2011*, pp. 112–116, Greece, October 2011.

[17] G. Baldini, G. Steri, and R. Giuliani, "Identification of Wireless Devices From Their Physical Layer Radio-Frequency Fingerprints," 2018.

[18] A. Mahmood and M. A. Jensen, "Data-dependent transmitter fingerprints for radio authentication," in *Proceedings of the 2014 IEEE Radio and Wireless Symposium, RWS 2014*, pp. 265–267, USA, January 2014.

[19] T. J. Bihl, K. W. Bauer, and M. A. Temple, "Feature selection for RF fingerprinting with multiple discriminant analysis and using ZigBee device emissions," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1862–1874, 2016.

[20] J. Lopez, N. C. Liefer, C. R. Busho, and M. A. Temple, "Enhancing Critical Infrastructure and Key Resources (CIKR) Level-0 Physical Process Security Using Field Device Distinct Native Attribute Features," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1215–1229, 2018.

[21] B. Danev, D. Zanetti, and S. Capkun, "On physical-layer identification of wireless devices," *ACM Computing Surveys*, vol. 45, no. 1, article 6, 2012.

[22] O. Gungor and C. E. Koksal, "On the basic limits of RF-fingerprint-based authentication," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 62, no. 8, pp. 4523–4543, 2016.

[23] W. Wang, Z. Sun, S. Piao, B. Zhu, and K. Ren, "Wireless physical-layer identification: modeling and validation," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2091–2106, 2016.

[24] J. Hall, M. Barbeau, and E. Kranakis, "Enhancing intrusion detection in wireless networks using radio frequency fingerprinting," in *Proceedings of the Third IASTED International Conference on Communications, Internet, and Information Technology*, pp. 201–206, USA, November 2004.

[25] O. Ureten and N. Serinken, "Wireless security through RF fingerprinting," *Canadian Journal of Electrical and Computer Engineering*, vol. 32, no. 1, pp. 27–33, 2007.

[26] K. B. Rasmussen and S. Capkun, "Implications of radio fingerprinting on the security of sensor networks," in *Proceedings of the Security and Privacy in Communications Networks and the Workshops (SecureComm)*, 2007.

[27] B. Danev, T. S. Heydt-Benjamin, and S. Capkun, "Physical-layer identification of RFID devices," in *Usenix Security Symposium*, pp. 199–214, Montreal, Canada, 2009.

[28] Y. Zheng, B. Jeon, L. Sun, J. Zhang, and H. Zhang, "Student's t-Hidden Markov Model for Unsupervised Learning Using Localized Feature Selection," *IEEE Transactions on Circuits & Systems for Video Technology*, no. 99, p. 1, 2017.

[29] J. Han, C. Qian, P. Yang et al., "GenePrint: generic and accurate physical-layer identification for UHF RFID tags," *IEEE/ACM Transactions on Networking*, no. 99, p. 1, 2015.

[30] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the internet impasse through virtualization," *The Computer Journal*, vol. 38, no. 4, pp. 34–41, 2005.

[31] D. R. Reising, M. A. Temple, and J. A. Jackson, "Authorized and Rogue Device Discrimination Using Dimensionally Reduced RF-DNA Fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1180–1192, 2015.

[32] Y. Jia, J. H. Ma, and L. Gan, "Radiometric Identification Based on Low-Rank Representation and Minimum Prediction Error Regularization," *IEEE Communications Letters*, vol. 21, no. 8, pp. 1847–1850, 2017.

[33] H. J. Patel, M. A. Temple, and R. O. Baldwin, "Improving ZigBee device network authentication using ensemble decision tree classifiers with radio frequency distinct native attribute fingerprinting," *IEEE Transactions on Reliability*, vol. 64, no. 1, pp. 221–233, 2015.

[34] R. Klein, M. A. Temple, M. J. Mendenhall, and D. R. Reising, "Sensitivity Analysis of Burst Detection and RF Fingerprinting Classification Performance," in *Proceedings of the ICC 2009 - 2009 IEEE International Conference on Communications*, pp. 1–5, Dresden, Germany, June 2009.

[35] W. C. Suski II, M. A. Temple, M. J. Mendenhall, and R. F. Mills, "Using spectral fingerprints to improve wireless network security," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 2185–2189, December 2008.

[36] O. Ureten and N. Serinken, "Bayesian detection of Wi-Fi transmitter RF fingerprints," *IEEE Electronics Letters*, vol. 41, no. 6, pp. 373-374, 2005.

[37] D. R. Reising, M. A. Temple, and M. E. Oxley, "Gabor-based RF-DNA fingerprinting for classifying 802.16e WiMAX mobile subscribers," in *Proceedings of the 2012 International Conference on Computing, Networking and Communications, ICNC'12*, pp. 7–13, USA, February 2012.

[38] D. Sun and X. P. University, "Application of Wavelet Transform on Speech Signal Noise Reduction," *Information & Communications*, 2017.

[39] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 123–151, 2005.

[40] R. W. Klein, M. A. Temple, and M. J. Mendenhall, "Application of wavelet-based RF fingerprinting to enhance wireless network security," *Journal of Communications and Networks*, vol. 11, no. 6, pp. 544–555, 2009.

[41] H. L. Van Trees and K. L. Bell, *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*, IEEE, 2007.

[42] Y. Lin, X. Zhu, Z. Zheng, Z. Dou, and R. Zhou, "The individual identification method of wireless device based on dimensionality reduction and machine learning," *The Journal of Supercomputing*, pp. 1–18, 2017.

[43] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Tensors via Convex Optimization," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5249–5257, Las Vegas, NV, USA, June 2016.

[44] I. Omerhodzic, S. Avdakovic, A. Nuhanovic, and K. Dizdarevic, "Energy distribution of EEG signals: EEG signal wavelet-neural network classifier," *World Academy of Science, Engineering and Technology*, vol. 37, pp. 1240–1245, 2010.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 2492956, 10 pages
https://doi.org/10.1155/2018/2492956

Research Article

An Evolutionary Computation Based Feature Selection Method for Intrusion Detection

Yu Xue,[1,2] **Weiwei Jia**,[1] **Xuejian Zhao**,[3] **and Wei Pang**[4]

[1]*School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China*
[2]*Jiangsu Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications, China*
[3]*School of Modern Posts, Nanjing University of Posts and Telecommunications, Nanjing 210044, China*
[4]*School of Natural and Computing Sciences, University of Aberdeen, AB24 3UE, UK*

Correspondence should be addressed to Yu Xue; xueyu@nuist.edu.cn

Received 27 June 2018; Accepted 23 September 2018; Published 9 October 2018

Guest Editor: Weizhi Meng

As the important elements of the Internet of Things system, wireless sensor network (WSN) has gradually become popular in many application fields. However, due to the openness of WSN, attackers can easily eavesdrop, intercept, and rebroadcast data packets. WSN has also faced many other security issues. Intrusion detection system (IDS) plays a pivotal part in data security protection of WSN. It can identify malicious activities that attempt to violate network security goals. Therefore, the development of effective intrusion detection technologies is very important. However, many dimensions of the datasets of IDS are irrelevant or redundant. This causes low detection speed and poor performance. Feature selection is thus introduced to reduce dimensions in IDS. At the same time, many evolutionary computing (EC) techniques were employed in feature selection. However, these techniques usually have just one Candidate Solution Generation Strategy (CSGS) and often fall into local optima when dealing with feature selection problems. The self-adaptive differential evolution (SaDE) algorithm is adopted in our paper to deal with feature selection problems for IDS. The adaptive mechanism and four effective CSGSs are used in SaDE. Through this method, an appropriate CSGS can be selected adaptively to generate new individuals during evolutionary process. Besides, we have also improved the control parameters of the SaDE. The K-Nearest Neighbour (KNN) is used for performance assessment for feature selection. KDDCUP99 dataset is employed in the experiments, and experimental results demonstrate that SaDE is more promising than the algorithms it compares.

1. Introduction

Wireless sensor networks (WSNs) are typical distributed sensor networks, which can realize data acquisition, processing, and transmission. It can monitor, perceive, and collect data from various sources or monitoring objects in the areas covered by the network and transmit them to users after data processing. As an emerging infrastructure for the application of Internet of Things, WSNs are widely used, for example, environmental monitoring, defense, urban management, medical applications, and other aspects [1, 2]. At present, most of the deployed WSNs collect scalar data like humidity and location. In practical applications of smart home, traffic monitoring, and medical monitoring, the wireless multimedia sensor network can process multimedia data, such as videos, audios, and images [3]. Therefore, WSNs are increasingly associated with people's usual economic and social activities. However, due to the weakness of wireless links, the lack of physical protection of nodes, and the dynamic nature of topology, WSNs are facing a variety of data security risks. The openness of WSNs allows attackers to easily eavesdrop, intercept, and tamper with packets. The most common attacks are denial of service attacks, Hello flooding attacks, replay routing attacks, and so on [4, 5]. These attacks may leak data and cause security problems in WSNs. Users are less likely to use large-scale WSNs that lack security protection and have privacy issues. Therefore, in order to promote the wider use and development of WSN, it is very important to address the security issues of WSN.

Intrusion detection system (IDS) plays a pivotal part in data security protection of WSNs [6], which can identify malicious activities that attempt to violate network security goals. IDSs identify malicious activities by monitoring the system in real time. Once they find abnormal situation, a warning will be issued. Dorothy first proposed an abstract model of the IDS in 1987 [7], which is a real-time IDS framework. At present, various IDSs have been deployed to detect anomalies [8]. In addition, neural networks [9], particle swarm intelligence [10], differential evolution algorithm [11], and other technologies [12] have been used in IDS to improve its performance. Among them, the metaheuristic algorithms have been well used to solve the IDS problems [13–15]. At the same time, there are many studies on IDS applied to WSNs [16, 17]. In [17], the meaning and function of external signals used in WSN are defined. In addition, it realizes distributed deployment and real-time IDS by improving DCA-RT dendritic cell algorithm. A distributed network IDS which is applicable to wireless networks is put forward in [18]. It is based on the principle of classification rule induction and swarm intelligence theory. Without the need to exchange sensitive data, this system can enable effectual model training for the IDS. Nowadays, large-scale distributed intrusion detection and intrusion detection data fusion technology are the main development directions for IDS [19]. However, IDSs need to detect huge amounts of data. Actually, most of the features in the datasets are redundant or irrelevant, which can result in an increase in training time and low detection speed. In the study of network information security, it is always prominent to find out the methods that can quickly and effectively get information of destroying security from intrusion detection data. Feature selection is significant for intrusion detection, since it can reduce the time complexity of the classifier and improve the efficiency by using optimized feature subsets.

In the current big data environment, mining the knowledge contained in big data is very important for guiding practical life and applications. Feature selection has therefore become more significance [20–25]. Feature selection is applied to intrusion detection in our paper. At present, feature selection has become a hot topic in machine learning [26]. As a way of achieving dimension reduction, it selects the best feature combination rather than the whole dataset. According to the independent relationship between feature selection and classifiers, feature selection is usually separated into two groups: the filter and the wrapper [27].

A filter method is independent of any classifiers. It only considers the relevance between features and class labels. It ranks the features through the experience of statistics, information theory, and many other disciplines. Student's t-test [28] and Fisher Discriminant Ratio [29] are typical hypothesis test means in statistical techniques. Meanwhile, features can be sorted from different perspectives such as entropy or information gain [30]. This is a methodological perspective based on information theory. Amiri *et al.* [31] has put forward an improved feature selection algorithm on account of mutual information. It can effectively identify the characteristics of the attacks by calculating the mutual information. Evaluating the quality of a subset of features can also apply the correlation [32]. If the correlation between a feature subset and classification is high, but the correlation between a feature and the others in the subset is low, then this feature subset is good. Besides, distance measurement is also used for feature selection [33]. The commonly used distance measures include Euclidean distance, standardized Euclidean distance, and martensitic distance.

In Wrappers, the subsequent learning algorithm is embedded into feature selection process and the performance of algorithms is determined by testing prediction performance of the feature subset. Besides, the impact of a single feature on the final result is taken into account. The typical means include sequence forward selection (SFS) [34] and sequence backward selection (SBS) [35]. The disadvantage of SBS is that it can only add features and cannot remove features. SFS is the opposite of SBS. Both SFS and SBS use greedy strategies, which can easily fall into local optimal values. The L to R selection algorithm (LRS) [36] has been offered to deal with such problem. There are two forms of this algorithm. On the one hand, it is a null set at the beginning. The algorithm appends L features each round first and then removes R features from it. In this way, the evaluation function value is made to the best. For another, the algorithm begins with the complete set, removing R features first round and then adding L features to make the evaluation function value optimal. The sequence floating selection is developed by LRS algorithm. Compared with LRS, the distinction of the two lies in that L and R of the sequence floating selection are not fixed but will change. It includes Sequence Floating Forward Selection (SFFS) and Sequence Floating Forward Selection (SFBS) [37]. SFFS starts from an empty set. It selects a subset x of unselected features in each round so that the evaluation function is optimal after adding subset x and then selects subset z from the selected features to optimize the evaluation function after removing subset z. SFBS is similar to SFFS, but the difference is that SFBS begins with the complete set. It removes features first in each round and then adds features.

Traditional filter and wrapper methods individually evaluate and select subsets. However, some features are not independent, but they play a great performance when they work with each other. Thus, the traditional method is not very good in this respect. Evolutionary computing (EC) methods have already been used for feature selection and classification in virtue of its overall optimization capabilities [38, 39], for instance, Particle Swarm Optimization (PSO) [40–43], Genetic Algorithm [44–46], ant colony optimization [47, 48], and some of the algorithms mentioned in [49], whereas the solution space of the feature selection problem increases exponentially with the rise of the dimension of the dataset. Therefore, more and more features lead to huge solution space. Also, a large number of uncorrelated or redundant features produce many local optima in a large solution space. Therefore, most EC methods still have local optimal stagnation problems [50]. Another reason for this problem may be that many of these methods lack the ability to explore and utilize search spaces in an appropriate manner [51]. Therefore, the applicable search methods should be automatically used based on the specific feature selection

problems. However, many existing evolutionary algorithms have only one search strategy and cannot effectively deal with the complex situations that arise in real-world problems. In other words, in many existing feature selection algorithms, only one Candidate Solution Generation Strategy (CSGS) is used to generate a new solution. In addition, IDSs need to address large-scale issues. Recently, EC methods using adaptive mechanisms have been exploited to deal with continuous optimization issues, and the performance is promising [52–56]. The adaptive mechanism is rarely used for feature selection in IDS. Therefore, a self-adaptive differential evolution (SaDE) [57] method with several CSGSs are introduced to cope with the issue of feature selection for IDSs. In SaDE, an adaptive mechanism is introduced to DE algorithm and improve its control parameter. DE is an effective method, and mechanism can increase the diversity of solutions. Combining these two can search the optimal strategy for current problem dynamically during the search process.

The remainder of this paper is organized as follows. In Section 2, the SaDE algorithm is presented. Section 3 introduces the experiment and gives results of the discussions. Conclusions and the future research work are provided in Section 4.

2. Self-Adaptive Differential Evolution

2.1. Initialization and DE Algorithm. The DE method is on account of evolutionary theory. As a heuristic random search method in view of group difference, the basic idea stems from the competitive strategy of the survival of the fittest in Darwin's theory of biological evolution. According to the differential vector between the parent's individuals, DE performs mutation, crossover, and selection operations. The algorithm contains the following aspects.

2.1.1. Initialization and Updating Mechanisms in DE. Unlike traditional initialization methods, this paper uses a mixed initialization strategy. Most particles are initialized with a few features, and the remaining particles are initialized with a large subset of features. It has been demonstrated in [50] that this initialization strategy can greatly improve the selection performance. *pbest* represents the best value of single particles. *gbest* represents particles' global best value. *Pbest* and *gbest* are updated according to their classification performance.

2.1.2. Mutation. The DE algorithm implements the mutation operation by the difference method. Random selection of two diverse individuals in a group and scaling vector differences are the common difference strategy. Afterwards, the vector is synthesized with the individual to be mutated. Formula (1) is used to generate a new individual.

$$V_i(g+1) = X_{r_1}(g) + F\left(X_{r_2}(g) - X_{r_3}(g)\right) \quad (1)$$

where g represents the g_{th} iteration of evolution. i, r_1, r_2, and r_3 are random integers of $[1, 2, \ldots, ps]$. ps represents particles' number. Moreover, $i \neq r_1 \neq r_2 \neq r_3$. F is called scaling factor, which is used to scale difference vector. It is a constant. $x_i(g)$ represents the i_{th} individual in the g_{th} generation population. $V_i(g+1)$ is the newly generated particle in the next generation. Through mutation, a new intermediate population $\{V_i(g+1),\ i = 1, 2, \ldots, ps\}$ is finally generated.

2.1.3. Crossover. Crossover aims to select individuals randomly, because DE is also a random algorithm. The crossover operations are performed between $X_i(g)$ and $V_i(g+1)$. The trial vector is generated according to formula (2).

$$U_{i,j}(g+1) = \begin{cases} V_{i,j}(g+1), & if\ rand \leq CR\ or\ j = j_{rand} \\ X_{i,j}(g), & otherwise \end{cases} \quad (2)$$

where CR is called crossover probability. It is a random value between 0 and 1. j_{rand} is a random integer of $[1, 2, \ldots, D]$. D represents the dimensions. A new individual $U_i(g+1)$ is randomly generated from a probability distribution. The reason for doing such an operation is to ensure that at least one component of $U_i(g+1)$ is contributed by the corresponding component in $V_i(g+1)$. Other variables have the same explanation as mentioned above.

2.1.4. Selection. DE adopts a greedy choice strategy. On the basis of fitness value, better individuals are selected as new ones of new population. Formula (3) below is used for selecting. Among them, f is the fitness function. Other variables have the same explanation as mentioned above.

$$X_i(g+1) = \begin{cases} U_i(g+1), & if\ f(U_i(g+1)) \leq f(X_i(g)) \\ X_i(g), & otherwise \end{cases} \quad (3)$$

2.2. Representation of Solutions. In this paper, feature selection is transformed into combinatorial optimization problems of "0" and "1", with "0" meaning not selecting the corresponding feature and "1" otherwise. The binary string is used to represent the solution. The string dimension set to D dimensions is the same as the total amount of the feature. Threshold θ is used to limit the vector range for each dimension to between 0 and 1. That is to say, if the value of the d_{th} dimension of the position is greater than θ, the corresponding value in the binary vector will be set to 1, which means choosing the d_{th} feature. Otherwise, it will be set to 0.

2.3. The Self-Adaptive Mechanism. The main goal of this mechanism is to generate the probabilities of CSGSs on account of their performance and to choose the suitable CSGS for every particle on account of these probabilities. CSGSs which have been successfully used in recent generations will be in higher probability to be selected in future generations. When a CSGS does not work well, it should be replaced by another CSGS that has good performance. We will give a brief introduction to the mechanism.

The 4 CSGSs used in our paper are assigned the initial probability. During the evolution process, the probability changes. Let p_q represents the selection probability of the q_{th} strategy, where q = 1, 2, 3, . . . , Q. Q is the number of CSGSs used, and in this research, Q =4. Then, the initial probability of each CSGS is set to be 1/4. The sum of these probabilities is 1, and p_q is recalculated according to the performance of CSGS in producing new solutions. In this research, the roulette wheel technique is applied to choose CSGSs because it can randomly select targets with high probabilities in each cycle [58]. Subsequently, the selected CSGS is applied to the corresponding particle for generating the candidate solution. The candidate solution is then evaluated and the update mechanism described in the second part is used to determine whether *pbest* and *gbest* should be updated. The $nsFlag_{i,q}$ and $nfFlag_{i,q}$ ($(i = 1, 2, \ldots, ps,\ q = 1, 2, \ldots, Q)$, where ps is the number of particles and Q has mentioned above) in the binary matrices $nsFlag_{ps\times Q}$ and $nfFlag_{ps\times Q}$ are used to record the information that reflects the relationship between the generated solution and the corresponding *pbest*. In other words, supposing that newly generated solution is preferable than the old one, afterward, $nsFlag_{i,q}$ = 1. Otherwise, $nfFlag_{i,q}$= 1. When a generation starts, $nsFlag_{ps\times Q}$ and $nfFlag_{ps\times Q}$ are initialized to $ps \times Q$-dimensional zero matrices.

In the evolution process, the i_{th} particle selects the q_{th} strategy to produce new solutions. Supposing that newly generated solution is preferable than the old one, afterwards, the corresponding position of the q_{th} strategy used by the i_{th} particle in matrix $nsFlag_{ps\times Q}$ is set to 1, which is $nsFlag_{i,q}$ = 1. Otherwise, the corresponding position in $nfFlag_{ps\times Q}$ is set to 1, that is, $nfFlag_{i,q}$ = 1. After repeated evolution for the LP generations, *nsFlag* and *nfFlag* are reinitialized to record the information in the following generation. When the evolution of the present generation is completed, all rows in *nsFlag* and *nfFlag* will be merged and the results will be recorded in $S_{k,q}$ ($k = 1, 2, \ldots, LP, q = 1, 2, \ldots, Q$, where LP is the number of generations, k is the k_{th} generation for each LP generations) and $F_{k,q}$, respectively. In other words, $S_{k,q}$ records the number of the new solutions that are produced by the q_{th} CSGS and succeed in entering into the following generation. Correspondingly, $F_{k,q}$ records the amount of the new solutions produced by the q_{th} CSGS that fail to enter into the next generation. After the evolutionary process is repeated for LP generations, all the elements of $S_{k,q}$ and $F_{k,q}$ make up the matrix $S_{LP\times Q}$ and $F_{LP\times Q}$, respectively. The strategy selection probabilities of the CSGSs are recalculated based on the statistical data stored in matrices S and F. Both $S_{LP\times Q}$ and $F_{LP\times Q}$ are initialized to be a $LP \times Q$-dimensional zero matrix at the first generation of each LP generations. After repeating the evolution of the LP generations, we can obtain the success and failure information of the CSGSs. The following steps are used to recalculate the probability of the q_{th} $(q = 1, 2, \ldots Q)$ strategy.

$$S_q^1 = \sum_{k=1}^{LP} S_{k,q} \tag{4}$$

$$S_q^2 = \begin{cases} \varepsilon, & if\ S_q^1 = 0 \\ S_q^1, & otherwise \end{cases} \tag{5}$$

$$S_q^3 = \frac{S_q^1}{\left(S_q^2 + \sum_{k=1}^{LP} F_{k,q}\right)} \tag{6}$$

$$P_q = \frac{S_q^3}{\sum_{q=1}^{Q} S_q^3} \tag{7}$$

where (4) is used to compute the sum of each column of matrix $S_{LP\times Q}$. S_q^3 is the proportion of the new solutions produced according to the q_{th} strategy and replaced their corresponding *pbests* successfully within LP generations. Meanwhile, the matrices $S_{LP\times Q}$ and $F_{LP\times Q}$ are initialized. In (5), the small value $\varepsilon = 0.0001$ is applied to avert division by 0. In other words, if $S_q^1 = 0$, then S_q^2 is equal to ε. Otherwise, S_q^2 is equal to S_q^1. The probabilities are normalized by (7) to ensure that they always sum to 1. The above steps are used to produce new probabilities for the CSGSs based on their performance during LP generations evolution. The CSGSs are chosen according to the new probabilities. Apparently, if the probability value is greater, the probability of selecting the corresponding CSGS is greater.

2.4. Candidate Solution Generation Strategy (CSGS). In our research, we use four powerful CSGSs which are inspired by mutation strategies of DE to generate new solutions [59]. They are used in the mutation operation. For simplicity, the symbol $DE/a/b$ is used to represent different mutation operators. a represents the basic vector, and b represents the number of difference vectors used. They are described as follows:

(1) The first strategy is named DE/rand/1. This has been described in formula (1).

(2) The second strategy is the generation of the next generation by the current individual, the current optimal individual, and four different random individuals. It is called DE/current-to-best/2, which is described in (8) as follows:

$$\begin{aligned} V_i(g+1) = X_i(g) &+ F\left(X_{best}(g) - X_i(g)\right) \\ &+ F\left(X_{r_1}(g) - X_{r_2}(g)\right) \\ &+ F\left(X_{r_3}(g) - X_{r_4}(g)\right) \end{aligned} \tag{8}$$

where $X_{best}(g)$ is the best individual in the g_{th} generation population. $X_i(g)$ represents the i_{th} individual in the g_{th} generation population. The meaning of other variables has been introduced previously.

(3) The third strategy is the generation of the next generation by a random individual and four different random individuals. It is called DE/rand/2, which is described as formula (9) as follows. Other variables have been mentioned before.

$$\begin{aligned} V_i(g+1) = X_{r_1}(g) &+ F\left(X_{r_2}(g) - X_{r_3}(g)\right) \\ &+ F\left(X_{r_4}(g) - X_{r_5}(g)\right) \end{aligned} \tag{9}$$

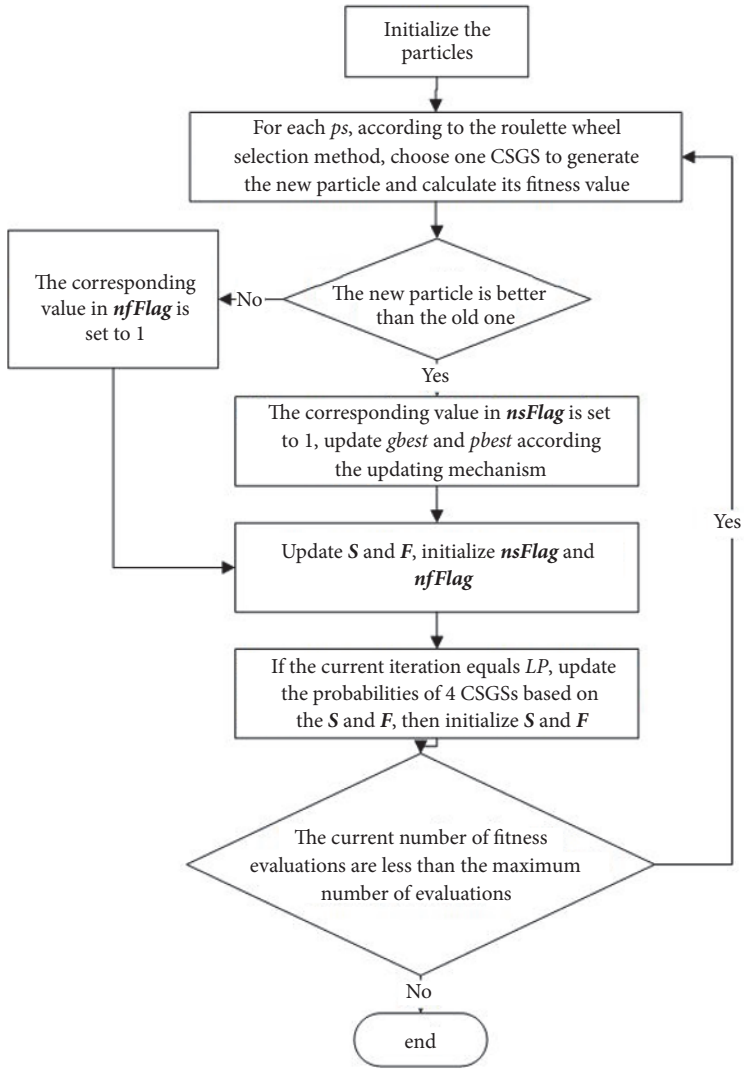


Figure 1: The flow chart of SaDE algorithm.

(4) The fourth strategy is called DE/current-to-rand/1. It includes mutation and crossover, which is described as formula (10) as follows:

$$U_i(\mathrm{g}+1) = X_i(g) + k\left(X_{r_1}(g) - X_i(g)\right) + F\left(X_{r_2}(g) - X_{r_3}(g)\right) \tag{10}$$

where k is the combination coefficient and it is a random number between 0 and 1. Other variables have been mentioned previously.

The procedure of the SaDE algorithm is shown Figure 1. The algorithm finally outputs *gbest*. The variables in the figure have been introduced in the second section.

3. Experiments and Results

The performance of the proposed method is assessed by carrying out the experiments. The sections below briefly describe the dataset, data preprocessing, parameter settings, and results of the experiments.

3.1. Datasets and Data Preprocessing. The dataset employed in this research is the KDDCUP99 dataset [60]. It is a well-known test dataset in the domain of network IDS. Each instance of this dataset has 41 feature attributes and one label. There are 13 types of content characteristics of Transmission Control Protocol (TCP) connection. There are nine types of time-based network traffic statistics and ten host-based traffic features, including four major categories and twenty-two minor categories of attacks: DoS, Probing, R2L, and U2R [61]. A number of 5 million records are included in the KDDCUP99 dataset. A 10% training subset and the test subset are offered as well. In order to save experimental time, the dataset is randomly reduced. 70% of it is used as a training set and 30% is used as a test set, in which we randomly selected 3,458 training samples and 1,482 test samples together to constitute the experimental data.

Table 1: Percentage of classification accuracies on training sets in kddcup99.

	SFFS	SBFS	Standard PSO	SaDE
Max	99.62	99.65	99.71	99.68
Min	99.28	99.42	99.68	99.68
Mean	99.48	99.58	99.68	**99.68**
Std	0.12	0.07	0.01	0.00
T-Sig	+	+	-	

Table 2: Solution sizes on training sets in kddcup99.

	SFFS	SBFS	Standard PSO	SaDE
Max	8.00	39.00	28.00	23.00
Min	2.00	37.00	14.00	11.00
Mean	**4.35**	38.42	19.81	17.69
Std	1.55	0.70	3.09	3.48
T-Sig	+	+	+	

Table 3: Percentage of classification accuracies on test sets in kddcup99.

	SFFS	SBFS	Standard PSO	SaDE
Max	98.99	98.92	98.99	98.99
Min	97.50	9818	98.45	98.38
Mean	98.60	98.64	98.68	**98.71**
Std	0.35	0.17	0.15	0.18
T-Sig	-	-	-	

Datasets are numerically processed before they are trained, as the classifier can only recognize quantitative. For the sake of testing the function of the algorithm better after improving the parameters, we also generate 4 new datasets from the KDDCUP99 dataset. Among them, we randomly selected 4 times from the original data and randomly selected 1% of the original dataset each time. We denote these datasets as DataNum1, DataNum2, DataNum3, and DataNum4, respectively. The K-Nearest Neighbour (KNN) method is applied as a classification method to evaluate subsets of features generated. In KNN, 3-fold cross validation is employed to measure the classification accuracy.

3.2. Parameter Settings. We choose SFFS, SBFS, standard PSO, and SaDE for comparison. According to past experience, each algorithm runs 26 times on the KDDCUP99 dataset. With regard to 4 CSGSs used in our paper, initial CR=0.5, F is selected from normal distribution with μ=0.5 and σ=0.3. Furthermore, ps =100. The generations of evolution named LP were empirically set to 10.

3.3. Results and Analysis. The results according to solution size and classification accuracy on the training set and the test set will be shown in the part. The solution size is the number of features chosen by the feature selection that are most beneficial to ameliorate the classification accuracy. The best result will be bold. We compare the performance of SaDE and other algorithms on DataNum1 and compare the performance of SaDE after improving the control parameter on DataNum1 to DataNum4.

Table 1 shows the classification accuracy of SaDE and other algorithms on training sets. As indicated in Table 1, the results of solution sizes are obtained by the algorithms mentioned above, including Max, Min, mean values (Mean), and standard deviations (Std). Min represents the minimum value of classification accuracy. Max means the opposite meaning of it. Mean expresses the average of the classification accuracy over 26 runs and Std shows the standard deviation in the same situation. The t-test is a statistical test used to check hypothesis with the average value of the given trust level. In our experiments DF (degree freedom) =50, and the t is equal to 2,009 (when the trust level is equal to 0.95). Therefore the results obtained are statistically important when t is less than -2,009 or higher than +2,009. We only check two cases: IMPORTANT (+) or NOT IMPORTANT (-). 'T-Sig' means the algorithm introduced in this paper is significantly distinct from other algorithms. Table 2 provides the solution sizes of the mentioned methods on training sets. Table 3 presents the classification accuracies on the test sets. According to the comparison between the SaDE and other methods, we can see that SaDE has the highest classification accuracy on test sets and training sets. Simultaneously, it has the second fewest discriminative features. Although the SFFS method has the fewest discriminative features, its amount of the selected feature is too few and its classification accuracy is poor. Moreover, the standard deviation of the classification results

Table 4: Classification accuracies of thresholds of SaDE in test sets in 4 datasets.

Dataset	0.6		0.7		0.8		0.5	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
DataNum1	98.71	0.18	98.74	0.19	98.60	0.21	**99.23**	0.13
T-Sig	+		+		+			
DataNum2	98.97	0.12	99.02	0.16	98.95	0.25	**99.05**	0.10
T-Sig	+		-		-			
DataNum3	99.09	0.19	99.15	0.22	99.17	0.14	**99.17**	0.14
T-Sig	-		-		-			
DataNum4	98.46	0.14	98.44	0.13	**98.57**	0.22	98.46	0.15
T-Sig	-		-		+			

Table 5: Classification accuracies of thresholds of SaDE in training sets in 4 datasets.

Dataset	0.6		0.7		0.8		0.5	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
DataNum1	99.68	0.18	99.68	0.19	99.66	0.21	**99.68**	0.13
T-Sig	-		-		-			
DataNum2	99.70	0.01	99.71	0.02	99.67	0.01	**99.71**	0.01
T-Sig	+		-		+			
DataNum3	99.68	0.01	99.68	0.01	99.66	0.02	**99.68**	0.01
T-Sig	-		-		+			
DataNum4	99.82	0.01	99.82	0.01	99.81	0.02	**99.82**	0.01
T-Sig	-		-		+			

of SaDE is good no matter it is in the test sets or the training sets. This result indicates that SaDE has good robustness. By comparison, although SFFS has the least characteristics, its robustness is not as good as that of SaDE. That is because adaptive mechanism and 4 CSGSs can increase the diversity of solutions. It can search the optimal strategy for current problem dynamically during the search process. Considering the statistically significant difference on the training sets, from the perspective of classification accuracy, the results obtained are statistically important between SaDE and SFFS and SBFS, and not important between SaDE and standard PSO. From the perspective of solution sizes, the results obtained are statistically important between SaDE and the other three methods. According to the difference on the test sets, the results obtained are not important between SaDE and other methods.

In addition, from these Tables 1–3, we can see that other algorithms are inferior to SaDE according to classification detection rate and the number of feature reduction. In summary, we can conclude that SaDE is an effective technique in IDS. It can also select the most useful and representative subset of intrusion detection features to reduce computational cost for IDS. From this, we can see that the adaptive mechanism and multiple CSGSs can improve the performance of the DE algorithm on IDS.

We improve the performance of SaDE by optimizing its parameters. We tested the different values of SaDE parameters in the above four datasets to test their effectiveness on the detection rate. Tables 4 and 5 show the effect of different thresholds on the classification accuracy of test sets and training sets in DataNum1 to DataNum4, respectively. The unit is the percentage. The threshold is a significant part in the initialization phase. It determines whether the features are selected. In experiments comparing SaDE with other algorithms, we set the threshold to 0.6 based on experimental experience. To improve the algorithm's performance, we set 4 different values of the threshold within the range $[0, 1]$. The reasons for this setting are briefly explained in the Section 2. From the table, we can see that, whether in the test sets or the training sets, when the threshold is set to 0.5, the classification accuracy is the best. Besides, in most cases, the robustness is also the best. Considering the statistically significant difference on the test sets and training sets, when the threshold is set to different values, the results obtained are statistically important between 0.5 and 0.8(0.6), but not significant between 0.5 and 0.7. Therefore, by optimizing the parameters, it helps improve the classification accuracy.

4. Conclusions and Future Work

Nowadays, information technology is entering the era of Internet of Things (IoTs) from the Internet age. With the application of IoTs, WSN is facing more and more data security problems. Security is a key issue in WSN design, because it seriously affects the application prospect of WSN. Intrusion detection is an important way to ensure network security. The improvement of its technology is also an aspect of guaranteeing the data security of WSN. The feature selection problem has been analyzed and the SaDE algorithm has been introduced to solve this kind of problem of IDS.

The KDDCUP99 intrusion dataset was applied to assess the performance of the introduced algorithm. Our scheme applies an adaptive mechanism in the DE algorithm to find the CSGS that is most suitable for generating new solutions. At the same time, we have improved the control parameters of SaDE. According to the results of experiments, it can be seen that the improved SaDE can effectively solve the IDS problem. On the one hand, by comparing the SaDE algorithm with other methods, we can see that the SaDE algorithm can reduce about 57% of the features in the problem. In addition, the SaDE method is superior to other algorithms in terms of classification accuracy of training sets and test sets. For another, four datasets generated from KDDCUP99 were used to test the control parameters. When the threshold is set to 0.5, the classification accuracy of SaDE is better than other values, and the performance of SaDE has been improved.

In the problems of intrusion detection, multiobjective feature selection is also a field which has been researched for many years, and SaDE algorithm has not been used in this field. Therefore, we can also resolve the multiobjective feature selection problem in intrusion detection by combining the classifier and SaDE algorithm in the future. Moreover, we can also make improvements in the initialization section.

Data Availability

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

Disclosure

This work is based on the conference paper that was presented in "The 4th International Conference on Cloud Computing and Security (ICCCS2018)".

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61403206, 61771258, and 61876089), the Natural Science Foundation of Jiangsu Province (BK20141005 and BK20160910), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (14KJB520025), the Priority Academic Program Development of Jiangsu Higher Education Institutions, the Open Research Fund of Jiangsu Engineering Research Center of Communication and Network Technology, NJUPT (JSGCZX17001), and the Natural Science Foundation of Jiangsu Province of China under Grant BK20140883.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, and G. Wang, "Security and privacy in the medical internet of things: a review," *Security and Communication Networks*, vol. 2018, Article ID 5978636, 9 pages, 2018.

[3] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[4] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[5] P. Li, X. Yu, H. Xu, J. Qian, L. Dong, and H. Nie, "Research on secure localization model based on trust valuation in wireless sensor networks," *Security and Communication Networks*, vol. 2017, Article ID 6102780, 12 pages, 2017.

[6] J. Granjal and A. Pedroso, "An intrusion detection and prevention framework for internet-integrated CoAP WSN," *Security and Communication Networks*, vol. 2018, Article ID 1753897, 14 pages, 2018.

[7] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.

[8] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255–277, 2017.

[9] H. I. Ahmed, N. A. Elfeshawy, S. F. Elzoghdy, H. S. Elsayed, and O. S. Faragallah, "A neural network-based learning algorithm for intrusion detection systems," *Wireless Personal Communications*, vol. 97, no. 2, pp. 3097–3112, 2017.

[10] I. Ahmad and F. E Amin, "Towards feature subset selection in intrusion detection," in *Proceedings of the 7th IEEE Joint International Information Technology and Artificial Intelligence Conference (ITAIC '14)*, pp. 68–73, December 2014.

[11] A. A. Aburomman and M. B. Ibne Reaz, "A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems," *Information Sciences*, vol. 414, pp. 225–246, 2017.

[12] S. Rastegari, P. Hingston, and C.-P. Lam, "Evolving statistical rulesets for network intrusion detection," *Applied Soft Computing*, vol. 33, pp. 348–359, 2015.

[13] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: a review," *Applied Soft Computing*, vol. 10, no. 1, pp. 1–35, 2010.

[14] S. Revathi and A. Malathi, "Optimization of KDD Cup 99 dataset for intrusion detection using hybrid swarm intelligence with random forest classifier," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 7, pp. 1382–1387, 2013.

[15] Y. Y. Chung and N. Wahid, "A hybrid network intrusion detection system using simplified swarm optimization (SSO)," *Applied Soft Computing*, vol. 12, no. 9, pp. 3014–3022, 2012.

[16] I. Onat and A. Miri, "An intrusion detection system for wireless sensor networks," in *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '05)*, pp. 253–259, Québec, Canada, August 2005.

[17] X. Xiao and R. Zhang, "Study of immune-based intrusion detection technology in wireless sensor networks," *Arabian Journal for Science and Engineering*, vol. 42, no. 8, pp. 3159–3174, 2017.

[18] C. Kolias, V. Kolias, and G. Kambourakis, "TermID: a distributed swarm intelligence-based approach for wireless intrusion detection," *International Journal of Information Security*, vol. 16, no. 4, pp. 401–416, 2017.

[19] G. Li, Z. Yan, Y. Fu, and H. Chen, "Data fusion for network intrusion detection: a review," *Security and Communication Networks*, vol. 2018, Article ID 8210614, 16 pages, 2018.

[20] R. Gurusamy and V. Subramaniam, "A machine learning approach for MRI brain tumor classification," *Computers, Materials and Continua*, vol. 53, no. 2, pp. 91–109, 2017.

[21] C. Wu, E. Zapevalova, Y. Chen et al., "Time optimization of multiple knowledge transfers in the big data environment," *Computers, Materials and Continua*, vol. 54, no. 3, pp. 269–285, 2018.

[22] S. A. Yıldızel and A. U. Öztürk, "A study on the estimation of prefabricated glass fiber reinforced concrete panel strength values with an artificial neural network model," *Computers, Materials and Continua*, vol. 552, pp. 42–51, 2016.

[23] Y. Zheng, B. Jeon, L. Sun, J. Zhang, and H. Zhang, "Student's t-hidden markov model for unsupervised learning using localized feature selection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[24] Y. Zheng, L. Sun, S. Wang, J. Zhang, and J. Ning, "Spatially regularized structural support vector machine for robust visual tracking," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2018.

[25] Y. Zhang, X.-F. Song, and D.-W. Gong, "A return-cost-based binary firefly algorithm for feature selection," *Information Sciences*, vol. 418-419, pp. 561–574, 2017.

[26] Z. Yong, G. Dun-wei, and Z. Wan-qiu, "Feature selection of unreliable data using an improved multi-objective PSO algorithm," *Neurocomputing*, vol. 171, pp. 1281–1290, 2016.

[27] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1–4, pp. 131–156, 1997.

[28] W. C. Navidi, *Statistics for Engineers and Scientists*, vol. 2, McGraw-Hill, New York, NY, USA, 2006.

[29] S. Theodoridis and K. Koutroumbas, "Pattern recognition and neural networks," in *Advanced Course on Artificial Intelligence*, vol. 2049 of *Lecture Notes in Computer Science*, pp. 169–195, Springer Berlin Heidelberg, 2001.

[30] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[31] F. Amiri, M. M. R. Yousefi, and C. Lucas, "Mutual information-based feature selection for intrusion detection systems," *Journal of Network & Computer Applications*, vol. 34, no. 4, pp. 1184–1199, 2011.

[32] M. M. Kabir, M. M. Islam, and K. Murase, "A new wrapper feature selection approach using neural network," *Neurocomputing*, vol. 73, no. 16–18, pp. 3273–3283, 2010.

[33] M. Liu and D. Zhang, "Feature selection with effective distance," *Neurocomputing*, vol. 215, pp. 100–109, 2016.

[34] P. Pudil and J. Hovovicova, "Novel methods for subset selection with respect to problem knowledge," *IEEE Intelligent Systems*, vol. 13, no. 2, pp. 66–74, 1998.

[35] T. Marill and D. M. Green, "On the effectiveness of receptors in recognition systems," *IEEE Transactions on Information Theory*, vol. 9, no. 1, pp. 11–17, 1963.

[36] S. D. Strearns, "On selecting features for pattern classifiers," in *Proceedings of the 3rd International Conference on Pattern Recognition (ICPR)*, 1976.

[37] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[38] Y. Xue, T. Ma, B. Zhao, and A. X. Liu, "An evolutionary classification method based on fireworks algorithm," *International Journal of Bio-Inspired Computation*, vol. 11, no. 3, pp. 149–158, 2018.

[39] Y. Zhang, D.-W. Gong, J.-Y. Sun, and B.-Y. Qu, "A decomposition-based archiving approach for multi-objective evolutionary optimization," *Information Sciences*, vol. 430-431, pp. 397–413, 2018.

[40] J. Tian and H. Gu, "Anomaly detection combining one-class SVMs and particle swarm optimization algorithms," *Nonlinear Dynamics*, vol. 61, no. 1-2, pp. 303–310, 2010.

[41] B. Xue, M. Zhang, and W. N. Browne, "Multi-objective particle swarm optimisation (PSO) for feature selection," in *Proceedings of the 14th International Conference on Genetic and Evolutionary Computation (GECCO '12)*, pp. 81–88, July 2012.

[42] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 64–75, 2017.

[43] Y. Zhang, D. Gong, Y. Hu, and W. Zhang, "Feature selection algorithm based on bare bones particle swarm optimization," *Neurocomputing*, vol. 148, pp. 150–157, 2015.

[44] B. M. Aslahi-Shahri, R. Rahmani, M. Chizari et al., "A hybrid method consisting of GA and SVM for intrusion detection system," *Neural Computing and Applications*, vol. 27, no. 6, pp. 1669–1676, 2016.

[45] R. Li, J. Lu, Y. Zhang, and T. Zhao, "Dynamic Adaboost learning with feature selection based on parallel genetic algorithm for image annotation," *Knowledge-Based Systems*, vol. 23, no. 3, pp. 195–201, 2010.

[46] F. Souza, T. Matias, and R. Araújo, "Co-evolutionary genetic Multilayer Perceptron for feature selection and model design," in *Proceedings of the IEEE 16th Conference on Emerging Technologies and Factory Automation (ETFA '11)*, September 2011.

[47] Z. Yan and C. Yuan, "Ant colony optimization for feature selection in face recognition," in *Applied Soft Computing, Biometric Authentication*, vol. 3072 of *Lecture Notes in Computer Science*, pp. 221–226, Springer Berlin Heidelberg, 2004.

[48] N. M. O'Boyle, D. S. Palmer, F. Nigsch, and J. B. Mitchell, "Simultaneous feature selection and parameter optimisation using an artificial ant colony: Case study of melting point prediction," *Chemistry Central Journal*, vol. 2, no. 1, 2008.

[49] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2016.

[50] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms," *Applied Soft Computing*, vol. 18, pp. 261–276, 2014.

[51] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[52] Y. Xue, J. Jiang, B. Zhao, and T. Ma, "A self-adaptive artificial bee colony algorithm based on global best for global optimization," *Soft Computing*, pp. 1–18, 2017.

[53] Y. Xue, Y. Zhuang, T. Ni, S. Ni, and X. Wen, "Self-adaptive learning based discrete differential evolution algorithm for

solving CJWTA problem," *Journal of Systems Engineering and Electronics*, vol. 25, no. 1, pp. 59–68, 2014.

[54] Y. Xue, B. Zhao, T. Ma, and W. Pang, "A self-adaptive fireworks algorithm for classification problems," *IEEE Access*, vol. 6, pp. 44406–44416, 2018.

[55] Y. Wang, B. Li, T. Weise, J. Wang, B. Yuan, and Q. Tian, "Self-adaptive learning based particle swarm optimization," *Information Sciences*, vol. 181, no. 20, pp. 4515–4538, 2011.

[56] C. Li, S. Yang, and T. T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 3, pp. 627–646, 2012.

[57] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[58] D. B. Fogel, "Introduction to simulated evolutionary optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 5, no. 1, pp. 3–14, 1994.

[59] J. Jiang, Y. Xue, T. Ma, and Z. Chen, "Improved artificial bee colony algorithm with differential evolution for the numerical optimisation problems," *International Journal of Computational Sciences and Engineering*, vol. 16, no. 1, pp. 73–84, 2018.

[60] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proceedings of the 2nd IEEE Symposium on Computational Intelligence for Security and Defence Applications (CISDA '09)*, pp. 1–6, IEEE, July 2009.

[61] Y. Zhu, J. Liang, J. Chen, and Z. Ming, "An improved NSGA-III algorithm for feature selection used in intrusion detection," *Knowledge-Based Systems*, vol. 116, pp. 74–85, 2017.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 3780407, 11 pages
https://doi.org/10.1155/2018/3780407

Research Article

Differential Cryptanalysis on Block Cipher Skinny with MILP Program

Pei Zhang[1,2] and Wenying Zhang[1,2]

[1]*School of Information Science and Engineering, Shandong Normal University, Jinan, Shandong 250358, China*
[2]*Cyberspace Security Lab, Jinan, Shandong 250358, China*

Correspondence should be addressed to Wenying Zhang; wenyingzh@sohu.com

Received 27 July 2018; Revised 11 September 2018; Accepted 16 September 2018; Published 4 October 2018

Guest Editor: Zhaoqing Pan

With the widespread use of RFID technology and the rapid development of Internet of Things, the research of lightweight block cipher has become one of the hot issues in cryptography research. In recent years, lightweight block ciphers have emerged and are widely used, and their security is also crucial. Skinny-64/192 can be used to protect data security such as the applications of wireless multimedia and wireless sensor networks. In this paper, we use the new method to verify the security of Skinny-64/192. The method is called mixed-integer linear programming (MILP) which can characterize precisely the linear operation and nonlinear operation in a round function. By applying MILP program, we can automatically find a 11-round differential characteristic for Skinny-64/192 with the minimum number of active s-boxes. The probability of differential trail is 2^{-147}, that is, far greater than 2^{-192} which is the probability of success for an exhaustive search. In addition, comparing this method with the one proposed by Sun et al., we also have a great improvement; that is, no new variables will be added in ShiftRows operation. It can reduce greatly the number of variables and improve the running speed of the computer. Besides, the experimental result proves that Skinny-64/192 is safe on 11-round differential analysis and validates the effectiveness of the MILP method.

1. Introduction

Nowadays, with the development of big data and artificial intelligence technology, data security problem becomes increasingly serious. The problem of data security exists in the whole life cycle of data, from data collection and transfer to data usage, with the focus on data confidentiality [1, 2], integrity, and availability. Due to the continuous occurrence and intensification of data leakage events on the Internet, the confidentiality of data is particularly important. The confidentiality of data, on the one hand, is the direct protection of data and, on the other hand, is providing further privacy protection on the basis of leaks. From a technical point of view, to ensure the confidentiality of data without hindering the availability of data, the general approach is to encrypt the data by encryption algorithms.

The traditional encryption algorithms are DES [3] and AES [4]; they are widely used in the field of hardware and software. The hardware applications include data security in radio frequency IC card and encryption of hard disk data. And the software applications include voice, video information encryption, and database data encryption. The most classic applications are security applications on wireless network: one is the IEEE 803.11 protocol for WLAN and the IEEE 803.16 protocol for WMAN, and the other is the ZigBee protocol. The application of these encryption algorithms ensures the security of data effectively.

At present, the traditional cryptographic algorithm for encrypting wireless multimedia data faces a lot of challenges, such as fast resource consumption, high implementation cost, and other disadvantages. In this context, lightweight block cipher emerged. Compared with traditional cryptographic algorithms, lightweight block ciphers run faster and have lower resource consumption and implementation costs while ensuring data security. They are more suitable for radio frequency identification (RFID) tags, wireless sensor network

(WSN) [5], wireless multimedia, and other micro devices. In recent years, many lightweight block ciphers have been proposed, such as PRESENT [6], PRINCE [7], Midori [8], and Skinny [9, 10], many of which have been defined as ISO standards and widely used in various fields.

Skinny is a family of tweakable lightweight block ciphers proposed by Beierle et al. at CRYPTO in 2016. It is a Substitution Permutation Network (SPN) [11, 12] structure. It supports two block lengths n=64 or 128 bits and for each of them, the tweakey t can be either n, 2n, or 3n. Skinny has been analyzed by many methods since it was proposed, such as impossible differential cryptanalysis [10] and related-key impossible differential attack [13].

Differential cryptanalysis [3, 13, 14] was firstly introduced by Biham and Shamir to analyze DES block cipher in 1990. Differential analysis is one of the most effective attack methods in block ciphers. Differential analysis is a selective plaintext attack, and its basic idea is to study the probability of differential propagation of specific plaintext differential values in the encryption process. We separate the block cipher from the permutation area and then carry out the key recovery attack on this basis. In other words, we find a high probability differential trail. Finally, by adding several rounds before and after the differential characteristic, guessing Round-keys used in these rounds, encrypting plaintexts, and decrypting ciphertexts, we can determine the right key of block cipher.

Mixed-integer linear programming (MILP) [14, 15] is a mathematical optimization or feasibility scheme, where some or all variables are limited to integers. In many cases, the term refers to an integer linear program (ILP), which is linear in terms of objective function and constraint except for the integer constraint. MILP is frequently used in business and economics to solve problems of optimization.

In [14], Mouha et al. proposed an automatic search method based on MILP. However, the drawback of this method is that the proposed constraint cannot describe the trail of the differential propagation of the linear diffusion layer. Besides, Sun et al. [15] perfected the MILP method, then combined the MILP method and differential analysis into PRESENT, and finally obtained satisfactory experimental results. In this paper, we apply the new MILP method to obtain a lower bound on the number of active s-boxes for differential cryptanalysis. Then, we use the maximum differential probability of the s-boxes to derive an upper bound for the probability of the best characteristic.

The Organizational of the Paper. The paper is organized as follows: In Section 2, we introduce some basic properties and definitions and describe how to construct a MILP program by constraints to get the minimum number of active s-boxes and the corresponding trail of differential propagation. In Section 3, the MILP program is constructed to search the differential trail of Skinny-64/192. Through specific instances, the optimal solution of the minimum number of active s-boxes is obtained for 11-round differential characteristic of Skinny-64/192. We conclude the paper and look forward to the future work in Section 4. The auxiliary materials are given in Appendix.

Our Contributions. In this paper, we apply a method proposed recently for obtaining a high probability of differential characteristic in Skinny-64/192 called MILP method, which is used to search the minimum number of active s-boxes automatically. Its minimum number of active s-boxes is 54 of 11-round differential characteristic. The number of active s-boxes is one of the commonly used methods for evaluating the security of symmetric key encryption schemes against differential attack. As far as we know, this is the first time to combine differential analysis with MILP method to be applied to Skinny-64/192.

MILP can characterize accurately the linear operation and nonlinear operation in the round function. Then a high probability of 11-round differential characteristic is automatically searched. The probability of differential trail is 2^{-147}, that is, far greater than 2^{-192} which is the probability of success for an exhaustive search. This experimental result proves that 11-round differential analysis of Skinny-64/192 is safe, which can provide a safe reference for data encryption on wireless devices. At the same time, we also verified the effectiveness of MILP method through this experiment. In addition, we also have an improvement on MILP program; that is, no new variables can be added in ShiftRows operation, which can reduce the number of total variables greatly and improve the running speed of the computer.

2. The Minimum Number of Active S-Boxes for Differential Cryptanalysis

In this section, we will describe how to construct the MILP program to calculate the number of active s-boxes for differential analysis. This requires an accurate description of the nonlinear layers and linear layers in order to ensure the number of active s-boxes is minimum. In general, if a large number of active s-boxes exist, which indicates that the differential diffusion is fast, this suggests that the cryptographic algorithm is not vulnerable to attacks and has a high security. The core theorem for constructing the MILP program will be described in detail in the following.

2.1. Differential Cryptanalysis

Definition 1. For every input bit-level difference, a 0-1 variable x_i is introduced such that $x_i = 1$ if and only if the difference at this bit is nonzero, as

$$x_i = \begin{cases} 0, & \textit{the differences do not exist}, \\ 1, & \text{otherwise}. \end{cases} \tag{1}$$

2.2. Constraints for Nonlinear and Linear Operation. Generally, the SPN-structured encryption algorithm consists of s-box, XOR, ShiftRows, and MixColumn operations. In this subsection, we describe these four basic operations by constraints. Based on this, we can construct an r-round

inequality model for a specific encryption algorithm. This model can describe the trail of differential propagation accurately. Then by selecting the appropriate objective function, we can convert this model into a MILP program, using this MILP program to search automatically for the objective function.

Constraints Describing the S-Box Operation [15]. Suppose $(x_{i_0}, \ldots, x_{i_{w-1}})$ and $(y_{j_0}, \ldots, y_{j_{v-1}})$ are the input and output bit-level differences of a w×v s-box marked by S_t. Firstly, to ensure that $S_t = 1$ holds if and only if $(x_{i_0}, \ldots, x_{i_{w-1}})$ are not all zero, we require the following.

$$\begin{aligned} S_t - x_{i_k} &\geq 0, \quad k \in \{0, \ldots, w-1\} \\ x_{i_0} + x_{i_1} + \cdots + x_{i_w} - S_t &\geq 0 \end{aligned} \tag{2}$$

For bijective s-boxes, nonzero input difference must result in nonzero output difference and vice versa.

$$\begin{aligned} &wy_{j_0} + wy_{j_1} + \cdots + wy_{j_{v-1}} - \left(x_{i_0} + x_{i_1} + \cdots + x_{i_{w-1}}\right) \\ &\geq 0 \\ &vx_{j_0} + vx_{j_1} + \cdots + vx_{j_{w-1}} - \left(y_{i_0} + y_{i_1} + \cdots + y_{i_{v-1}}\right) \geq 0 \end{aligned} \tag{3}$$

Constraints Describing the XOR Operation. The bit-wise input difference is (x_i, x_{i+1}) and the corresponding bit-wise output difference is y for the XOR operation. The following linear constraints describe the relation between the input and output difference.

$$\begin{aligned} x_i + x_{i+1} - y &\geq 0 \\ x_i - x_{i+1} + y &\geq 0 \\ -x_i + x_{i+1} + y &\geq 0 \\ x_i + x_{i+1} + y &\leq 2 \end{aligned} \tag{4}$$

Constraints Describing the ShiftRows or ShuffleCell Operation. For every ShuffleCell operation, its input difference $(y_0, y_1, \ldots, y_{i-1}, y_i)$ and output difference $(z_0, z_1, \ldots, z_{i-1}, z_i)$ are based on bit. If $(z_0 = y_2, z_1 = y_i, \ldots, z_{i-1} = y_0, z_i = y_1)$, the constraints include the following.

$$\begin{aligned} z_0 - y_2 &= 0 \\ z_1 - y_i &= 0 \\ &\vdots \\ z_{i-1} - y_0 &= 0 \\ z_i - y_1 &= 0 \end{aligned} \tag{5}$$

Constraints Describing the MixColumn Operation. Let $(z_0, z_1, \ldots, z_{j-1}, z_j)$ and $(x_0, x_1, \ldots, x_{j-1}, x_j)$ be the input and output bit-wise differences for the MixColumn operation. Suppose $x_i = z_{j-2} + z_{j-1} + z_j$; it is essential to set an intermediate variable u and let $u = z_{j-2} + z_{j-1}$ to get $x_i = u + z_j$, so the constraints can be described as follows.

$$\begin{aligned} z_{j-2} + z_{j-1} - u &\geq 0 \\ z_{j-2} - z_{j-1} + u &\geq 0 \\ -z_{j-2} + z_{j-1} + u &\geq 0 \\ z_{j-2} + z_{j-1} + u &\leq 2 \\ u + z_j - x_0 &\geq 0 \\ u - z_j + x_0 &\geq 0 \\ -u + z_j + x_0 &\geq 0 \\ u + z_j + x_0 &\leq 2 \end{aligned} \tag{6}$$

Definition 2 (the objective function [16]). Some notations for differential are used in the model; e.g., S_j denotes the activity of an s-box and the objective function is as follows.

$$\begin{aligned} &\min \quad \sum_j S_j \\ &\text{s.t.} \quad S_j = \begin{cases} 0, & \textit{the sbox is not active}, \\ 1, & \textit{othersise}. \end{cases} \end{aligned} \tag{7}$$

The smaller the number of active s-boxes is, the slower the differential diffusion is. This illustrates that the encryption algorithm will be attacked in more rounds, and this will threaten its security.

3. Constructing the MILP Program to Calculate the Minimum Number of Active S-Boxes of Skinny-64/192

It is well known that the security of an encryption algorithm must be evaluated before being put into use. In this section, we use the newly proposed MILP method to evaluate the security of Skinny-64/192 on differential analysis. This is the first time to combine differential analysis with MILP method to be applied to Skinny-64/192; the method is called MILP program. The MILP program can automatically obtain the minimum number of active s-boxes on the 11-round differential analysis. And MILP program consists of inequalities which can describe precisely the linear and nonlinear operation.

3.1. Description of Skinny-64/192. Skinny is a family of tweakable lightweight block ciphers proposed by Beierle et

Table 1: S-box of Skinny-64/192.

x	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
s(x)	0xc	0x6	0x9	0x0	0x1	0xa	0x2	0xb	0x3	0x8	0x5	0xd	0x4	0xe	0x7	0xf

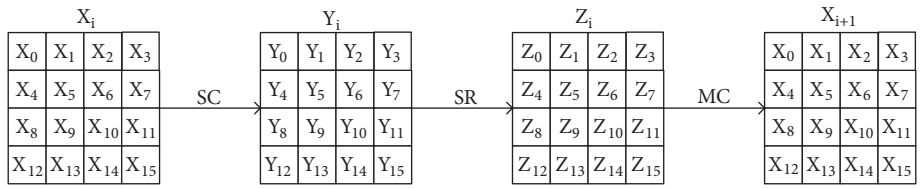


Figure 1: The i-th round function of Skinny-64/192.

al. at CRYPTO in 2016. The specifications for Skinny was given in [9]. Skinny-64/192 provides 64-bit block length and 192-bit key length. We now give a short description of Skinny-64/192. Skinny-64/192 uses the SPN structure with Midori-64-like state. The state is arranged in a 4×4 matrix.

$$P = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 & p_7 \\ p_8 & p_9 & p_{10} & p_{11} \\ p_{12} & p_{13} & p_{14} & p_{15} \end{pmatrix} \tag{8}$$

Every cell P_i is a nibble, $i \in 0, 1, \ldots, 15$.

The round function consists of SubCell, AddConstants, AddRoundTweakey, ShiftRows, and MixColumn. Since SubCell, ShiftRows, and MixColumn operations have an effect on differential diffusion, we only illustrate these operations in the paper. For more details, please refer to [9].

SubCells (SC). The 4×4 s-box defined in Table 1 is applied to each nibble in the state.

ShiftRows (SR). The rows of the state are rotated as in AES but to the right, i.e., the cell permutation is specified as follows.

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 & 1 & 2 & 3 \\ 7 & 4 & 5 & 6 \\ 10 & 11 & 8 & 9 \\ 13 & 14 & 15 & 12 \end{pmatrix} \tag{9}$$

MixColumn (MC). Each column in the state is multiplied by a binary matrix MC. MC is given as follows.

$$MC = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \tag{10}$$

Tweakey Schedule. Skinny-64/192 tweakey is updated through tweakey schedule. $K = TK1 \parallel TK2 \parallel TK3 = 192$ bits, and $TK1 = TK2 = TK3 = 64$ bits which be permuted by P_T. Then, each cell in the first and second rows of TK2, TK3 is updated using LFSR operations shown in [10].

In Figure 1, X_i, Y_i, Z_i represent the i-th round state, respectively. X_j, Y_j, Z_j represent a nibble of the state, respectively. Let $(x_0, x_1, \ldots, x_{63})$ and $(z_0, z_1, \ldots, z_{63})$ be the input and output bit-wise differences in a round for Skinny-64/192, and X_j, Y_j, Z_j, $j \in 0, 1, \ldots, 15$, and x_k, y_k, z_k, $k \in 0, 1, \ldots, 63$.

From the overall design of Skinny-64/192, its structure is compact and has the advantages of low delay, high throughput, and low number of gate circuits in hardware implementation. Therefore, Skinny-64/192 is more suitable for wireless multimedia and other micro device applications. Now, we apply the new MILP method to get the lower limit of the number of active s-boxes for the differential analysis of Skinny-64/192. Then, we use the maximum difference probability of the s-box to derive the upper bound of the best characteristic probability. Finally, the experimental results are used to determine whether Skinny-64/192 is safe on differential analysis.

3.2. Employing MILP's Method for Specific Operation

3.2.1. Compact Constraints for DDT of S-Box. In Skinny-64/192, combined with the Table 2, $y = s(x)$, $(y_0, y_1, y_2, y_3) = s1(x_0, x_1, x_2, x_3)$, it is possible to list the following vectors according to the input differential of 0001: [0, 0, 0, 1, 1, 0, 0, 0], [0, 0, 0, 1, 1, 0, 0, 1], [0, 0, 0, 1, 1, 0, 1, 0],

Table 2: The input and output differential distribution of Skinny-64/192's s-box.

Input difference (x_0, x_1, x_2, x_3)	Output difference (y_0, y_1, y_2, y_3)
0000	0000
0001	1000 1001 1010 1011
0010	0001 0011 0101 0110
0011	1000 1001 1010 1011 1100 1101 1110 1111
0100	0010 0110 0111 1011 1100 1101
0101	0010 0110 0111 1010 1100 1101
0110	0001 0011 0100 0111 1000 1010 1101 1110
0111	0001 0011 0100 0111 1001 1011 1100 1111
1000	0100 0101 1100 1101 1110 1111
1001	0100 0101 1100 1101 1110 1111
1010	0101 0110 1000 1001 1010 1011
1011	0001 0011 1100 1101 1110 1111
1100	0010 0110 0111 1000 1110 1111
1101	0010 0110 0111 1001 1110 1111
1110	0001 0011 0100 0111 1001 1011 1101 1110
1111	0001 0011 0100 0111 1000 1010 1100 1111

$[0, 0, 0, 1, 1, 0, 1, 1]$. Similarly, we can get all the differential vectors, so we get the input of SageMath [17].

points = [[0, 0, 0, 0, 0, 0, 0, 0] ,

[0, 0, 0, 1, 1, 0, 0, 0] , [0, 0, 0, 1, 1, 0, 0, 1] ,

[0, 0, 0, 1, 1, 0, 1, 0] , [0, 0, 0, 1, 1, 0, 1, 1] ,

[0, 0, 1, 0, 0, 0, 0, 1] , [0, 0, 1, 0, 0, 0, 1, 1] ,

[0, 0, 1, 0, 0, 1, 0, 1] , [0, 0, 1, 0, 0, 1, 1, 0] ,

[0, 0, 1, 1, 1, 0, 0, 0] , [0, 0, 1, 1, 1, 0, 0, 1] ,

[0, 0, 1, 1, 1, 0, 1, 0] , [0, 0, 1, 1, 1, 0, 1, 1] ,

[0, 0, 1, 1, 1, 1, 0, 0] , [0, 0, 1, 1, 1, 1, 0, 1] ,

[0, 0, 1, 1, 1, 1, 1, 0] , [0, 0, 1, 1, 1, 1, 1, 1] ,

[0, 1, 0, 0, 0, 0, 1, 0] , [0, 1, 0, 0, 0, 0, 1, 1, 0] ,

[0, 1, 0, 0, 0, 1, 1, 1] , [0, 1, 0, 0, 1, 0, 1, 1] ,

[0, 1, 0, 0, 1, 1, 0, 0] , [0, 1, 0, 0, 1, 1, 0, 1] ,

[0, 1, 0, 1, 0, 0, 1, 0] , [0, 1, 0, 1, 0, 1, 1, 0] ,

[0, 1, 0, 1, 0, 1, 1, 1] , [0, 1, 0, 1, 1, 0, 1, 0] ,

[0, 1, 0, 1, 1, 1, 0, 0] , [0, 1, 0, 1, 1, 1, 0, 1] ,

[0, 1, 1, 0, 0, 0, 0, 1] , [0, 1, 1, 0, 0, 0, 1, 1] ,

[0, 1, 1, 0, 0, 1, 0, 0] , [0, 1, 1, 0, 0, 1, 1, 1] ,

[0, 1, 1, 0, 1, 0, 0, 0] , [0, 1, 1, 0, 1, 0, 1, 0] ,

[0, 1, 1, 0, 1, 1, 0, 1] , [0, 1, 1, 0, 1, 1, 1, 0] ,

[0, 1, 1, 1, 0, 0, 0, 1] , [0, 1, 1, 1, 0, 0, 1, 1] ,

[0, 1, 1, 1, 0, 1, 0, 0] , [0, 1, 1, 1, 0, 1, 1, 1] ,

[0, 1, 1, 1, 1, 0, 0, 1] , [0, 1, 1, 1, 1, 0, 1, 1] ,

[0, 1, 1, 1, 1, 1, 0, 0] , [0, 1, 1, 1, 1, 1, 1, 1] ,

[1, 0, 0, 0, 0, 1, 0, 0] , [1, 0, 0, 0, 0, 1, 0, 1] ,

[1, 0, 0, 0, 1, 1, 0, 0] , [1, 0, 0, 0, 1, 1, 0, 1] ,

[1, 0, 0, 0, 1, 1, 1, 0] , [1, 0, 0, 0, 1, 1, 1, 1] ,

[1, 0, 0, 1, 0, 1, 0, 0] , [1, 0, 0, 1, 0, 1, 0, 1] ,

[1, 0, 0, 1, 1, 1, 0, 0] , [1, 0, 0, 1, 1, 1, 0, 1] ,

[1, 0, 0, 1, 1, 1, 1, 0] , [1, 0, 0, 1, 1, 1, 1, 1] ,

[1, 0, 1, 0, 0, 1, 0, 1] , [1, 0, 1, 0, 0, 1, 1, 0] ,

[1, 0, 1, 0, 1, 0, 0, 0] , [1, 0, 1, 0, 1, 0, 0, 1] ,

[1, 0, 1, 0, 1, 0, 1, 0] , [1, 0, 1, 0, 1, 0, 1, 1] ,

[1, 0, 1, 1, 0, 0, 0, 1] , [1, 0, 1, 1, 0, 0, 1, 1] ,

[1, 0, 1, 1, 1, 1, 0, 0] , [1, 0, 1, 1, 1, 1, 0, 1,] ,

[1, 0, 1, 1, 1, 1, 1, 0] , [1, 0, 1, 1, 1, 1, 1, 1] ,

[1, 1, 0, 0, 0, 0, 1, 0] , [1, 1, 0, 0, 0, 1, 1, 0] ,

[1, 1, 0, 0, 0, 1, 1, 1] , [1, 1, 0, 0, 1, 0, 0, 0] ,

$$
\begin{gathered}
[1,1,0,0,1,1,1,0],[1,1,0,0,1,1,1,1],\\
[1,1,0,1,0,0,1,0],[1,1,0,1,0,1,1,0],\\
[1,1,0,1,0,1,1,1],[1,1,0,1,1,0,0,1],\\
[1,1,0,1,1,1,1,0],[1,1,0,1,1,1,1,1],\\
[1,1,1,0,0,0,0,1],[1,1,1,0,0,0,1,1],\\
[1,1,1,0,0,1,0,0],[1,1,1,0,0,1,1,1],\\
[1,1,1,0,1,0,0,1],[1,1,1,0,1,0,1,1],\\
[1,1,1,0,1,1,0,1,],[1,1,1,0,1,1,1,0],\\
[1,1,1,1,0,0,0,1],[1,1,1,1,0,0,1,1],\\
[1,1,1,1,0,1,0,0],[1,1,1,1,0,1,1,1],\\
[1,1,1,1,1,0,0,0],[1,1,1,1,1,0,1,0],\\
[1,1,1,1,1,1,0,0],[1,1,1,1,1,1,1,1]]
\end{gathered}
\tag{11}
$$

Running SageMath will output 202 inequalities. Then, the redundant inequalities are eliminated through a specific streamlined procedure (Appendix). Finally, the s1-box can be accurately characterized with 24 inequalities. The inequality of describing s1 is shown as follows.

$$
\begin{aligned}
-2x_0 + 3x_1 - 3x_2 - 2x_3 + 5y_0 + 4y_1 + y_2 + 7y_3 &>= 0\\
-x_0 + x_1 + 2x_2 + x_3 + y_0 + 3y_1 - 2y_3 &>= 0\\
4x_0 + 3x_1 + 2x_2 + 3x_3 - y_0 - y_1 - y_2 - y_3 &>= 0\\
2x_0 - x_1 + 2x_2 + 3y_0 - y_1 + 3y_2 - y_3 &>= 0\\
x_0 + 3x_1 + x_2 - 2x_3 + 2y_0 - y_1 - 2y_2 &>= -2\\
-3x_0 + 2x_1 + x_2 - 2x_3 - y_0 + 3y_1 + y_3 &>= -3\\
x_1 - 2x_2 + 2x_3 - y_0 - 2y_1 + y_2 + y_3 &>= -3\\
-2x_0 - 3x_1 + 2x_2 + x_3 + y_0 - y_1 + 3y_2 - y_3 &>= -4\\
-x_1 - 2x_2 - x_3 + y_0 - y_1 - 2y_2 + 2y_3 &>= -5\\
-x_1 - x_2 - x_3 + y_0 - 2y_1 + 2y_2 - 2y_3 &>= -5\\
2x_0 + x_1 + 3x_2 + 4x_3 - 3y_0 + 2y_1 - y_2 + 3y_3 &>= 0\\
x_0 - 2x_1 + 2x_2 - 2x_3 - y_0 + y_1 - y_2 - 2y_3 &>= -6\\
x_1 - 2x_2 + 2x_3 - y_0 - 2y_1 - y_2 - y_3 &>= -5\\
x_0 - x_1 - x_3 + y_0 + 2y_1 + 2y_2 + 2y_3 &>= 0\\
x_0 - x_2 + x_3 - y_0 + y_1 - y_3 &>= -2\\
-x_0 - x_2 - x_3 - y_0 + y_1 - y_3 &>= -4\\
-x_0 - x_1 - x_2 + x_3 + y_1 + y_3 &>= -2\\
3x_0 + x_1 + x_2 - 2x_3 + 2y_0 - y_1 + 2y_2 - y_3 &>= -1\\
x_1 + x_2 + y_0 - y_2 &>= 0\\
x_0 - x_1 - 2x_2 - x_3 + y_0 - 2y_2 + 2y_3 &>= -4\\
x_0 + x_1 + x_2 + 2x_3 - 2y_0 + y_1 + y_2 &>= 0\\
-x_0 - x_1 + x_2 - y_1 + y_2 &>= -2\\
x_0 + x_1 + x_2 - y_1 &>= 0\\
x_0 + x_2 - y_0 - y_1 - y_2 &>= -2
\end{aligned}
\tag{12}
$$

A round of 16 s-boxes can be characterized by 384 linear inequalities accurately. $x_0, x_1, x_2, x_3 \in X_0, y_0, y_1, y_2, y_3 \in Y_0$. X_0, Y_0 represent 4 bits. X_0 and Y_0 are input and output of s1, respectively.

In contrast, it is simple to construct constrains for the ShiftRows operation of Skinny-64/192. Referring to (5), the ShiftRows operation can be characterized precisely by the next 64 constraint equations. $y_i \in Y$, $z_i \in Z$, $i \in 0, 1, \ldots, 63$.

$$
\begin{aligned}
z_0 - y_0 &= 0\\
z_1 - y_1 &= 0\\
z_2 - y_2 &= 0\\
z_3 - y_3 &= 0\\
&\vdots\\
&\vdots\\
z_{60} - y_{48} &= 0\\
z_{61} - y_{49} &= 0\\
z_{62} - y_{50} &= 0\\
z_{63} - y_{51} &= 0
\end{aligned}
\tag{13}
$$

In the ShiftRows operation, comparing with the method in [15], we also make a great improvement in which no new variables will be added. It reduces the number of variables greatly and improves the running speed of the computer. In this case, we can reduce the use of 64 variables in one round. Therefore, the variable Z can be omitted.

3.2.2. Compact Constraints for the Linear Transform. In linear layer, MixColumn operations are the most difficult to be described utilizing the novel technique, but in this work, we can introduce an intermediate variable U to solve the problems. This operation is broken down into the following steps.

Table 3: The DDT of s-box for Skinny-64/192.

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1	0	0	0	0	0	0	0	0	4	4	4	4	0	0	0	0
0x2	0	4	0	4	0	4	4	0	0	0	0	0	0	0	0	0
0x3	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2
0x4	0	0	4	0	0	0	2	2	0	0	0	4	2	2	0	0
0x5	0	0	4	0	0	0	2	2	0	0	4	0	2	2	0	0
0x6	0	2	0	2	2	0	0	2	2	0	2	0	0	2	2	0
0x7	0	2	0	2	2	0	0	2	0	2	0	2	2	0	0	2
0x8	0	0	0	0	4	4	0	0	0	0	0	0	2	2	2	2
0x9	0	0	0	0	4	4	0	0	0	0	0	0	2	2	2	2
0xa	0	0	0	0	0	4	4	0	2	2	2	2	0	0	0	0
0xb	0	4	0	4	0	0	0	0	0	0	0	0	2	2	2	2
0xc	0	0	4	0	0	0	2	2	4	0	0	0	0	0	2	2
0xd	0	0	4	0	0	0	2	2	0	4	0	0	0	0	2	2
0xe	0	2	0	2	2	0	0	2	0	2	0	2	0	2	2	0
0xf	0	2	0	2	2	0	0	2	2	0	2	0	2	0	0	2

Step 1. Convert the matrix $MC_{4\times4}$ to $MC_{16\times16}$ of Skinny-64/192.

$$MC = \begin{pmatrix} 1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0\\ 0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0\\ 0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0\\ 0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1\\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0\\ 1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0 \end{pmatrix} \quad (14)$$

Step 2. After the MixColumn, we can get the value x_s of X_{i+1}, in which $s \in 64, 65, \ldots, 127$. For example $x_{64} = y_0 + y_{40} + y_{52}$.

Step 3. We introduce the intermediate variable U, $u_0 = y_{40} + y_{52}$, and then $x_{64} = y_0 + u_0$. Combining (4) and (6), the constraints between them can be expressed as follows.

$$\begin{aligned} y_{40} + y_{52} - u_0 &>= 0\\ y_{40} - y_{52} + u_0 &>= 0\\ -y_{40} + y_{52} + u_0 &>= 0\\ y_{40} + y_{52} + u_0 &<= 2\\ u_0 + y_0 - x_{64} &>= 0\\ u_0 - y_0 + x_{64} &>= 0\\ -u_0 + y_0 + x_{64} &>= 0\\ u_0 + y_0 + x_{64} &<= 2 \end{aligned} \quad (15)$$

In a round, we need to use 16 intermediate variables, $u_i \in U$, $i \in 0, 1, \ldots, 15$.

3.3. Calculate the Minimum Number of Active S-Boxes. In the MILP program, we must add a linear constraint to ensure that at least one s-box is active. The setting of objective function refers to (7). And all variables must be binary variable. In order to optimize the MILP model, we also need CPLEX [18] tool. Finally, we obtain the minimum number of active s-boxes which is 54 for 11-round differential analysis of Skinny-64/192. In Table 3, the

Table 4: 11-round differential trail.

Rounds	Input difference	Probability	Active s-boxes
1st	9000 0000 0000 0000	2^{-2}	1
2nd	5505 0000 0000 0000	2^{-8}	3
3rd	cc0c 00a0 0000 d000	2^{-13}	5
4th	6606 0060 0000 096f	2^{-21}	7
5th	aaff 8008 f000 a077	2^{-26}	10
6th	0505 0f40 bff0 c505	2^{-30}	10
7th	c00d 00d0 0d7d 2700	2^{-23}	8
8th	0f00 f000 0000 0196	2^{-13}	5
9th	0090 0400 7000 0000	2^{-9}	3
10th	0000 0000 0d00 0000	2^{-2}	1
11th	0000 0000 0000 0800	–	1

differential distribution table of s-box of Skinny-64/192 is presented.

First, the s1-box of Skinny-64/192 is set as an active s-box with the input difference of 1001(9). And the probability of obtaining the second round of input difference is 2^{-2}, and the number of active s-boxes is 3. By analogy, the output difference of the 11th round has an active s-box which is 1000(8). The total probability of the 11- round differential characteristic is 2^{-147}. The minimum number of the active s-boxes is 54 for 11 rounds of Skinny-64/192. The details are shown in Table 4.

According to Table 4, we can get a specific probability for each round of 11-round differential characteristic for Skinny-64/192 in Figure 2. The probability of differential trail is 2^{-147}, that is, far greater than 2^{-192} which is the probability of success for an exhaustive search.

The experimental result leads us to obtain the minimum number of active s-boxes which is 54 for the 11-round differential trail. Since the same number of rounds is attacked, the minimum active s-boxes number of the Skinny-64/192 is bigger than that of ENOCORO-128v2, PRESENR-80. This not only illustrates that Skinny-64/192 is relatively safe, but also can be implemented in hardware to protect the safety of data. The bigger the number of active s-boxes, the faster the differential diffusion; the security of cryptographic algorithm is relatively higher.

The MILP program corresponding to Skinny-64/192's 11-round differential trail consists of 7440 constraints and 1680 binary variables including 1520 continuous variables and 160 intermediate variables. Compared with Sun et al., our improvement reduced the use of 640 continuous variables in total and improved the speed of the computer. The experiments are implemented on a 64-bit operating system, Intel Core i7-7700 CPU @ 3.60GHz, with 16GB of RAM.

4. Conclusion

In this paper, a new result is obtained on the differential analysis of lightweight block cipher Skinny-64/192. We get a 11-round differential characteristic with minimum active s-boxes. The minimum number of active s-boxes is 54. The probability of 11-round differential trail is 2^{-147}, that is, far greater than 2^{-192} which is the probability of success for an exhaustive search.

The experimental result not only proves that Skinny-64/192 cannot resist 11-round differential analysis and validates the effectiveness of MILP method, but also has other important reference values. First, the lightweight block cipher Skinny-64/192 is relatively secure and can be used on wireless multimedia devices to protect data security. Second, Skinny-64/192 can resist 11-round differential analysis, so it can be used as a candidate encryption algorithm for differential privacy protection technology. Finally, by verifying the effectiveness of MILP method on differential analysis, the method can significantly reduce the workload of cryptanalysts. Besides, MILP method can be applied to more cryptanalysis, such as related-key differential analysis, impossible differential analysis, and related-key impossible differential analysis. We believe that there will be greater gains.

Appendix

The specific streamlined procedure to select certain number of inequalities (see Algorithm 1).

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

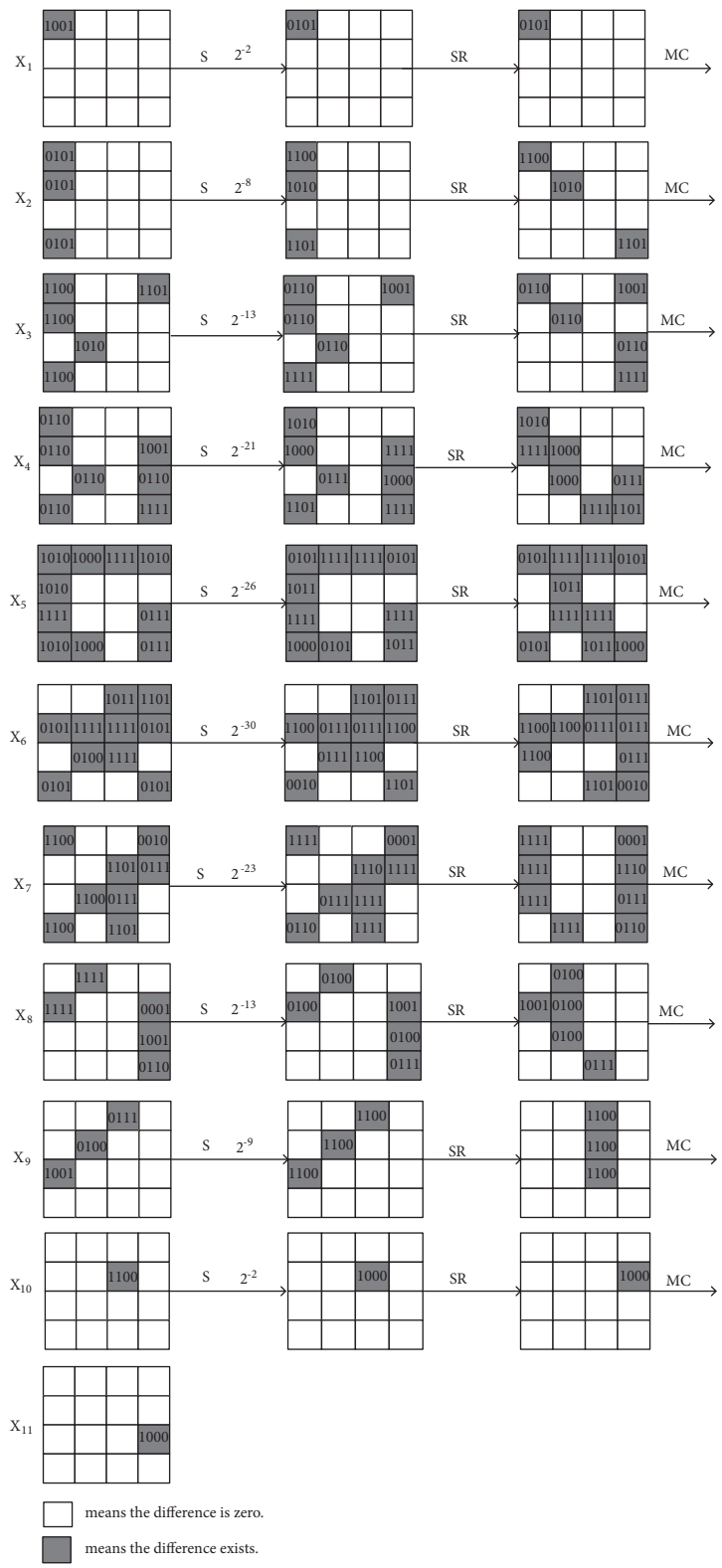


Figure 2: 11-round differential characteristic of Skinny-64/192.

```
#include <stdio.h>
# define N1 300
# define N2 200
# define M 9
int choose(int x[N1][M],int y[N2][M-1])
{
    int i, j, temp;
    int z[N1]={0};
    // How many points are not satisfied for each inequality.
    for (i=0;i<N1;i++)
    {
        for(j=0;j<N2;j++)
            if((x[i][0]*y[j][0]+x[i][1]*y[j][1]+x[i][2]*y[j][2]+x[i][3]*y[j][3]+x[i][4]*y[j][4]+x[i][5]*y[j][5]
              +x[i][6]*y[j][6]+x[i][7]*y[j][7]+x[i][8])<0)
              z[i]++;
     }
    temp=z[0]; j=0;
    // Finding the inequality and its count is the largest.
    for(i=1;i<N1;i++)
     {
        if(z[i]>temp)
        {j=i;temp=z[j];}
     }
    if(temp!=0)
    {
        // Delete the points corresponding to the largest inequality.
        for(i=0;i<N2;i++)
          {
            if(x[j][0]*y[i][0]+x[j][1]*y[i][1]+x[j][2]*y[i][2]+x[j][3]*y[i][3]+x[j][4]*y[i][4]+x[j][5]*y[i][5]
              +x[j][6]*y[i][6]+x[j][7]*y[i][7]+x[j][8]<0)
            {
              y[i][0]=0;y[i][1]=0;y[i][2]=0;y[i][3]=0;y[i][4]=0;y[i][5]=0;y[i][6]=0;
              y[i][7]=0;
             }
          }
        // Output inequality and the number of points that are not satisfied.
        for(i=0; i<8;i++)
        {
          if(x[j][i]<0||i==0)
            printf("%d*x%d",x[j][i],i+1);
          else
            printf("+%d*x%d",x[j][i],i+1);
         }
        printf("+%d%6d",x[j][8],temp);printf("\n");
    x[j][0]=0;x[j][1]=0;x[j][2]=0;x[j][3]=0;x[j][4]=0;x[j][5]=0;x[j][6]=0;x[j][7]=0;
    x[j][8]=0;
        return temp;
     }
    else
        return 0;
 }
void main()
{
//In SageMath, the coefficients of the inequality of the s1-box are obtained.
//Because there are so many points, I'll just list some of them here.
        int a[N1][M]={{0,-1,0,0,0,0,0,0,1},{-1,0,0,0,0,0,0,0,1}, . . . , {-1,-1,-1,-1,-1,0,1,-1,5}, {-1,-2,-1,-2,-1,-1,2,-2,8}};
  //It doesn't satisfy the s1-box.
  //Because there are so many points, I'll just list some of them here.
        int b[N2][M-1]={{0, 0, 0, 0, 0, 0, 0, 1}, {0, 0, 0, 0, 0, 0, 1, 0}, . . . , {1, 1, 1, 1, 1, 1, 0, 1}, {1, 1, 1, 1, 1, 1, 1, 0}};
    printf("     inequalities                                    counting\n");
    while(choose(a, b)!=0)
    choose(a, b);
 }
```

Algorithm 1

Acknowledgments

This work is partially supported by National Natural Science Foundation of China (Nos. 61672330, 61602287, and 61802235), the Key Research Development Project of Shandong Province (No. 2016GGX101024), China Postdoctoral Science Foundation (2018M632712), the Key Research and Development Plan of Shandong Province (2018GGX101037), and the Major Innovation Project of Science and Technology in Shandong Province (2018CXGC0702).

References

[1] L. S. Melro and L. R. Jensen, "Influence of functionalization on the structural and mechanical properties of graphene," *Computers, Materials and Continua*, vol. 53, no. 2, pp. 111–131, 2017.

[2] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[3] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[4] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-The Advanced Encryption Standard*, Springer, Berlin, Germany, 2002.

[5] G. Cheng, C. Yang, X. Yao et al., "When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 99, pp. 1–11, 2018.

[6] M. H. Faghihi Sereshgi, M. Dakhilalian, and M. Shakiba, "Biclique cryptanalysis of MIBS-80 and PRESENT-80 block ciphers," *Security and Communication Networks*, vol. 9, no. 1, pp. 27–33, 2016.

[7] J. Borghoff, A. Canteaut, T. Güneysu et al., "PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications," in *Advances in Cryptology – ASIACRYPT 2012*, vol. 7658 of *Lecture Notes in Computer Science*, pp. 208–225, Springer Berlin Heidelberg, Heidelberg, Germany, 2012.

[8] S. Banik, A. Bogdanov, T. Isobe et al., "Midori: a block cipher for low energy," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, Lecture Notes in Comput. Sci., pp. 411–436, Springer Berlin, 2014.

[9] C. Beierle, J. Jean, Moradi A. et al., "The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS," in *Proceedings of the Part II, of the 36th Annual International Cryptology Conference on Advances in Cryptology-CRYPTO*, vol. 9815, pp. 123–153, Springer-Verlag, 2016.

[10] M. Tolba, A. Abdelkhalek, and A. M. Youssef, "Impossible Differential Cryptanalysis of Reduced-Round SKINNY," 2017.

[11] G. Han and W. Zhang, "Improved biclique cryptanalysis of the lightweight block cipher piccolo," *Security and Communication Networks*, vol. 2017, 2017.

[12] Y. Zheng, B. Jeon, and L. Sun, "Student's t-Hidden Markov Model for Unsupervised Learning Using Localized Feature Selection," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 99, no. 1-1, 2017.

[13] R. Ankele, S. Banik, A. Chakraborti et al., "Related-key impossible-differential attack on reduced-round Skinny," in *Applied Cryptography and Network Security*, vol. 10355 of *Lecture Notes in Comput. Sci.*, pp. 208–228, Springer, Cham, 2017.

[14] N. Mouha, Q. Wang, D. Gu, and B. Preneel, "Differential and linear cryptanalysis using mixed-integer linear programming," in *Information Security and Cryptology*, pp. 57–76, Springer, 2012.

[15] S. Sun, L. Hu, P. Wang et al., "Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES (L) and Other Bit-Oriented Block Ciphers, ASIACRYPT," 2014.

[16] L. Sun, W. Wang, R. Liu, and M. Wang, "MILP-aided bit-based division property for ARX ciphers," *Science China Information Sciences*, vol. 61, no. 11, Article ID 118102, 2018.

[17] R. A. Mezei, *An Introduction to SAGE Programming: With Applications to SAGE Interacts for Numerical Methods*, Inc, 2015.

[18] "Division C.: Using the CPLEX Callable Library," 1997.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 8172725, 13 pages
https://doi.org/10.1155/2018/8172725

Research Article

Deep Learning Hash for Wireless Multimedia Image Content Security

Yu Zheng,[1] **Jiezhong Zhu**,[1] **Wei Fang**,[1] **and Lian-Hua Chi**[2]

[1]*School of Computer & Software, Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing 210044, Jiangsu, China*
[2]*Department of Computer Science and Information Technology, La Trobe University, VIC 3086, Australia*

Correspondence should be addressed to Yu Zheng; yzheng@nuist.edu.cn

Received 24 July 2018; Accepted 30 August 2018; Published 25 September 2018

Academic Editor: Weizhi Meng

With the explosive growth of the wireless multimedia data on the wireless Internet, a large number of illegal images have been widely disseminated in wireless networks, which seriously endangers the content security of wireless networks. However, how to identify and classify illegal images quickly, accurately, and in real time is a key challenge for wireless multimedia networks. To avoid illegal images circulating on the Internet, each image needs to be detected, extracted features, and compared with the image in the feature library to verify the legitimacy of the image. An improved image deep learning hash (IDLH) method to learn compact binary codes for image search is proposed in this paper. Specifically, there are three major processes of IDLH: the feature extraction, deep secondary search, and image classification. IDLH performs image retrieval by the deep neural networks (DNN) as well as image classification with the binary hash codes. Different from other deep learning-hash methods that often entail heavy computations by using a conventional classifier, exemplified by K nearest neighbor (K-NN) and support vector machines (SVM), our method learns classifiers using binary hash codes, which can be learned synchronously in training. Finally, comprehensive experiments are conducted to evaluate IDLH method by using CIFAR-10 and Caltech 256 image library datasets, and the results show that the retrieval performance of IDLH method can effectively identify illegal images.

1. Introduction

With the rapid development of multimedia technology, the retrieval of wireless multimedia data has become very convenient and easy. The security of multimedia data has attracted more and more attention from researchers. Multimedia data content security belongs to the branch of information security and requires direct understanding and analysis of the information content transmitted in the network. Judging rapidly from the massive information, filtering, and monitoring the abnormal information in the network are the key to ensure the security of the wireless network content. Meanwhile, a large number of illegal images are disseminated in the network, which seriously endangers the security of network content. So, it is very important practical significance to research the content security-oriented image recognition technology and identify and supervise illegal image information in the network. Although there are a lot of image retrieval methods currently, due to the various disadvantages, such as the low expression ability of image feature, high dimension of feature, and low precision of image retrieval, the retrieval results are not always effective.

How to retrieve the large-scale image resources quickly and effectively meet the needs is urgently to solve. Since the hash-based learning algorithm can effectively preserve the similarity between the original feature spaces and the hash code spaces, more and more scholars have drawn attention to it. Particularly, learning deep hash algorithm has greatly improved the retrieval performance.

Besides the widely used text-based search methods, content-based image retrieval (CBIR) has attracted widespread attention of more and more scholars in the past decade [1]. As we all know, the nearest neighbor search algorithm is an effective method for searching for similar data samples [2]. We should not only consider the scalability issue, but also consider most practical large-scale applications which are

affected by dimensional disasters [3]. The high-level encoding complexity prevents widespread adoption in real-time multimedia systems [4]. One of the many practical applications, the approximate nearest neighbors (ANN), is very efficient. However, the method requires huge storage costs and hard to handling high-dimensional data. Hence, quantization techniques have been proposed to encode high-dimensional data vectors recently. Due to the fact that hashing-based ANN search techniques can reduce the storage via storing the compact binary codes, hash technology is widely used in computer vision, machine learning, information retrieval, and other related fields, in view of the retrieval of large-scale data. The goal of hash is to turn high-dimensional data into the low-dimensional compact binary codes [5]. For example, d dimension data is turned into r dimension (d>>r), and generally r dimension data is between dozens of bits and hundreds of bits. After turning high-dimensional data into hash code, we calculate the distance or similarity between the data rapidly. Because of the high efficiency of binary hash code in Hamming distance calculation and the advantages of storage space, hash code is very efficient in large-scale image retrieval.

One of the most famous jobs in hash and the most applied job in the industry was locality sensitive hashing (LSH) [3], which was put forward by Gionis, Indyk, and Motwani in 1999[1]. LSH is one of the most popular data independent methods, which generates hash functions by random projections [6]. In addition to traditional Euclidean distance LSH has been generalized to accommodate other distance and similarity measures such as p-norm distance [3], Mahalanobis metric [7], and kernel similarity [8]. And L. Qi, X. Zhang, W. Dou, and Q. Ni put forward another application direction of LSH in 2017 [9]. They proposed a privacy-preserving and scalable service recommendation approach based on distributed LSH (SerRecdistri-LSH). The purpose of this application is to use SerRecdistri-LSH method to handle service recommendation in distributed cloud environment.

A disadvantage of the LSH family is that LSH usually needs long bit length (≥ 1000) to achieve both high precision and recall. This may leads to a huge storage overhead and thus limits the sale at which an LSH algorithm may be applied. In order to achieve the desired search accuracy, LSH often requires the long hash codes, thereby reducing the recall rate. This problem can be alleviated by using multiple hash tables, but it greatly increases storage costs and query time. [10].

So, learning-based data-dependent hashing methods have become increasingly popular because of the benefit that learned compact binary codes can effectively and highly efficiently index and organize massive data [3]. The goal of the data-dependent hashing methods is to generate short hash codes (typically ≤ 200) using available training data. Various hashing algorithms have been proposed in the literature, of which a large category focuses on linear hashing algorithms which learn a set of hyperplanes as linear hash functions. The representative algorithms in this category include unsupervised PCA Hashing [11], Iterative Quantization (ITQ) [12], and Isotropic Hashing [13] and supervised Minimal Loss Hashing (MLH) [14], Semisupervised Hashing (SSH) [11], Supervised Discrete Hashing (SDH) [6], Ranking-Based Supervised Hashing [15], FastHash [16], etc. A bilinear form of hash functions was introduced by [17, 18].

As an extension of linear hash functions, a variety of algorithms have been proposed to generate nonlinear hash functions in a kernel space, including Binary Reconstructive Embedding (BRE) [10], Random Maximum Margin Hashing (RMMH) [14], Kernel-Based Supervised Hashing (KSH) [12], and the kernel variant of ITQ [13]. In parallel, harnessing nonlinear manifold structures has been shown to be effective in producing compact neighborhood-preserving hash codes. The early algorithm in this fashion is Spectral Hashing (SH) [19], which produces hash codes through solving a continuously relaxed mathematical program similar to Laplacian Eigenmaps. More recently, Anchor Graph Hashing (AGH) [20] leveraged anchor graphs for solving the eigenfunctions of the resulting graph Laplacians, making hash code training and out-of-sample extension to novel data both tractable and efficient for large-scale datasets. Shen et al. [6, 21, 22] proposed a general induction.

The rest of this paper is organized as follows. Section 2 briefly reviews the related works about deep-learning-hashing. And we present framework and steps of the improved deep learning-based hash, IDLH, algorithm in detail in Section 3. Section 4 evaluates the effectiveness of the IDLH through a series of contrastive experiments and carefully analyzes the experimental results, followed by the conclusion in Section 5.

2. Related Work

Recently, deep-learning-hashing, as a popular research topic, has drawn increasing attention and research efforts in information retrieval, computer vision, and machine learning. Semantic hashing is recognized as the earliest starting deep learning hash [23]. This method establishes a deep generative model to discover hidden binary features. Such a deep model is made as a stack of restricted Boltzmann machines (RMBs) [24]. After learning a multilayer RBM by pretraining and fine-tuning the document collection, the hash code of any document is obtained by simply thresholding the deepest output. Such a hash code provided by deep RBM is shown to maintain a semantically similar relationship of input documents into code space, where each hash code (or hash key) is used as a memory address to locate the corresponding document. In this way, semantically similar documents are mapped to adjacent memory addresses, enabling efficient searching through hash table lookups. In order to improve the performance of deep RBMS, a supervised version was proposed in [25] and the idea of nonlinear neighborhood component analysis (NCA) embedding in [26] was adopted. The supervised information is derived from training a given neighbor/nonneighbor relationship between samples. Then, based on the depth RBM, the objective function of the NCA is optimized so that the depth RBM yields a hash code. Note that supervised deep RBMS can be applied to wide data fields other than text data. In [25], the depth RBMs are supervised using a Gaussian distribution, and the models of

the visible units in the first layer are successfully applied to the processing of massive image data.

In [27], a deep neural network was developed to learn multilevel nonlinear transformations, mapping the original image to a compact binary hash code to support large-scale image retrieval for learning binary image representations. A deep hashing model is established under the three-layer constraint on the deep network: (1) the reconstruction error between the original real-valued image feature vector and the resulting binary code is minimized; (2) each bit of the binary code has a balance; (3) all bits are independent of each other. Similar constraints were used in previous unsupervised hash or binary coding methods, such as iterative quantization (ITQ) [28]. In [27], a supervised version is called supervised depth hash 3, in which a discriminative item containing two pairs of supervised information is added to the objective function of the deep hash model.

It is worth noting that, in the training of sparse neural networks, in addition to the sparse similarity maintaining hash, depth hash, and supervised depth hash, the pretraining phase is not included. Instead, the hash codes are learned from scratch using a set of training data. However, no pretraining can make the general hash code less efficient. In particular, the sparse similarity keep hash method is found to be inferior to the existing supervised hashing method, i.e., kernel-based supervised hashing (KSH) [29], in terms of search accuracy on some image datasets [30]; the deep hash method and its supervised version are slightly better than ITQ and its supervised version CCA+ ITQ, respectively [31]. Note that KSH, ITQ, and CCA+ITQ develop a shallow learning framework.

One of the main purposes of deep learning is to learn the robust and powerful representation of complex data. It is very natural to use deep learning to explore compact hash codes, which can be thought of as binary representation of the data. The deployed CNN consists of three convolution pools, including rectified linear activation, maximum pool merging, and local contrast normalization, a standard fully connected layer, and an output layer with softmax activation functions. In [32], a new method called deep semantic sorting hashing is proposed to learn hash values, thereby maintaining multilevel semantic similarity between multilabel images. This method is combined with convolutional neural network hashing method, taking image pixels as input, training depth CNN, and jointly learning image feature representation and hash value by this method. The deployed CNN consists of five convolution-pooling layers, two fully connected layers, and a hash layer (i.e., output layer).

Indexing massive amounts of multimedia data, such as images and videos, is a natural application based on learning hashing. In particular, due to the well-known semantic divide, hashing methods have been extensively studied for image search and retrieval, as well as mobile product search [33, 34]. Although hashing techniques have been applied to the active learning framework to cope with big data applications, these hash algorithms are rarely applied to image classification. For the image recognition problem with many categories, the computational and memory overhead primarily stems from the large number of classifiers to be learned. Table 1 show that the complexities can be high at the stages of both training and deploying these classifiers. Inspired by [35, 36], we proposed a combination classifiers with DNN and binarizing classifiers to solve some classic security calculation problems [37–39]. Different from other deep learning-hash methods that often entails heavy computations by using a conventional classifier, exemplified by K-NN and SVM, our method learns classifiers using binary hash codes, which are simultaneously learned from the training data. The combination classifiers can provide both image features and accelerate image classification, and thus it make the large-scale image recognition faster. The advantages of the extending hashing techniques from fast image search to image classification inspire us to apply them to deep learning hash framework.

Table 1: A huge number of parameters need to be learned and stored with different datasets. Considering a classification task with C different categories and D-dimensional feature representation, even the simplest linear models are comprised of D×C parameters.

Image Dataset	Categories	Dimensions	Parameters
ImageNet	21,841	4,096	89,460,736
ILSVRC	1,000	4,096	4,096,000
SUN397	397	12,288	4,878,336
CIFAR-10	10	4,096	40,960
Caltech-256	256	4,096	1,048,576

With the development of Internet technology, the spreading form of illegal information on the Internet is changing gradually. Traditionally, the form of communication based on word description has been transformed into a diversified form of communication based on video and image. Therefore, the original keyword blocking, web content grading, blacklist restriction, and other filtering methods have not been able to block illegal information dissemination. At present, network illegal image recognition is mainly divided into the following categories.

(1) Erotic Image Recognition. The identification of online pornographic images mainly includes methods based on skin color feature extraction detection and judgment and methods based on limb judgment. The limb state is determined by extracting the divided skin color information and the connection feature of the human body posture and further determining whether the image transmitted on the network contains pornographic content.

(2) Images Involving State Secrets or Military Secrets. The recognition of secret-related images mainly involves steganography, which hides secret information in image, video, and other carriers for secret transmission. At present, the high-order statistical features based on modeling the complex correlation of image neighborhood become the mainstream features in the field of steganalysis. SRM (Spatial Rich Model), PSRM (Projection Speciation Rich Model), and other models are based on such high-order, high-dimensional features and have achieved good detection results. Steganalysis based on depth learning is a hotspot in

the field of information hiding in order to identify the illegal secret-related images accurately and quickly.

(3) Images Containing Antihuman Content such as Terrorist Violence. The identification of such images is mainly based on image contrast techniques. Image comparison includes techniques such as image feature extraction, high-dimensional spatial feature index establishment, and similarity measure. It is a very worthwhile to study how to quickly compare the massive network images to the illegal target images, so that the recall rate and the precision rate can be taken into account.

3. The Proposed Method

In this section, we will present the notations, as summarized in Table 2 firstly. The concept of deep learning stems from the field of artificial neural networks. Deep learning is deep neural network learning and is a learning structure with multiple hidden layers. In the process of deep learning, the network is trained layer by layer. Each layer of the learning network extracts certain features and information and takes the training result as a deeper input. Finally, the entire network is fine-tuned with a top-down algorithm.

Through deep learning, complex function expressions can be learned, thereby completing the concept of high-level abstraction from the underlying information. It has been widely used in language understanding, target recognition, and speech perception.

Figure 1 shows that the proposed framework IDLH includes three components. The first component is preprocessing layer on the image dataset. The second component is the training self-coding network with a layer-by-layer greedy learning algorithm to obtain the feature expression function of the image. The third is hash layer which retrieves images similar to the query image with compact binary codes and categorizes the query one by the majority semantic label within the hashing bucket.

3.1. Preprocessing. Since the depth learning algorithm used in this paper is an unsupervised learning algorithm, it can automatically learn the deep features of the image from the original pixel information of the image. Therefore, the original pixel value of the whole image can be directly used as input data for the deep learning model. In order to facilitate the training of the network, it is necessary to preprocess the image. Through preprocessing, the image is simply scaled, sample-by-sample mean-value reduction, and whitening is processed to reduce the redundant information in the image and facilitate the deep learning network for training and calculation. Preprocessing can be further grouped into three suboperations.

(1) Normalization. Normalization can prevent neuron output saturation caused by excessive net input absolute value. We use the sigmoid function to do normalization, as shown as follows:

Table 2: Summary of notations.

Symbol	Definition
x_i	the gray value of the image pixels
$\mu^{(i)}$	mean pixels
U	an arbitrary orthogonal matrix and defines in the ZCA whitening
J(W,b)	the quantization loss between the learned binary values and the real values
λ	a weight attenuation parameter
S_i	ith set
a, b	two thresholds
D	dimensionality of data points
M, K	number of candidate images
E	the objective function

$$x_i = \frac{1}{(1 + e^{-x_i})} \tag{1}$$

x_i is the gray value of the image pixels.

(2) Pointwise Mean Reduction. This process mainly is to get rid of the redundant information of the image, the mean value is eliminated for each point of the image, the average brightness of the image is removed, and the DC component of the data is eliminated. Assuming that $x^{(i)} \in R^n$ is the gray value of each pixel of image I, we use formulae (2) and (3) to zero-mean image.

$$\mu^{(i)} = \frac{1}{n}\sum_{j=1}^{n} x_j^{(i)} \tag{2}$$

$$x_j^{(i)} = x_j^{(i)} - \mu^{(i)} \tag{3}$$

(3) Whitening. Whitening is an important pretreatment process; its purpose is to reduce the redundancy of input data, so that the whitened input data has the following properties: (i) low correlation between features; (ii) all features having the same variance, then the formula is as formula (4) .

In formula (4) the rotation matrix of $x_{rot,i}$ is $U^T x_i$. Generally, when x is in interval [-1,1], $\varepsilon \approx 10^{-5}$.

$$x_{ZCAwhite} = U\frac{U^T x_i}{\sqrt{\lambda_i + \varepsilon}} = U\frac{x_{rot,i}}{\sqrt{\lambda_i + \varepsilon}} \tag{4}$$

3.2. Training Stack Sparse Self-Encoding Network. Stack self-coding neural network has strong expressive ability, which mainly benefits from its hierarchical feature representation. Through one level of feature learning, we can learn the hierarchical structure between features. Stack self-encoding neural network is a neural network model composed of multilayer sparse self-encoder, that is, the output of the former self-encoder as the input of the latter self-encoder.

In the training the original input $x^{(k)}$ is used as input to train the first self-encoded neural network. At this point, for each training sample $x^{(k)}$, the output $h_1^{(k)}$ of the hidden

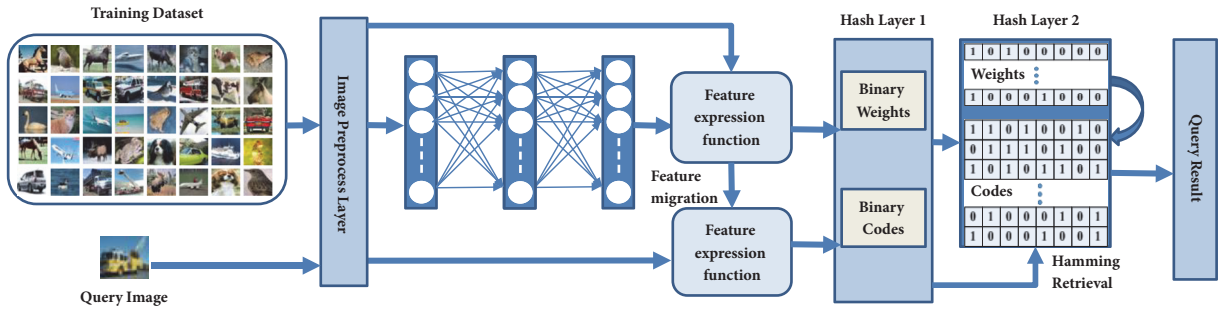


Figure 1: Deep learning-hash retrieval framework. IDLH consists of three main components (preprocessing layer, deep neural network layer, and hash layer). The object of the first layer is simply scaled, sample-by-sample mean-value reduction, and whitening. In the second component, we develop a deep neural network to obtain the feature expression function of the image. And the classifier weights and feature binary codes are simultaneously learned in the last component-hash layer.

layer can be obtained, and the output of the hidden layer can be used as the input of the second self-encoder to continue training the second self-encoder. Then, the output $h_2^{(k)}$ of the second hidden layer of the self-encoder can be obtained. The output $h_1^{(k)}$ of the first hidden layer of the self-encoder is called a first-order feature, and the output $h_2^{(k)}$ of the second hidden layer of the self-encoder is called a second-order feature. In order to classify, the two-order feature $h_2^{(k)}$ can be used as the input of Softmax regression.

Figure 2 shows the flowchart of the proposed method. And there are mainly three processes (supervised pretraining, rough image retrieval, and accurate image retrieval). The object of the first process is to transform the high-dimensional feature vector into a low-dimensional compact two value codes through hash function. In the second procedure, we pick out M candidate images by calculating Hamming distance. In the third process, we calculate the Euclidean distance between the candidate image and the image to be retrieved and accurately extract K images from the M candidate images.

Figure 3 shows a block diagram of a self-encoding neural network. The stacking self-encoding network contains 3 hidden layers (feature layers). The input layer inputs the original data i into the first layer of the feature layer, the output result of the former layer serves as the input of the next layer, and the output of the third layer serves as the feature expression of the image. In our method, it is also used as the input of the hash classifier, and it is possible to use the characteristics of the STD neural network to classify the features.

By using the matrix representation of the binary codes vectors and the output of the 3th layer of the network, we use the gradient descent method to solve the neural network.

For a single sample (x, y), the cost function is as shown in

$$J(W,b;x,y) = \frac{1}{2}\left\|h_{W,b}(x) - y\right\|^2 \tag{5}$$

For datasets containing m samples, the optimization cost function is formulated by the following formula:

$$\begin{aligned}\min_{W,b} J &= \left[\frac{1}{m}\sum_{i=1}^{m} J\left(W,b;x^{(i)},y^{(i)}\right)\right] \\ &\quad + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2 \\ &= \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{W,b}\left(x^{(i)}\right) - y^{(i)}\right\|^2\right)\right] \\ &\quad + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2\end{aligned} \tag{6}$$

The first term $J(W,b)$ represents the mean variance term. The second term aims to prevent the data from overfitting by reducing the magnitude of the weight. λ is a weight attenuation parameter. It is used to balance the relative importance of mean square deviation terms and weight attenuation terms. Our purpose is to minimize the quantization loss $J(W,b)$ between the learned binary values and the real values of an input image according to parameters W and b.

3.3. Hash Algorithm Retrieval. The image retrieval method based on hash algorithm maps the high-dimensional content features of images into Hamming space (binary space) and generates a low-dimensional hash sequence to represent a picture. This method reduces the requirement of computer memory space for image retrieval system, improves the retrieval speed, and better adapts to the requirements of mass image retrieval.

Inspired by [6, 8], we use a set of hash functions to hash data into different buckets. After we do some hash mapping on the original image feature data, we hope that the original two adjacent feature data can be hash into the same bucket with the same bucket number. And then, after hash mapping of all the data in the original feature set, we can get a hash table. These original feature data sets are scattered into hash table buckets, and the data belonging to the same bucket is probably adjacent to the original data. However, there is also

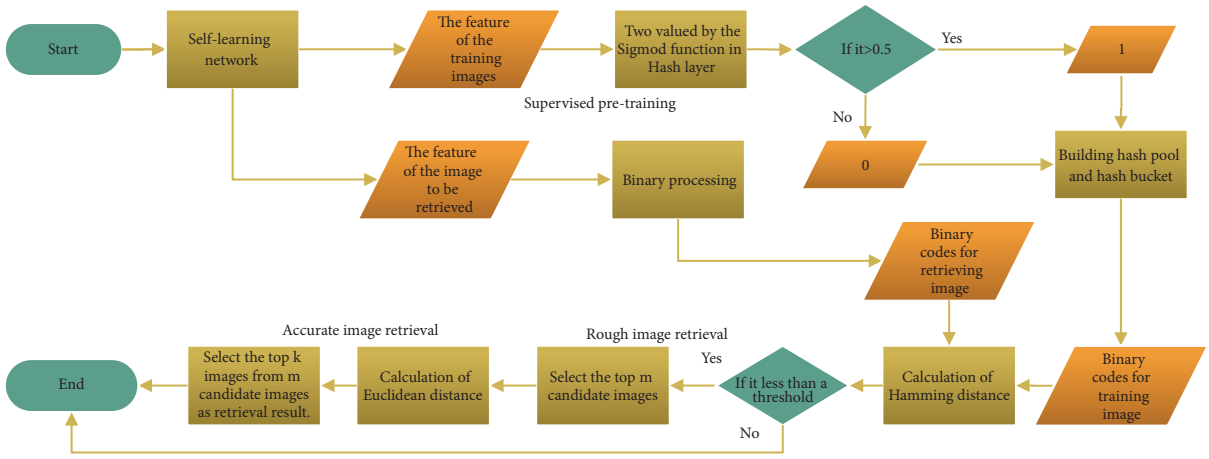


Figure 2: Deep learning-hash retrieval flowchart. IDLH mainly includes three processes (supervised pretraining, rough image retrieval, and accurate image retrieval).

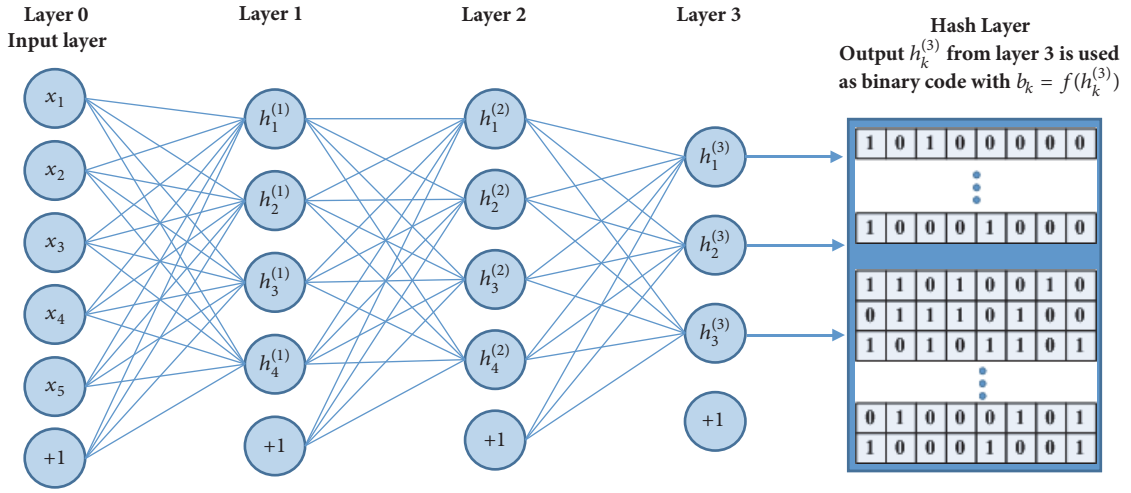


Figure 3: Self-learning network based on stack self-encoding network. The neurons labeled x_i is the input of the neural network and "+ 1" are the offset nodes (intercept entries) of the neural network. The layer 0 is the input layer of neural network, and layer 3 is the output layer of neural network. The middle layers of layer 0 layer to layer 3 are the hidden layers of neural network.

a small probability in events; that is, the nonadjacent data is hash to the same barrel. Set the hash function as the follows:

$$h_k(x) = \operatorname{sgn}\left(w_k^t x + b_k\right) \tag{7}$$

Here w_k is the projection vector and b_k is the corresponding intercept. The code value generated by formula (7) is $\{-1, 1\}$, and we use the following formula to convert it into two value codes:

$$y_k = \frac{1}{2}\left(1 + h_k(x)\right) \tag{8}$$

Given a sample point $\in R^D$, we can compute a K-bit binary code y for x with formula (9). The hash function performs the mapping as $h_k : R^D \longrightarrow B$

$$y = \{h_1(k), h_2(k), \ldots, h_k(k)\} \tag{9}$$

Then, for a given set of hash functions, we can map them to a set of corresponding binary codes by formula (10).

$$Y = H(X) = \{h_1(X), h_2(X), \ldots, h_k(X)\} \tag{10}$$

Here $X = \{x_n\}_{n=1}^N \in R^{D\times N}$ is the feature data matrix with points as columns. Such a binary encoding process can also be viewed as mapping the original data point to a binary valued space, namely, Hamming space.

3.4. Similarity Measure. After obtaining the binary hash code of the image, it is necessary to measure similarity between the retrieved image and the library image in the Hamming space. The smaller the Hamming distance is, the closer distance between the two data is, and the degree of similarity is higher; otherwise, the two data similarity is lower.

$$d_H\left(y_i, y_j\right) = y_i \oplus y_j \tag{11}$$

Table 3: Image library image storage structure.

Hash Sequence ID	Hash Code	Image ID
0	010011101011	Cat1.jpg
1	001110101010	Cat2.jpg
.....	.....	.....
200	101010101001	Cat200.jpg
.....	.....	.....

Here $\oplus$ is an XOR operation. The two sets of y_i and y_j represent the hash code of the search image feature and the image library is mapped through the hash function. The new image features learned by the stack self-encoding network are generated by the hash function. The storage structure of the image feature vectors is shown in Table 3.

As can be seen from Table 3, the hash code of the image is related to the image *ID* and the image name one by one. In the process of searching, the image feature vector is obtained through deep learning by a hash function, the original data is mapped into a new data space, and a corresponding hash code is obtained. The hash code is used to calculate the Hamming distance in the Hamming space as a measure of similarity between images. Finally, the storage structure of the image feature vector is used to find the corresponding image *ID* of the hash code, and the output retrieval result is output to the user.

3.5. Image Secondary Search. In the first-level search phase, the features learned from the deep learning network are mapped into the Hamming space using the hash function. In the similarity measurement phase, the traditional Euclidean distance is abandoned. Measure the similarity between images by comparing the Hamming distance between the image features of the query image and the image of the library image. In order to further improve the accuracy of retrieval without affecting the real-time performance, we can retrieve the image by the second level retrieval. These steps are described in detail as follows: After one level retrieval, we choose the K images with the most similarity in the first-level retrieval result and then calculate the Euclidean distance between the original feature vector of the K images and the original feature vector of the query image. The results obtained as the similarity measure of the images and output the retrieval result that has been ranked from the high and low with the similarity distance.

Although the Hash algorithm maps the high-dimensional feature vectors of the image into a hash-coded form, the problem of "dimensional disasters" is solved, and the retrieval efficiency is greatly accelerated. However, when the similarity comparison is performed, the Hamming distances of the image features are simply compared using the results of the primary search, and occasionally undesirable results may still appear on the search results. If we want to increase the accuracy of the search, we must increase the hash code length. However, excessively long codes will increase the amount of calculations, increase the memory burden, and reduce the real-time nature of retrieval, failing to achieve the goal of reducing the size of data. In order to solve this problem, keep the retrieval efficiency, and further improve the retrieval accuracy, we propose a search strategy for secondary retrieval, the specific steps of which are as follows.

Step 1. Through the first-level search in the Hamming space, the similarity degree of the images is sorted, and the top K sorting images are selected.

Step 2. For the k images in Step 1, calculate the Euclidean distance one by one from its original image feature vector to the image feature vector of the query image.

Step 3. The Euclidean distance calculated in Step 2 is sorted. The smaller the calculated value is, the higher similarity between images is, and the similarity is sorted from high to low and output as the final search result.

In the second search, it is necessary to pay attention to the selection of the K value, although the larger the K value is, the better the search effect is, but accordingly, the longer the time is consumed. Therefore, it is necessary to combine various factors to select the appropriate K value.

4. Experimental and Performance Analysis

In this section, we thoroughly compare the proposed approach with the improved deep learning hash retrieval methods on several benchmark datasets. Through a series of experiments, the effectiveness and feasibility of the proposed algorithm are verified.

4.1. Database. Two mostly used databases in the recent deep learning hash works are taken into our evaluation. The two image libraries are derived from the CIFAR-10[11] core experimental image library dataset and the Caltech 256 image library dataset.

CIFAR-10 dataset contains 10 object categories and each class consists of 6,000 images, resulting in a total of 60,000 images. The dataset is split into training and test sets which are averagely divided into 10 object classes. The Caltech 256 image library dataset contains 29,780 color images which are grouped intro 256 classes.

First, test the CIFAR-10 image dataset. There are a total of 50,000 training samples which are used for training on the deep learning network. The remaining 10,000 images are used as test samples. And then we randomly select 50 images from database as the query images. For the Hidden Image Retrieval algorithm based on deep learning mentioned in this paper, the image pixel data is directly used as input, while for other algorithms, the 512-dimensional GIST feature is used as the feature expression of the image. Note quantization all images into $32*32$ sizes before experiment.

For the Caltech 256 image, a total of 256 classes are included, and each class contains at least 70 images. Therefore, 70 images of each class, a total of 17,920 images, are randomly selected and are used as training images. The remaining images are used as test samples. In addition, all of the images' size is set to $64*64$ again when training.

4.2. Evaluation Metrics. We measure the performance of compared methods using Precision-recall and Average-Retrieval Precision (ARP) curves. Precision is the ratio of the correct number of images m in the search result to the number k of all returned images. The formula is as follows:

$$precision = \frac{m}{k} \times 100\% \tag{12}$$

Recall is the ratio of the correct number of images m in the search results to the number g of images in the image library. The formula is as follows:

$$recall = \frac{m}{q} \times 100\% \tag{13}$$

Assume that the search result of the query image i is B_i and A_i means that the category is the same between the query image and the return image, then the accuracy rate for the image query result $P(i)$ can be defined by the following formula:

$$P(i) = \frac{|A(i) \cap B(i)|}{|B(i)|} \tag{14}$$

Average-Retrieval Precision (ARP): the average value of all the images in the same class as the retrieval rate obtained from the retrieval image is defined as follows:

$$ARP(ID_m) = \frac{1}{N} \sum_{id(i)=ID_m} P(i) \tag{15}$$

Here ID_m is the category index number of the image, m is the category index, N is the number of images whose category is ID_m, and $id(i)$ is the category index number of the query image.

4.3. Performance Analysis. In the proposed algorithm, IDLH, the length of the hash sequence and the depth of the hidden layer in the deep learning network are two key parameters. When the hash sequence length is small, different feature vectors can easily be mapped into the same hash sequence, so the retrieval accuracy is low. However, if the hash sequence is too long, a large storage space is required and a long time is consumed, which reduces the real-time performance. For the number of hidden layers, the number of layers in the hidden layer is too small, which is not conducive to learning strong image features. However, if the depth of the hidden layer is too large, the difficulty of training is increased. In order to verify the effectiveness and feasibility of our algorithm, we conducted the following experiments.

(1) Results on CIFAR-10 Dataset. Figure 4(a) shows the search results of the Average-Retrieval Precision using our proposed algorithm, IDLH, compared with the LSH algorithm [3] and other three deep learning algorithms, the DH algorithm [21], the DeepBit algorithm [40], and the UH-BDNN algorithm [41] on CIFAR-10 dataset with 8, 16, 32, 48, 64, 96, 128, and 256 bits. Figure 4(b) shows the Precision-recall curve under 48-bit encoding. It can be seen that the algorithm has a higher precision than the other hashing algorithms with the same recall rate. However, the advantage is not obvious, and the average accuracy is slightly higher than other hash algorithms.

In order to overcome the above defects, we use deep learning to perform hash mapping on image features, perform hash encoding of different bits on the same feature, calculate the Precision-recall of the search results under the condition of different coded bits, and determine the impact of the encoding length on the retrieval results.

As Figure 5 shows, with the increase in the number of coded bits, the Precision-recall is continuously increasing. With the increase in the number of coded bits, the image is better expressed. However, after the number of coded bits reaches 64, even if the number of coded bits increases, the average accuracy rate increases relatively slowly. Because the information of the tiny image is relatively simple, when the number of encoded bits reaches 64 bits, a relatively good image expression has been obtained, and the performance of the algorithm has basically stabilized. At this time, despite increase in the number of encoding bits, it is not very helpful to improve the accuracy rate.

In addition, we want to test the influence of the number of hidden layers in the deep learning network on the retrieval result by changing the number of hidden layers.

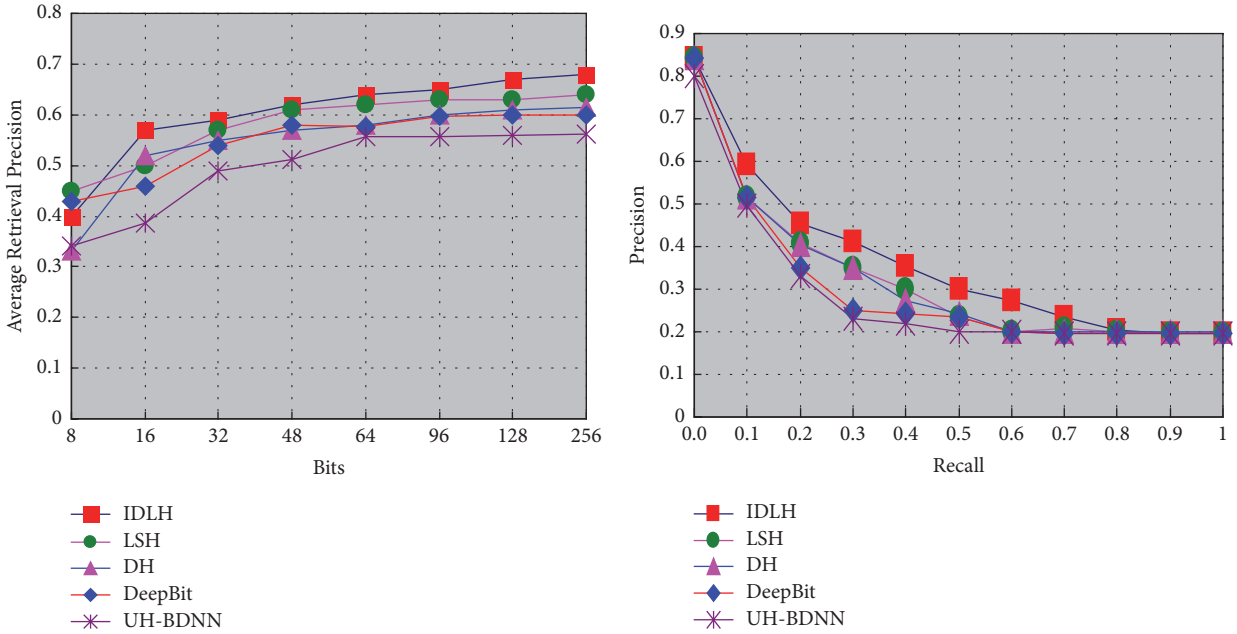
Figure 6 shows the effect of deep learning networks on experimental results in the case of different hidden layer numbers.

It can be seen that deeper networks do not have much improvement in performance, which is different from the expectation that more hidden layers will help learn stronger image features. Since the image library data used in the experiment is a tiny image library, relatively good image characteristics can be learned using a deep learning network with fewer layers. However, if the image library is replaced with a more colorful image, the deep neural network can acquire more detailed image features, and the deepening of the learning network will greatly help the study of image features.

(2) Results on Caltech 256 Image Data Set. Figure 7(a) shows the results of the Average-Retrieval Precision results when the number of coded bits is different. Compared with the black and white image library, the proposed algorithm embodies the advantage of image feature learning and leads the Average-Retrieval Precision to other hash retrieval algorithms. In Figure 7(b), we can also see that the algorithm proposed in this paper has a higher Precision-recall than other algorithms under the same recall rate, and it has better search performance.

As shown in Figure 8, as the number of coded bits increases, the precision rate increases with the same recall rate. This feature of the color image library is more pronounced than the black and white image library. Because the color image contains more information, more coding is needed to express it, and the increase of encoding helps to learn the features of the image. The experimental results also show that the deep learning network has learned more excellent image features.

In Figure 9, the precision rate is significantly improved by the increase in the number of hidden layers in the



(a) Average-Retrieval Precision

(b) Precision-recall at 48 bits

Figure 4: Five kinds of algorithm retrieval performance comparison on CIFAR-10 dataset.

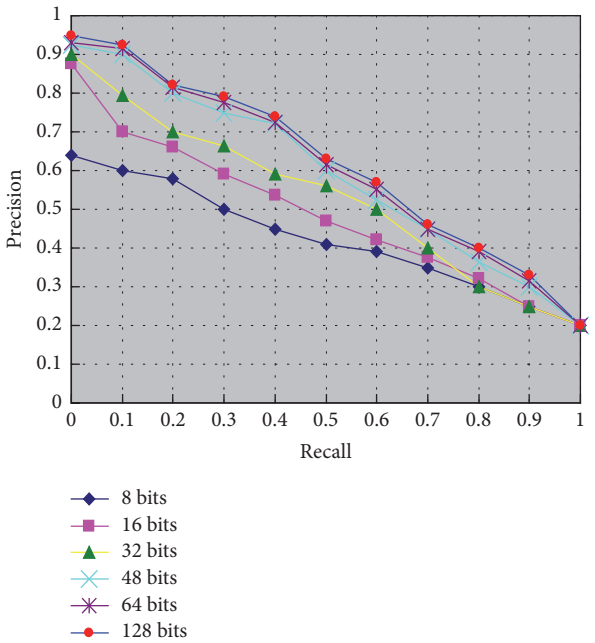


Figure 5: Precision-recall curves with lengths.

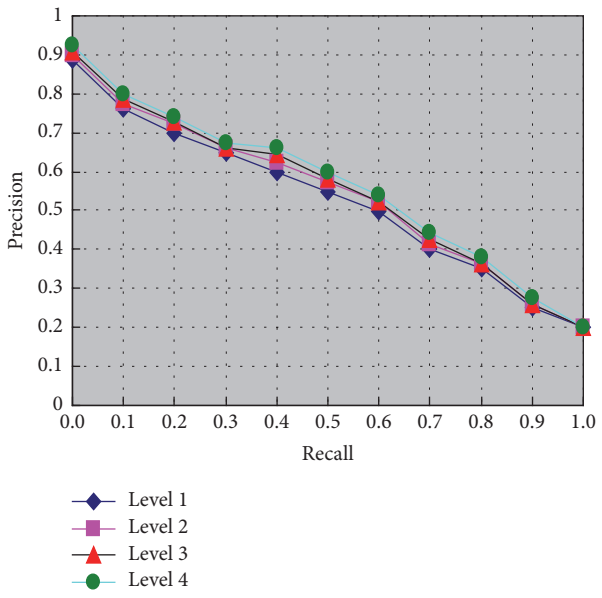
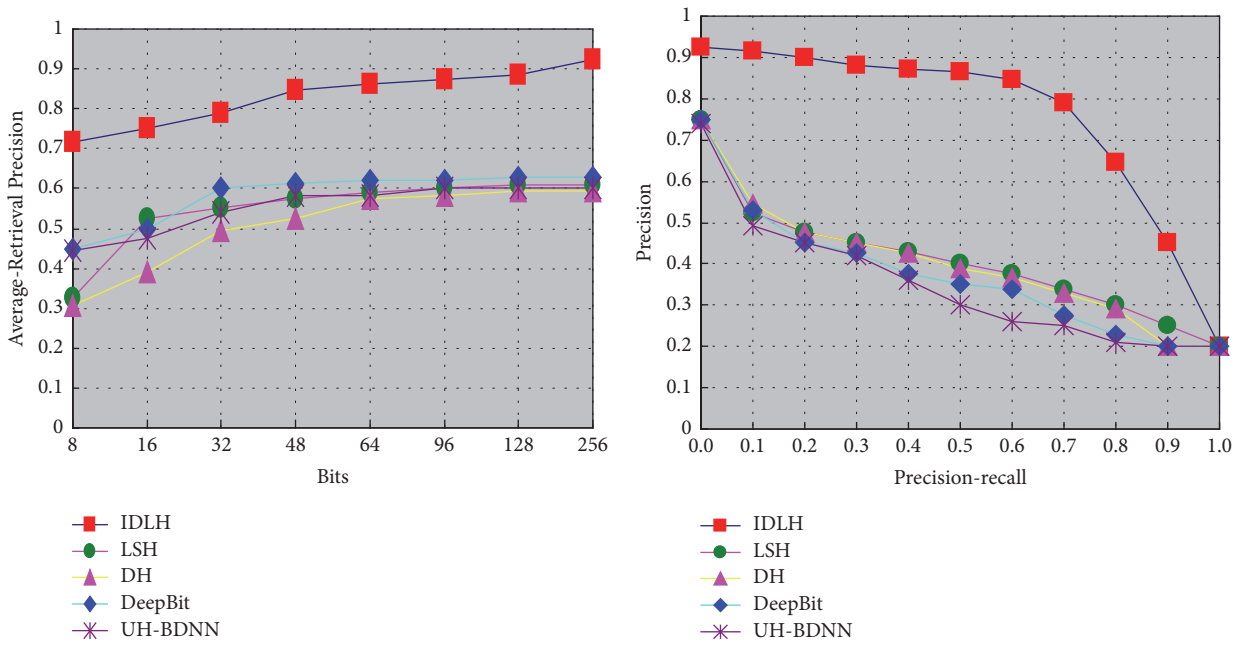


Figure 6: Precision-recall curves with different code different hidden layers.

color image library Caltech 256. This is because the information contained in a more colorful image is more complex. Adding a hidden layer can learn more details of the image and help improve the accuracy of the search.

Next, we tested the performance of secondary image retrieval. The value of k in the secondary search is 20, and the experimental results are shown in Figure 10. As can be seen from the results, secondary retrieval can effectively improve the retrieval accuracy when the number of coded bits is small. However, with the increase in the number of encoding bits, the results of the secondary search and the accuracy of the primary search are not much different. This is because the shorter the hash sequence is, the easier the feature vectors with different original features are mapped to the same hash code. In order to make up for the errors



(a) Average-Retrieval Precision

(b) Precision-recall at 48 bits

Figure 7: Five kinds of algorithm retrieval performance comparison on Caltech 256 set.

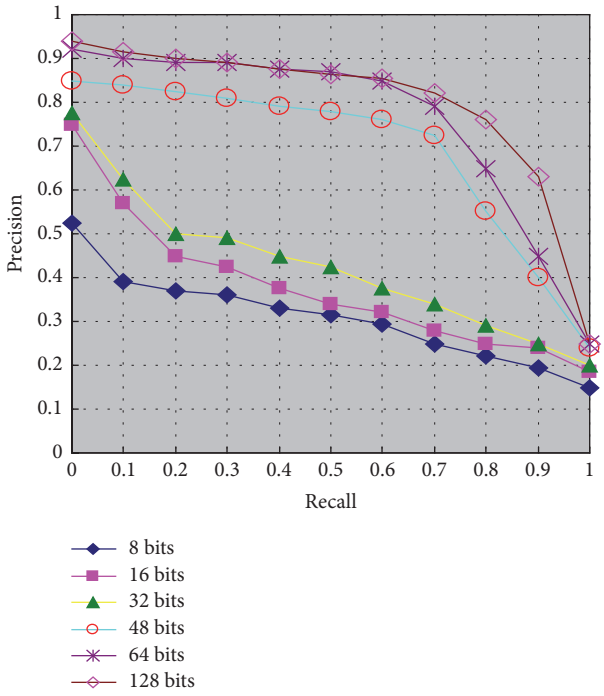


Figure 8: Precision-recall curves with different code lengths.

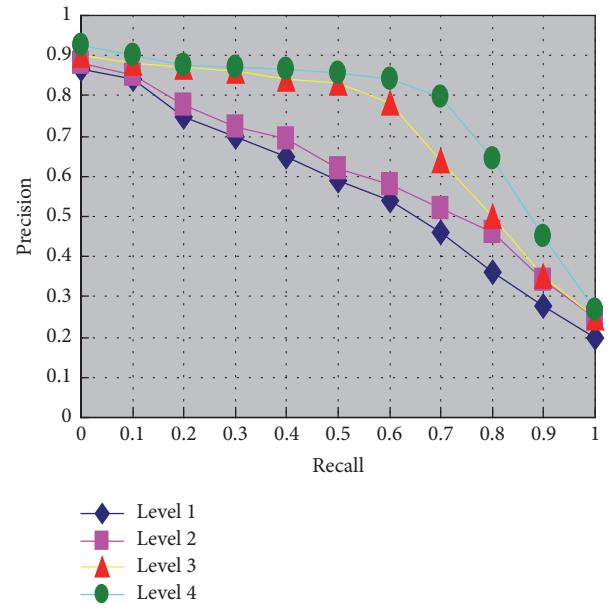


Figure 9: Precision-recall curves with different hidden layers.

caused by the short hash code, it is necessary to perform secondary search. When the number of encoding bits is small, a secondary retrieval method is used in IDLH, and the search accuracy rate can be improved at the expense of a small search speed.

5. Conclusion

With the rapid development of data storage and digital process, more and more digital information is transformed and transmitted over the Internet day by day, which brings people a series of security problems as well as convenience [42]. The researches on digital image security, that is, image encryption, image data hiding, and image authentication, become more important than ever [38, 39]. The most essential problem of

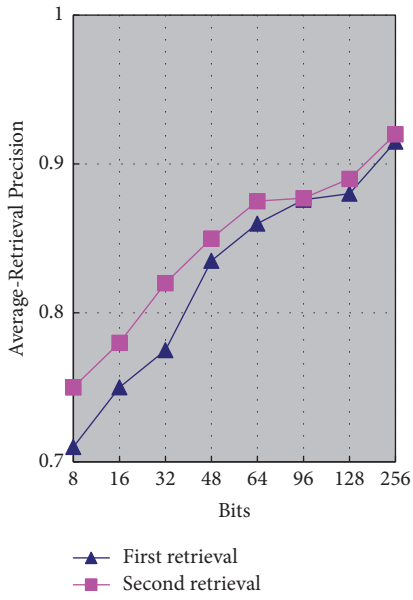


Figure 10: Average-Retrieval Precision with first and second retrieval.

image recognition is to extract robust features. The quality of feature extraction is directly related to the effect of recognition, so most of the previous work on image recognition is spent on artificial design features [43]. In recent years, the emergence of deep learning technology has changed the status of artificial design classification characteristics. Deep learning technology simulates the mechanism of human visual system information classification processing, from the most primitive image pixels to lower edge features, then to the target components that are combined on the edge, and finally to the whole target, depth learning can be combined by layer by layer. The high-level feature is the combination of low level features. From low level to high level, features are more and more abstract and show semantics more and more. From the underlying features to the combination of high-level features, it is the depth of learning that is done by itself. It does not require manual intervention. Compared with the characteristics of the artificial design, this combination of features can be closer to the semantic expression.

In terms of illegal image retrieval, the traditional recognition method should establish a recognition model for each type of recognition task. In the actual application, a recognition model needs a recognition server. If there are many identification tasks, the cost is too high. We used the deep neural network to recognize the illegal image, it only needs to collect the samples of every kind of illegal image and participate in the training of the deep neural network. Finally, a multiclassification recognition model is trained. When classifying unknown samples, deep neural network accounting calculates the probability that the image belongs to each class.

We all know that, in the image detection process, the accuracy and recall rate are mutually influential. Ideally, both must be high, but in general, the accuracy is high and the recall rate is low; the recall rate is high and the accuracy is low. For image retrieval, we need to improve the accuracy under the condition of guaranteeing the recall rate. For image disease surveillance and anti-illegal images, we need to enhance the recall under the condition of ensuring accuracy. Therefore, in different application scenarios, in order to achieve a balance between accuracy and recall, perhaps some game theory (such as Nash Equilibrium [44, 45]) and penalty function [46–48] can provide related optimization solutions.

In this paper, we proposed an improved deep-learning-hashing approach, IDLH, which optimized over two major image retrieval process.

(a) In the feature extraction process, the self-encoded network of the look-ahead type is trained by using unlabeled image data, and the expression of robust image features is learned. This unlabeled learning method does not require image library labeling and reduces the requirements for the image library. At the same time, it also takes advantage of the deep learning network's strong learning ability and obtains better image feature expression than ordinary algorithms.

(b) On the index structure, a secondary search is proposed, which further increases the accuracy of the search, at the expense of very little retrieval time.

Through experiments, the algorithm proposed in this paper is compared with other classic hashing algorithms on multiple evaluation indicators. Firstly, we tested the learning networks of different code lengths and depths in order to test their effect on the retrieval system and then tested the performance of the secondary search. Through the above-mentioned series of experiments for different parameters, the effectiveness of the improved deep learning hash retrieval algorithm proposed in this paper is verified, and through the experimental data, the good retrieval results are proved. In addition, the proposed deep hashing training strategy can also be potentially applied to other hashing problems involving data similarity computation.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was funded by the National Natural Science Foundation of China (Grants nos. 61206138 and 61373016).

References

[1] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, article 5, 2008.

[2] G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*, MIT Press, Cambridge, MA, USA, 2006.

[3] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *25th Int. Conf*, pp. 518–529, 1999.

[4] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[5] G.-L. Tian, M. Wang, and L. Song, "Variable selection in the high-dimensional continuous generalized linear model with current status data," *Journal of Applied Statistics*, vol. 41, no. 3, pp. 467–483, 2014.

[6] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the 20th Annual Symposium on Computational Geometry (SCG '04)*, pp. 253–262, ACM, June 2004.

[7] B. Kulis, P. Jain, and K. Grauman, "Fast similarity search for learned metrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2143–2157, 2009.

[8] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems, NIPS 2009*, pp. 1509–1517, Canada, December 2009.

[9] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2616–2624, 2017.

[10] M. A. Carreira-Perpiñán and R. Raziperchikolaei, "Hashing with binary autoencoders," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 557–566, USA, June 2015.

[11] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2393–2406, 2012.

[12] Y. Gong, S. Lazebnik, and A. Gordo, "Iterative quantization: a Procrustean approach to learning binary codes for large-scale image retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 2916–2929, June 2011.

[13] W. Kong and W. J. Li, "Isotropic hashing," *NIPS*, vol. 25, 2012.

[14] M. Norouzi and D. J. Fleet, "Minimal loss hashing for compact binary codes," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pp. 353–360, USA, July 2011.

[15] J. Wang, W. Liu, A. X. Sun, and Y.-G. Jiang, "Learning hash codes with listwise supervision," in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision, ICCV 2013*, pp. 3032–3039, Australia, December 2013.

[16] G. Lin, C. Shen, Q. Shi, A. Van Den Hengel, and D. Suter, "Fast supervised hashing with decision trees for high-dimensional data," in *Proceedings of 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*', pp. 1971–1978, USA, 2014.

[17] Y. Gong, S. Kumar, H. A. Rowley, and S. Lazebnik, "Learning binary codes for high-dimensional data using bilinear projections," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 484–491, USA, June 2013.

[18] W. Liu, J. Wang, Y. Mu, and S. Kumar, "Compact hyperplane hashing with bilinear functions," in *The 29th International Conference on Machine Learning (ICML12)*, pp. 467–474, 2012.

[19] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS '08)*, pp. 1753–1760, Vancouver, Canada, December 2008.

[20] W. Liu, J. Wang, S. Kumar, and S. F. Chang, "Hashing with graphs," in *The 28th international conference on machine learning (ICML11)*, 2011.

[21] F. Shen, X. Zhou, Y. Yang, J. Song, H. T. Shen, and D. Tao, "A fast optimization method for general binary code learning," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5610–5621, 2016.

[22] F. Shen, W. Liu, S. Zhang, Y. Yang, and H. T. Shen, "Learning binary codes for maximum inner product search," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 4148–4156, Chile, December 2015.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.

[24] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *The American Association for the Advancement of Science: Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[25] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, 2008.

[26] R. Salakhutdinov and G. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," *Journal of Machine Learning Research*, vol. 2, pp. 412–419, 2007.

[27] V. E. Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 2475–2483, USA, June 2015.

[28] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.

[29] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2074–2081, Providence, RI, USA, June 2012.

[30] J. Masci, A. Bronstein, M. Bronstein, and P. Sprechmann, "Sparse similarity-preserving hashing," in *Int. Conf. Learn. Represent*, pp. 1–13, 2014.

[31] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1092–1104, 2012.

[32] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 1556–1564, June 2015.

[33] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–11.

[34] J. He, J. Feng, X. Liu et al., "Mobile product search with Bag of Hash Bits and boundary reranking," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pp. 3005–3012, USA, June 2012.

[35] F. Shen, Y. Mu, Y. Yang et al., "Classification by retrieval: Binarizing data and classifiers," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017*, pp. 595–604, Japan, August 2017.

[36] P. Li, S. Zhao, and R. Zhang, "A cluster analysis selection strategy for supersaturated designs," *Computational Statistics & Data Analysis*, vol. 54, no. 6, pp. 1605–1612, 2010.

[37] A. Pradeep, S. Mridula, and P. Mohanan, "High security identity tags using spiral resonators," *Cmc-Computers Materials & Continua*, vol. 52, no. 3, pp. 187–196, 2016.

[38] Y. Cao, Z. Zhou, X. Sun, and C. Gao, "Coverless information hiding based on the molecular structure images of material," *Computers, Materials and Continua*, vol. 54, no. 2, pp. 197–207, 2018.

[39] Y. Liu, H. Peng, and J. Wang, "Verifiable diversity ranking search over encrypted outsourced data," *Cmc-Computers Materials & Continua*, vol. 55, no. 1, pp. 037–057, 2018.

[40] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 1183–1192, USA, July 2016.

[41] T. Do, A. Doan, and N. Cheung, "Learning to Hash with Binary Deep Neural Network," in *Computer Vision – ECCV 2016*, vol. 9909 of *Lecture Notes in Computer Science*, pp. 219–234, Springer International Publishing, Cham, 2016.

[42] Rui Zhang, Di Xiao, and Yanting Chang, "A Novel Image Authentication with Tamper Localization and Self-Recovery in Encrypted Domain Based on Compressive Sensing," *Security and Communication Networks*, vol. 2018, Article ID 1591206, 15 pages, 2018.

[43] Xia ShuangKui and Jianbin Wu, "A Modification-Free Steganography Method Based on Image Information Entropy," *Security and Communication Networks*, vol. 2018, Article ID 6256872, 8 pages, 2018.

[44] J. Zhang, B. Qu, and N. Xiu, "Some projection-like methods for the generalized Nash equilibria," *Computational Optimization and Applications*, vol. 45, no. 1, pp. 89–109, 2010.

[45] Biao Qu and Jing Zhao, "Methods for Solving Generalized Nash Equilibrium," *Journal of Applied Mathematics*, vol. 2013, Article ID 762165, 6 pages, 2013.

[46] C. Wang, C. Ma, and J. Zhou, "A new class of exact penalty functions and penalty algorithms," *Journal of Global Optimization*, vol. 58, no. 1, pp. 51–73, 2014.

[47] Y. Wang, X. Sun, and F. Meng, "On the conditional and partial trade credit policy with capital constraints: A Stackelberg Model," *Applied Mathematical Modelling*, vol. 40, no. 1, pp. 1–18, 2016.

[48] S. Lian and Y. Duan, "Smoothing of the lower-order exact penalty function for inequality constrained optimization," *Journal of Inequalities and Applications*, Paper No. 185, 12 pages, 2016.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 2373545, 10 pages
https://doi.org/10.1155/2018/2373545

Research Article

Privacy-Preserving Sorting Algorithms Based on Logistic Map for Clouds

Hua Dai,[1,2] **Hui Ren**,[1,3] **Zhiye Chen**,[4] **Geng Yang**,[1,2] **and Xun Yi**[5]

[1]*Nanjing University of Post and Telecommunication, Nanjing 210023, China*
[2]*Jiangsu Security and Intelligent Processing Lab of Big Data, Nanjing 210023, China*
[3]*China Information Consulting and Designing Institute Co., Ltd., Nanjing 210019, China*
[4]*TIZA Information Industry Corporation Inc., Nanjing 210019, China*
[5]*Royal Melbourne Institute of Technology University, Melbourne 3001, Australia*

Correspondence should be addressed to Hua Dai; daihua@njupt.edu.cn

Received 4 June 2018; Accepted 7 August 2018; Published 4 September 2018

Academic Editor: Zhaoqing Pan

Outsourcing data in clouds is adopted by more and more companies and individuals due to the profits from data sharing and parallel, elastic, and on-demand computing. However, it forces data owners to lose control of their own data, which causes privacy-preserving problems on sensitive data. Sorting is a common operation in many areas, such as machine learning, service recommendation, and data query. It is a challenge to implement privacy-preserving sorting over encrypted data without leaking privacy of sensitive data. In this paper, we propose privacy-preserving sorting algorithms which are on the basis of the logistic map. Secure comparable codes are constructed by logistic map functions, which can be utilized to compare the corresponding encrypted data items even without knowing their plaintext values. Data owners firstly encrypt their data and generate the corresponding comparable codes and then outsource them to clouds. Cloud servers are capable of sorting the outsourced encrypted data in accordance with their corresponding comparable codes by the proposed privacy-preserving sorting algorithms. Security analysis and experimental results show that the proposed algorithms can protect data privacy, while providing efficient sorting on encrypted data.

1. Introduction

With the profits from data sharing and parallel, elastic, and on-demand computing, clouds are becoming more and more popular with companies and individuals. Many kinds of services are provided by cloud service providers (CSP), such as Amazon EC2 and Alibaba Cloud. As one of the most important technologies, machine learning is very useful and wildly adopted in many areas, such as prediction [1, 2] and multimedia data processing [3, 4]. And it usually utilizes huge data volume, such as wireless multimedia data and human health data, to build intelligent models and systems for practical applications. Due to the need of large and elastic scale of storage and computing resources, those huge volume data are usually processed in clouds [5–7]. Data owner (DO) outsources their data in the cloud server (CS) for on-demand services which enhance the efficiency of complex computation such as machine learning and save the hardware/software cost.

However, in the cloud environment, DO lose direct control of their own data placed in remote CS, which may cause concerns about their outsourced data being illegally acquired or abused by CSPs, especially for sensitive data, such as national defence data and human health data. Although many CSPs claim that they deployed several safety measures in CS, such as access control, firewalls, or intrusion detection, doubts about the privacy of outsourced data obstruct the promotion and application of cloud computing. How to preserve the security and privacy of DO's outsourced data while CS providing reliable and efficient computing services has become a hot issue [8–10].

Data encryption is a common technique to protect the privacy of outsourced data on clouds, such as sensitive wireless multimedia data and human health data. Sorting is

one of the basic methods in practical applications, such as machine learning, service recommending, and data query. However, applying to sort over encrypt data on clouds is a challenge without leaking private information. The existing privacy-preserving sorting algorithms based on order-preserved encryption (OPE) [11–14] have security problems [15]. In addition, privacy-preserving sorting algorithms based on fully homomorphic encryption (FHE) [16–19] are too slow because of the complexity of FHE. It is significant to research the efficient privacy-preserving sorting algorithms for clouds.

In this paper, we assume that the honest-but-curious threat model [20] is adopted where CS strictly abides by established protocols but has the curiosity to snoop on DO's private data. On the basis of the threat model, we propose privacy-preserving sorting algorithms based on the logistic map. The main contributions of this paper are as follows. Firstly, by introducing the logistic map, we propose a secure comparison model which can be utilized to compare data without knowing their real values. Secondly, we give a data preprocessing algorithm. Data owners preprocess their private data by a symmetric encryption and logistic map. The encrypted data and corresponding comparable codes are generated and then outsourced to clouds, where the former is to protect data privacy, while the latter is to support secure comparisons. Finally, on the basis of secure comparison model, we propose privacy-preserving sorting algorithms for clouds. Also, security analysis and performance experiment are given, where results show that the proposed algorithms can protect data privacy from curious cloud administrators while providing efficient sorting on encrypted data.

The paper is organized as follows. Section 2 describes the related work. Section 3 gives the problem descriptions. Section 4 gives notations and necessary preliminaries. In Section 5, we firstly present secure comparison model on the basis of the logistic map, and then the data preprocessing algorithm and privacy-preserving sorting algorithms are given. Section 6 analyzes the security of our schemes. Section 7 gives evaluations on correctness, correlation coefficient, and performance of our proposed schemes.

2. Related Works

There are two kinds of methods achieving encrypted data sorting on clouds, one is sorting algorithms based on order-preserving encryption (OPE) [11–14], and the other is sorting algorithms based on fully homomorphic encryption [16–19].

Agrawal et al. [11] originally proposed an OPE method which is a deterministic encryption scheme whose encryption function preserves numerical ordering of the plaintext. Due to the unachievable of the indistinguishability against chosen-plaintext attack (IND-CPA) in [11], Boldyreva et al. proposed an efficient OPE scheme [12] which is based on a natural relation between a random order-preserving function and the hypergeometric probability distribution. Jaiman et al. [13] proposed an OPE algorithm by introducing shuffling, impurity insertion, and randomness in order-preserving functions. Liu et al. [14] propose a new OPE model which uses message space expansion and nonlinear space split to hide data distribution and frequency. Any proposed OPE is clearly suitable for application of privacy-preserving sorting in clouds if the data security is ensured. However, OPE is vulnerable to ciphertext-only attack [15], especially when encrypted data are massive. Therefore, those sorting algorithms based on OPE have potential security risks.

Gentry et al. [21] proposed the fully homomorphic encryption (FHE) which is a special encryption algorithm which allows computation (such as addition and multiplication) on the ciphertext. Melchor et al. [16] give an idea about sorting encrypted data by FHE. Chatterjee et al. [17] propose the sorting algorithm over encrypted data on the basis of FHE. They tried to get higher sorting efficiency by reducing costs of reencryption. Afterwards, they applied the algorithm to clouds [18, 19]. The volume of encrypted data generated by FHE is very large, due to the inclusion of big floating-point numbers which take the place of numerous storage space. Thus, the calculation of comparison for sorting based on FHE is very complex and its time efficiency is also very slow. Since the fully homomorphic encryption based sorting requires CS to reencrypt data frequently, it is not suitable for storing and managing big data on clouds.

To support efficiency and privacy in sorting algorithms for cloud environments, we propose logistic map based privacy-preserving sorting algorithms in this paper, the abstract of which has been shown in [22].

3. Problem Description

The model of the privacy-preserving sorting for clouds, proposed in this paper, is similar to recent works [18, 19]. It mainly consists of two entities, data owner (DO) and cloud server (CS). The interactions between DO and CS are introduced as follows: firstly, DO encrypts its sensitive data and generates corresponding codes which are used for privacy-preserving sorting. Then DO outsources the encrypted data and codes to CS. Secondly, CS stores the data uploaded by DO and performs data sorting over the received data. Any proved secure symmetric encryption could be adopted, such as DES and AES. If authorized users want to access DO's sensitive data, they can get the encrypted data from CS and perform decryption to obtain the plaintext data by using the shared key with DO.

In this paper, we assume that CS provides services following the curious-but-honest threat model [20]. CS is assumed to strictly follow the established protocols, but it attempts to snoop on DO's private data. There are two kinds of attacks: (1) CS has already known DO's preprocessing algorithms but does not know its initial parameters, and it tries to use exhaustive attacks against to encrypted data for plaintext information; (2) because of the massive quantity of outsourced encrypted data, statistical attacks are common methods for CS to analyze the distribution of ciphertext and speculate on the relationship between the ciphertext and plaintext to obtain plaintext information.

We focus on privacy-preserving sorting for clouds, and the key issues of this paper are introduced as follows: (1) privacy protection on outsourced data: DO preprocesses its plaintext data to keep it in confidential. Thus, CS cannot obtain DO's plaintext information via the outsourced data; (2)

Table 1: Notation descriptions.

Notations	Descriptions
μ	The bifurcation parameter of the logistic map function.
n	The iteration number of the logistic map function.
t	The constraint factor where $0 < t < x/(2 \cdot \mu^n)$
$L(t/x, n)$	The logistic mapping function.
d_i	A plaintext data item of DO.
k	The key of a symmetric encryption which is only owned by DO.
$Enc(x, k)$	The encryption function where x is the input plaintext data and k is a key.
g_i	A secure data pair after data preprocessing on d_i, $g_i = (e, c)$, where $g_i.e = Enc(d_i, k)$ and $g_i.c = L(t/x, n)$ are the corresponding encrypted data and logistic mapping codes of d_i.

privacy-preserving sorting on encrypted data: if a symmetric encryption algorithm is adopted, the privacy is guaranteed, but the encrypted data is hard for data sorting. Therefore, privacy-preserving sorting algorithms have to support sorting on encrypted data even without knowing the values.

4. Preliminaries and Notations

4.1. Preliminaries. Chaos theory [23, 24] originated in the 1960s, which has been widely adopted in medicine, astrophysics, image encryption, and hydromechanics. The basic characteristic of chaotic motion is extremely sensitive to the initial value. The difference between two chaotic motions with different initial values will become larger and larger over time. Therefore, on the basis of any given initial conditions, the chaotic motion is unpredictable.

Logistic map [25–27] is one of the important and practical chaotic motions and has been widely used in data encryption [28–32]. The equation of the logistic map is shown as

$$L(x, n) = \begin{cases} \mu \cdot L(x, n-1) \cdot (1 - L(x, n-1)) & n > 1 \\ x & n = 1, \end{cases} \quad (1)$$

where $x \in [0, 1]$, $\mu \in [0, 4]$ is called bifurcation parameter, and n is the iteration number. Studies [25–27] show that the sequence generated from (1) is chaotic if $\mu \in (3.5699456, 4]$. The output of such logistic map is extremely sensitive to the initial parameters. Any minor changes of initial parameters will lead to a tremendous difference of outputs. Therefore, the sequences generated by the logistic map are unpredictable.

4.2. Notations. The notations used in this paper are described as shown in Table 1.

5. Privacy-Preserving Sorting Algorithms

5.1. Secure Comparison Model Based on Logistic Map. The chaos characteristic of the logistic map can be used to compare data values secretly under certain conditions. Current work such as [28–32] mainly focuses on data encryption with logistic map algorithms, but there is no literature discussing the secure comparison of encrypted data with the logistic map.

If we use the logistic map for data comparison directly, then we may get wrong comparison results during sorting. But if proper constraint factors are introduced in the logistic map, we will get correct compare results invariably. The main idea of the proposed secure comparison model based on the logistic map is briefly given by three lemmas as follows.

Lemma 1. *For any given data x, where $x \geq 1$, t/x is settled as the initial value for the logistic map function, i.e., (1), where t is the constraint factor, where $0 < t < x/(2 \cdot \mu^n)$ and $\mu \in (3.5699456, 4]$. Then we have*

$$0 < L\left(\frac{t}{x}, n\right) < \frac{1}{2}. \quad (2)$$

Proof. We use mathematical inductions to prove Lemma 1 as follows.

(1) When $n = 1$, according to (1), we have

$$L\left(\frac{t}{x}, 1\right) = \frac{t}{x}. \quad (3)$$

According to the given assumptions $0 < t < x/(2 \cdot \mu)$ and $\mu \in (3.5699456, 4]$, we have $0 < L(t/x, 1) < 1/(2 \cdot \mu)$, and then $0 < L(t/x, 1) < 1/2$ is deduced.

(2) When $n = 2$, according to (1), we have

$$L\left(\frac{t}{x}, 2\right) = \mu \cdot L\left(\frac{t}{x}, 1\right) \cdot \left(1 - L\left(\frac{t}{x}, 1\right)\right). \quad (4)$$

According to the conclusion of (1), we have $1/2 < 1\text{-}L(t/x, 1) < 1$ and $L(t/x, 1) > 0$. In addition, because of $\mu \in (3.5699456, 4]$, then we deduce that

$$0 < L\left(\frac{t}{x}, 2\right) < \mu \cdot L\left(\frac{t}{x}, 1\right). \quad (5)$$

According to the given assumption $0 < t < x/(2 \cdot \mu^2)$ when $n = 2$ and the deduced result $L(t/x, 1) = t/x$ in (3), we deduce (6) from (5), where

$$0 < L\left(\frac{t}{x}, 2\right) < \frac{\mu}{(2 \cdot \mu^2)}. \quad (6)$$

Due to $\mu \in (3.5699456, 4]$, then we have that $0 < L(t/x, 2) < 1/2$ holds.

(3) We assume that Lemma 1 holds when $n = k$, i.e., $0 < L(t/x, k) < 1/2$. According to the assumption $0 < t < x/(2\cdot\mu^n)$ and the deduced result $L(t/x, 1) = t/x$ in (3), we have

$$0 < L\left(\frac{t}{x}, 1\right) < \frac{1}{(2\cdot\mu^k)}. \tag{7}$$

When $n = k + 1$, according to (1), we have

$$\begin{aligned} L\left(\frac{t}{x}, k+1\right) &= \mu\cdot L\left(\frac{t}{x}, k\right)\cdot\left(1 - L\left(\frac{t}{x}, k\right)\right) \\ &= \mu^2\cdot L\left(\frac{t}{x}, k-1\right)\cdot\left(1 - L\left(\frac{t}{x}, k-1\right)\right) \\ &\quad\cdot\left(1 - L\left(\frac{t}{x}, k\right)\right) \\ &= \mu^k\cdot L\left(\frac{t}{x}, 1\right) \\ &\quad\cdot\left(1 - L\left(\frac{t}{x}, 1\right)\right)\cdots\left(1 - L\left(\frac{t}{x}, k-1\right)\right) \\ &\quad\cdot\left(1 - L\left(\frac{t}{x}, k\right)\right). \end{aligned} \tag{8}$$

According to the assumption of (3), i.e., $0 < L(t/x, k) < 1/2$, we have

$$\begin{aligned} &\left(1 - L\left(\frac{t}{x}, 1\right)\right)\cdots\left(1 - L\left(\frac{t}{x}, k-1\right)\right) \\ &\quad\cdot\left(1 - L\left(\frac{t}{x}, k\right)\right) < 1. \end{aligned} \tag{9}$$

On the basis of (7), (8), and (9), we have

$$\begin{aligned} &L\left(\frac{t}{x}, k+1\right) \\ &\quad= \mu^k\cdot L\left(\frac{t}{x}, 1\right) \\ &\qquad\cdot\left(1 - L\left(\frac{t}{x}, 1\right)\right)\cdots\left(1 - L\left(\frac{t}{x}, k-1\right)\right) \\ &\qquad\cdot\left(1 - L\left(\frac{t}{x}, k\right)\right) < \mu^k\cdot L\left(\frac{t}{x}, 1\right) \\ &\quad< \mu^k\cdot\frac{1}{(2\cdot\mu^k)}. \end{aligned} \tag{10}$$

Therefore, we have that $L(t/x, k+1) < 1/2$ holds.

According to the above mathematical induction proofs, we have that Lemma 1 holds. □

Lemma 2. *For any given data x and y, where $1 \le x \le y$, let t/x and t/y as initial values for the logistic map function, i.e., (1), and the nth iteration results are $L(t/x, n)$ and $L(t/y, n)$, respectively, then we have*

$$L\left(\frac{t}{x}, n\right) \ge L\left(\frac{t}{y}, n\right), \tag{11}$$

where $0 < t < x/(2\cdot\mu^n)$, $\mu \in (3.5699456, 4]$, and $x \ge 1$.

Proof. We also use mathematical inductions to prove Lemma 2 as follows.

(1) When $n = 1$, according to (1), we have

$$L\left(\frac{t}{x}, 1\right) = \frac{t}{x} \tag{12}$$

and

$$L\left(\frac{t}{y}, 1\right) = \frac{t}{y}. \tag{13}$$

After applying subtraction between $L(t/x, 1)$ and $L(t/y, 1)$, we have

$$L\left(\frac{t}{x}, 1\right) - L\left(\frac{t}{y}, 1\right) = \frac{t}{x} - \frac{t}{y}. \tag{14}$$

Since $1 \le x \le y$ and $0 < t < x/(2\cdot\mu)$ are given conditions when $n = 1$, we can easily deduce

$$L\left(\frac{t}{x}, 1\right) - L\left(\frac{t}{y}, 1\right) = \frac{t}{x} - \frac{t}{y} \ge 0. \tag{15}$$

Therefore, $L(t/x, 1) \ge L(t/y, 1)$ holds.

(2) When $n = 2$, according to (1), we have

$$L\left(\frac{t}{x}, 2\right) = \mu\cdot\left(\frac{t}{x}\right)\cdot\left(1 - \frac{t}{x}\right) \tag{16}$$

and

$$L\left(\frac{t}{y}, 2\right) = \mu\cdot\left(\frac{t}{y}\right)\cdot\left(1 - \frac{t}{y}\right). \tag{17}$$

After applying subtraction between $L(t/x, 2)$ and $L(t/y, 2)$, we have

$$\begin{aligned} L\left(\frac{t}{x}, 2\right) - L\left(\frac{t}{y}, 2\right) &= \mu\cdot\left(\frac{t}{x}\right)\cdot\left(1 - \frac{t}{x}\right) - \mu\cdot\left(\frac{t}{y}\right) \\ &\quad\cdot\left(1 - \frac{t}{y}\right) \\ &= \mu\cdot\left(\frac{t}{x}\right) - \mu\cdot\left(\frac{t}{x}\right)^2 - \mu \\ &\quad\cdot\left(\frac{t}{y}\right) + \mu\cdot\left(\frac{t}{y}\right)^2 \\ &= \mu\cdot\left(\frac{t}{x} - \frac{t}{y}\right) - \mu\cdot\left(\frac{t}{x} + \frac{t}{y}\right) \\ &\quad\cdot\left(\frac{t}{x} - \frac{t}{y}\right) \\ &= \mu\cdot\left(\frac{t}{x} - \frac{t}{y}\right) \\ &\quad\cdot\left(1 - \left(\frac{t}{x} + \frac{t}{y}\right)\right). \end{aligned} \tag{18}$$

In accordance with the given conditions $1 \le x \le y$, $0 < t < x/(2\cdot\mu^n)$, $n = 2$ and $\mu \in (3.5699456, 4]$, we have

$$0 < \frac{t}{y} \le \frac{t}{x} \le \frac{1}{(2\cdot\mu^2)} < \frac{1}{2}. \tag{19}$$

Then we have

$$\frac{t}{x} - \frac{t}{y} \ge 0 \tag{20}$$

and

$$1 - \left(\frac{t}{x} + \frac{t}{y}\right) > 0. \tag{21}$$

According to (18), (20), and (21), we deduce that

$$L\left(\frac{t}{x}, 2\right) - L\left(\frac{t}{y}, 2\right) = \mu \cdot \left(\frac{t}{x} - \frac{t}{y}\right) \cdot \left(1 - \left(\frac{t}{x} + \frac{t}{y}\right)\right) \ge 0. \tag{22}$$

Therefore, $L(t/x, 2) \ge L(t/y, 2)$ holds.

(3) We assume that Lemma 2 holds when $n = k$; then we have

$$L\left(\frac{t}{x}, k\right) \ge L\left(\frac{t}{y}, k\right). \tag{23}$$

When $n = k + 1$, according to (1), we have

$$L\left(\frac{t}{x}, k+1\right) = \mu \cdot L\left(\frac{t}{x}, k\right) \cdot \left(1 - L\left(\frac{t}{x}, k\right)\right) \tag{24}$$

and

$$L\left(\frac{t}{y}, k+1\right) = \mu \cdot L\left(\frac{t}{y}, k\right) \cdot \left(1 - L\left(\frac{t}{y}, k\right)\right). \tag{25}$$

After applying subtraction between $L(t/x, k+1)$ and $L(t/y, k+1)$, we have

$$L\left(\frac{t}{x}, k+1\right) - L\left(\frac{t}{y}, k+1\right) = \mu \cdot \left(L\left(\frac{t}{x}, k\right) - L\left(\frac{t}{y}, k\right)\right) \cdot \left(1 - L\left(\frac{t}{x}, k\right) - L\left(\frac{t}{y}, k\right)\right). \tag{26}$$

According to Lemma 1, we have $0 < L(t/x, k+1) < 1/2$ and $0 < L(t/y, k+1) < 1/2$; then we deduce

$$0 < 1 - L\left(\frac{t}{x}, k\right) - L\left(\frac{t}{y}, k\right) < 1. \tag{27}$$

According to (7), (23), and (27) and the given condition $\mu \in (3.5699456, 4]$, then we have

$$L\left(\frac{t}{x}, k+1\right) - L\left(\frac{t}{y}, k+1\right) \ge 0. \tag{28}$$

Therefore, $L(t/x, k+1) \ge L(t/y, k+1)$ holds.

According to the above mathematical induction proofs, we have that Lemma 2 holds. □

Definition 3. For a given data item $x_i > 1$, $L(t/x_i, n)$ is denoted as the corresponding *comparable code* of x_i, where $L(*)$ is the logistic map function as (1).

Lemma 4. *For a given data set $X = \{x_1, x_2, \ldots, x_m\}$, where $x_i > 1$ and $i \in \{1, 2, \ldots, m\}$, we can get the corresponding comparable codes set $Y = \{y_1, y_2, \ldots, y_m\}$, where $y_i = L(t/x_i, n)$. Then we have*

$$x_i \le x_j \longleftrightarrow y_i \ge y_j, \tag{29}$$

where $x_i \in X$ and $x_j \in X$.

Proof. To prove Lemma 4, we have to prove the sufficiency and necessity of Lemma 4, respectively, i.e., $x_i \le x_j \longrightarrow y_i \ge y_j$ and $y_i \ge y_j \longrightarrow x_i \le x_j$.

(Sufficiency) According to Lemma 2, we can easily deduce $y_i \ge y_j$ when $x_i \le x_j$, where $y_i = L(t/x_i, n)$ and $y_j = L(t/x_j, n)$. The sufficiency of Lemma 4 is proved.

(Necessity) We prove the necessity of Lemma 4 by contradiction. Assuming that $x_i > x_j$ holds when $y_i \ge y_j$, where $y_i = L(t/x_i, n)$ and $y_j = L(t/x_j, n)$; then we have $L(t/x_i, n) < L(t/x_j, n)$ according to Lemma 2, i.e., $y_i < y_j$. It is obvious that the derivation is inconsistent with the given hypothesis $y_i \ge y_j$. Therefore, if we have $y_i \ge y_j$, where $y_i = L(t/x_i, n)$ and $y_j = L(t/x_j, n)$, then $x_i \le x_j$ holds.

In accordance with the sufficiency and necessity proofs, we have that Lemma 4 holds. □

According to Lemma 4, the computation of comparable is order-preserving reversely with the increasing of input data. For any two real numbers both larger than 1, we can achieve the comparison by comparing their corresponding comparable codes. Obviously, such comparison does not need to know the real values of them. If the given real numbers are less than 1, they are still comparable by using our proposed secure comparison model based on the logistic map. For example, if they are less than -1, the corresponding absolute values will be bigger than 1. And if they are between -1 and 1, by adding the constant number 2, then the result data will be also bigger than 1. Therefore, any two real numbers can be compared. As a result, we have that the proposed secure comparison model based on the logistic map is capable of performing data comparison without knowing their corresponding values. In order to describe conveniently, we focus on the data larger than 1 in the subsequent chapters.

On the basis of secure comparison model, the privacy-preserving sorting mechanism is proposed in the next sections, including the data preprocessing algorithm and privacy-preserving sorting algorithm. The brief flowchart of our proposed work is shown in Figure 1.

5.2. Data Preprocessing Algorithm. DO preprocesses its outsourced data with encryption and logistic map in order to protect private data from CS and support privacy-preserving sorting in CS. We use a symmetric encryption algorithm such as DES and AES to preserve data privacy and the logistic map is utilized to generate comparable codes for secure comparison.

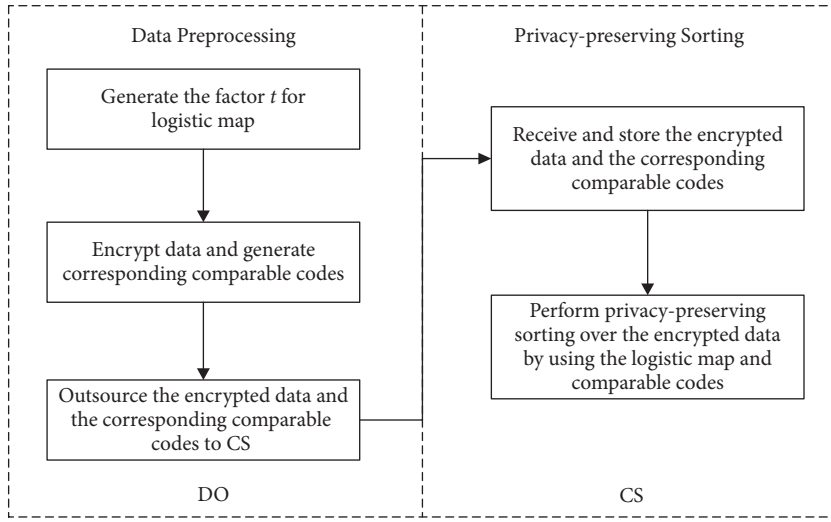


Figure 1: The brief flowchart of privacy-preserving sorting for clouds.

Begin
(1) $t = \text{rand}(0, \min(\{d_1, d_2, \ldots, d_m\})/(2 \cdot \mu^n))$;
(2) **FOR** $d_i \in \{d_1, d_2, \ldots, d_m\}$ **DO**
(3) Create a data pair g_i for d_i;
(4) $g_i.e = Enc(d_i, k)$; //generate encrypted data
(5) $g_i.c = L(t/d_i, n)$; //generate comparable code
(6) **END FOR**
(7) Upload and outsource $\{g_1, g_2, \ldots, g_m\}$ to CS;
END

Algorithm 1: **DP-LM**($\{d_1, d_2, \ldots, d_m\}$).

We assume that the outsourced data of DO are $\{d_1, d_2, \ldots, d_m\}$. After data preprocessing, a data pair $g_i = (e, c)$ will be generated for d_i, where $g_i.e$ and $g_i.c$ are the corresponding encrypted data and comparable code of d_i. In addition, we assume that k is a private key, while μ and n are bifurcation parameter and number of iterations of the logistic map function, respectively. And k, μ, and n are all owned by DO privately. The data preprocessing algorithm based on the logistic map (DP-LM) is shown in Algorithm 1.

In Algorithm 1, $\text{rand}(0, x)$ is to randomly pick a float number between 0 and x, $\min(S)$ is to get the minimum of the set S, $Enc(d_i, k)$ is to encrypt d_i with private key k by a symmetric encryption, and $L(*)$ is a logistic map function as (1). After finishing data preprocess, the generated data pairs will be uploaded and outsourced to CS.

5.3. Privacy-Preserving Sorting Algorithm. Privacy-preserving sorting is performed in CS after receiving the outsourced data from DO. Obviously, traditional sorting algorithms (e.g., merge sort, quicksort, and heap sort) cannot solve the problem of sorting encrypted data items, but by introducing the proposed secure comparison model, the encrypted data will be sorted by using the corresponding comparable codes.

We give the privacy-preserving quick sorting algorithm based on the logistic map (PQS-LM) for sorting over encrypted data in CS. The specific implementation of PQS-LM is shown in Algorithm 2.

Begin
(1) **IF** $start < end$ **THEN**
(2) $i = start$, $j = end + 1$;
(3) **WHILE** TRUE **DO**
(4) **WHILE** $i < end \wedge g_{start}.c < g_i.c$ **DO**
(5) i++;
(6) **END WHILE**
(7) **WHILE** $j > start \wedge g_j.c < g_{start}.c$ **DO**
(8) j--;
(9) **END WHILE**
(10) **IF** $i < j$ **THEN**
(11) $Swap(g_i, g_j)$;
(12) **ELSE**
(13) Finish the current loop and start the next loop;
(14) **END IF**
(15) **END WHILE**
(16) $Swap(g_j, g_{start})$;
(17) PQS-LM($\{g_1, g_2, \ldots, g_m\}$, *start*, j-1);
(18) PQS-LM($\{g_1, g_2, \ldots, g_m\}$, $j + 1$, *end*);
(19) **END IF**
END

Algorithm 2: **PQS-LM**($\{g_1, g_2, \ldots, g_m\}$, *start*, *end*).

In Algorithms 2, $Swap(x, y)$ is to swap the positions of elements x and y. The comparable codes are compared during sorting procedures, and the number of comparisons determines the efficiency of sorting. The complexity of the comparison based on our proposed model is equivalent to the comparison of plaintext. Therefore, the time complexity of PQS-LM is $O(m \cdot \log_2 m)$.

Other classic sorting algorithms, such as merge sorting and heap sorting, can also be improved to be the corresponding privacy-preserving sorting algorithms as Algorithm 2 on the basis of the proposed secure comparison model. We denote the privacy-preserving merge sorting and heap

sorting as PMS-LM and PHS-LM, respectively. Because the implementation ideas are similar to PQS-LM, we omit the details of those algorithms. The analysis and performance evaluations of our proposed privacy-preserving sorting algorithms will be given in the latter sections.

6. Security Analysis

There are two types of data outsourced in CS. One is the encrypted data generated by a symmetric encryption, and the other is the comparable code generated by the logistic map. The former is to protect data privacy, while the latter is to support secure comparisons. For encrypted data, it has an identical security level with the adopted symmetric encryption. For comparable codes generated by the logistic map, we conduct security analysis as follows.

(1) Space of initial parameters: the data preprocessing algorithm in this paper is based on a logistic chaotic system. The corresponding parameters are initialized before preprocessing, including the number of iterations n, the constraint factor t, and the bifurcation parameter μ. We assume that the attacker uses an exhaustive attack against the initial parameters. The precision of t and μ is assumed to be 10^{-p} and 10^{-q}, respectively. The space of initial parameters is $10^{p+q} \bullet n$. For example, if we take $p = 32$, $q = 32$ and randomly pick n from the interval $[1, 1000]$, then the space of initial parameters is 10^{67}. It is computation infeasible to commit successful attacks by using exhaustive search in such a large space.

(2) Sensitivity of initial parameters: since the sequence generated by the logistic map is extremely sensitive to initial parameters, any small modification of them leads to completely different results. For example, we take the same μ and n, where $\mu = 3.95362$ and $n = 3$, while we take two different constraint factors which are very close to each other, such as $t_1 = 1 \times 10^{-8}$ and $t_2 = 2 \times 10^{-8}$. For the real number 11, we will get two completely different comparable codes $L(t_1/11, 3) = 5.62 \times 10^{-8}$ and $L(t_2/11, 3) = 1.12 \times 10^{-7}$.

(3) Antistatistic ability: the logistic map has good cryptographic properties such as sensitivity to initial parameters, driven by white noise, unpredictability, etc. [31]. Even if an attacker obtains some statistic information about the input data and the corresponding comparable codes, he or she still cannot get configurations of initial parameters. Lots of simulation cases show that the data generated by the logistic map with different initial parameters are in equi-distribution [29, 32] which can prevent statistical attacks. In addition, we will give the correlation coefficient evaluation in the next section to analyze the antistatistic ability quantitatively.

As a result, we have that our proposed algorithms can support sorting over encrypted data while preserving data privacy.

7. Experiments

In this section, we give the correctness, correlation coefficient, and performance evaluations of our proposed method. The experimental datasets are generated by a random number generator. The software environment of the experiment is Windows 10 and NetBeans 8, and the hardware environment is Core i5 5200U and 8 GB DDR3 RAM.

7.1. Correctness Evaluation on Secure Comparison Model. We proposed the secure comparison model which is on the basis of the logistic map. It is the foundation of achieving the privacy-preserving sorting algorithms. Theoretical proofs are given to prove the correctness of the secure comparison model in the above sections, such as the proofs in Lemmas 1, 2, and 4. Additionally, we give the correctness evaluation on the proposed model by quantitative experiments.

In this evaluation, almost 100 thousand random numbers are generated as the input, and the corresponding comparable codes are calculated by the logistic map function with the initial parameter configuration as μ=3.67435, $n = 100$, and $t = 0.423124/(2 \times \mu^n)$. The diagram of the input data and the corresponding comparable codes are shown in Figure 2.

Figure 2 shows that the values of comparable codes decrease along with the increasing of the input data values. It indicates that the comparable codes computation is with the order-preserving property which is consistent with the proposed conclusions of our proposed secure comparison model. Therefore, the experimental result has verified the correctness of the security comparison model quantitatively.

7.2. Correlation Coefficient Evaluation on Secure Comparison Model. We use the Pearson correlation coefficient formula [33] to analyze the correlation between the input data and the corresponding comparable codes generated in secure comparison model. The correlation coefficient formula is shown as

$$C = \frac{n \cdot \sum_{x_i \in DS} \left(x_i \cdot L\left(t/x_i, n\right)\right) - \sum_{x_i \in DS} x_i \cdot \sum_{x_i \in DS} L\left(t/x_i, n\right)}{\sqrt{n \cdot \sum_{x_i \in DS} {x_i}^2 - \left(\sum_{x_i \in DS} x_i\right)^2} \cdot \sqrt{n \cdot \sum_{x_i \in DS} L\left(t/x_i, n\right)^2 - \left(\sum_{x_i \in DS} L\left(t/x_i, n\right)\right)^2}}, \tag{30}$$

where x_i and $L(t/x_i, n)$ are the input data and corresponding comparable codes, respectively, C is the correlation coefficient factor, and DS is the evaluated dataset. We calculate correlation coefficients on the basis of five datasets which are generated by a random number generator, and the results are shown in Table 2.

According to the result of Table 2, we can see that the average correlation coefficient decreases with the increasing

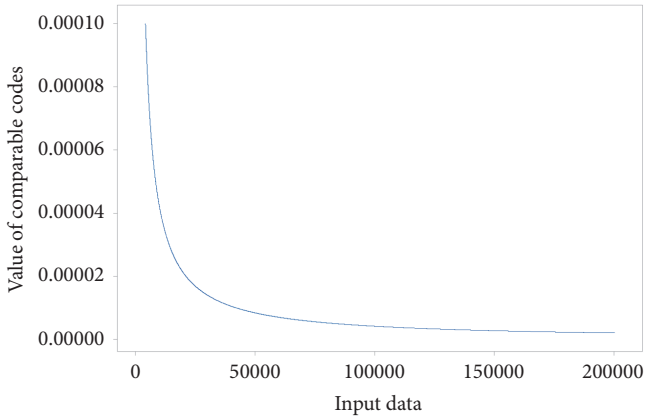


Figure 2: Values of comparable codes versus input data.

Table 2: Correlation coefficients of random datasets.

Dataset ID	Dataset scale	Average correlation coefficient
1	1000	-0.156
2	5000	-0.110
3	9000	-0.087
4	13000	-0.066
5	17000	-0.068

of dataset scales. And the average correlation coefficient is very small which indicates that the correlation between the input data and corresponding comparable codes is negligible. Therefore, our proposed secure comparison model is with the antistatistic ability.

7.3. Performance Evaluation on Algorithms. We implement our proposed logistic map-based data preprocessing and privacy-preserving sorting algorithms. To make comparisons with related works, we adopt the classic Boldyreva's order-preserving symmetric encryption (OPE) [12] to implement privacy-preserving sorting, which is more secure than [11] and more efficient than [13, 14]. We denote the OPE based data preprocessing as DP-OPE and denote the OPE based privacy-preserving quick sorting, merge sorting, and heap sorting algorithms as PQS-OPE, PMS-OPE, and PHS-OPE, respectively. Then we evaluate and compare the time cost performance of those algorithms.

It is noticeable that there are fully homomorphic encryption (FHE) based privacy-preserving sorting schemes proposed in [18, 19]. But they are too slow because of the complexity of FHE. The experiments of them show that thousands of seconds are consumed even sorting only 40 encrypted data items. Thus, we do not choose them to implement performance comparisons.

7.3.1. Evaluation on Time Cost of Data Preprocessing. The time cost of DP-OPE and DP-LM is evaluated on the basis of five given datasets. The experimental result is shown in Table 3.

Table 3: The time cost of data preprocessing (ms).

Dataset scale	DP-OPE	DP-LM
1000	1194.39	4.35
5000	4036.15	14.36
9000	6930.81	20.05
13000	10411.12	35.91
17000	14178.67	58.01

Table 3 shows that the time costs of DP-LM and DP-OPE are both increasing along with the expansion of datasets, but DP-LM is obviously much faster than DP-OPE. The reason is given as follows. DP-OPE needs to execute order-preserving encryption for plaintext by mapping amount of consecutive integers in a domain to integers in a much larger range. Each integer is assigned a pseudorandom value in its subrange. The OPE algorithm recursively bisects the range and samples from the domain at each recursion until it hits the input plaintext value. Thus, the calculation load of OPE is higher relatively which makes DP-OPE much slower than DP-LM.

7.3.2. Evaluation on Time Cost of Data Sorting. We also use the same datasets to evaluate the privacy-preserving sorting algorithms based on the logistic map and OPE. The result is shown in Table 4.

The experimental result in Table 4 shows that the performance of our proposed privacy-preserving sorting algorithms is better than those sorting algorithms based on OPE, respectively. The reason is that the output data of OPE, which is used for privacy-preserving sorting, is more complex than the comparable codes generated by the logistic map.

8. Conclusions

When the clouds provide outsourcing services, the privacy of outsourced data, such as national defence data and human health data, can be protected by common encryption. However, those encrypted data are useless for data sorting which is a common operation in many areas, such as machine learning, service recommending, and data query. It is a challenge to achieve privacy-preserving sorting in clouds. In this paper, we introduce a secure comparison model based on the logistic map and propose privacy-preserving sorting algorithms. The security analysis and experimental result show that the proposed algorithms can protect data privacy while providing efficient sorting on encrypted data.

Data Availability

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

Disclosure

The abstract of this paper appeared in the 4th International Conference on Cloud Computing and Security (ICCCS 2018), June 8-10, Haikou. This version is the full paper.

Table 4: The time cost of privacy-preserving algorithms (ms).

Dataset scale	Privacy-preserving quick sorting		Privacy-preserving merge sorting		Privacy-preserving heap sorting	
	PQS-LM	PQS-OPE	PMS-LM	PMS-OPE	PHS-LM	PHS-OPE
1000	0.63	0.65	0.76	0.81	0.74	0.75
5000	1.02	1.55	5.45	6.31	2.35	3.31
9000	3.15	4.23	7.23	7.85	3.19	4.12
13000	3.91	4.41	8.03	9.47	3.92	4.65
17000	4.65	5.01	11.75	12.31	5.31	5.45

Conflicts of Interest

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

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant nos. 61872197, 61572263, 61672297, 61502251, and 61472193; the Natural Science Foundation of Jiangsu Province under Grant nos. BK20151511, BK20141429, and BK20161516; the Postdoctoral Science Foundation of China under Grant no. 2015M581794; the Natural Science Foundation of Anhui Province under Grant no. 1608085MF127; and the Natural Research Foundation of Nanjing University of Posts and Telecommunications under Grant no. NY217119.

References

[1] P. H. Abreu, M. S. Santos, M. H. Abreu, B. Andrade, and D. C. Silva, "Predicting Breast Cancer Recurrence Using Machine Learning Techniques," *ACM Computing Surveys*, vol. 49, no. 3, pp. 1–40, 2016.

[2] D. E. Jones, H. Ghandehari, and J. C. Facelli, "A review of the applications of data mining and machine learning for the prediction of biomedical properties of nanoparticles," *Computer Methods and Programs in Biomedicine*, vol. 132, pp. 93–103, 2016.

[3] L. Zhu, Y. Zhang, Z. Pan, R. Wang, S. Kwong, and Z. Peng, "Binary and multi-class learning based low complexity optimization for HEVC encoding," *IEEE Transactions on Broadcasting*, vol. 63, no. 3, pp. 547–561, 2017.

[4] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[5] J. Wang, J. Wang, Y. Wu et al., "A Machine Learning Framework for Resource Allocation Assisted by Cloud Computing," *IEEE Network*, vol. 32, no. 2, pp. 144–151, 2018.

[6] J. Zhu and X. Li, "Scheduling for multi-stage applications with scalable virtual resources in cloud computing," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 5, pp. 1633–1641, 2017.

[7] D. C. Nascimento, C. E. Pires, and D. G. Mestre, "Applying machine learning techniques for scaling out data quality algorithms in cloud computing environments," *Applied Intelligence*, vol. 45, no. 2, pp. 530–548, 2016.

[8] Y. Liu, H. Peng, and J. Wang, "Verifiable Diversity Ranking Search Over Encrypted Outsourced Data," *Computers, Materials & Continua*, vol. 55, no. 1, pp. 37–57, 2018.

[9] S. H. Albakri, B. Shanmugam, G. N. Samy, N. B. Idris, and A. Ahmed, "Security risk assessment framework for cloud computing environments," *Security and Communication Networks*, vol. 7, no. 11, pp. 2114–2124, 2014.

[10] J. Cheng, R. Xu, X. Tang, V. Sheng, and C. Cai, "An Abnormal Network Flow Feature Sequence Prediction Approach for DDoS Attacks Detection in Big Data Environment," *Computers, Materials & Continua*, vol. 55, no. 1, pp. 95–119, 2018.

[11] R. Agrawal, J. Kiernan, R. Srikant, and Y. R. Xu, "Order preserving encryption for numeric data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '04)*, pp. 563–574, ACM, New York, NY, USA, June 2004.

[12] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Order-preserving symmetric encryption," in *Advances in Cryptology-EUROCRYPT 2009*, vol. 5479, pp. 224–241, Springer, Berlin, Germany, 2009.

[13] V. Jaiman and G. Somani, "An order preserving encryption scheme for cloud computing," in *Proceedings of the 7th International Conference on Security of Information and Networks, SIN 2014*, pp. 211–216, Glasgow, UK, September 2014.

[14] Z. Liu, X. Chen, J. Yang, C. Jia, and I. You, "New order preserving encryption model for outsourced databases in cloud environments," *Journal of Network and Computer Applications*, vol. 59, pp. 198–207, 2016.

[15] M. Naveed, S. Kamara, and C. V. Wright, "Inference attacks on property-preserving encrypted databases," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS 2015*, pp. 644–655, New York, NY, USA, October 2015.

[16] C. Aguilar-Melchor, S. Fau, C. Fontaine, G. Gogniat, and R. Sirdey, "Recent advances in homomorphic encryption: A possible future for signal processing in the encrypted domain," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 108–117, 2013.

[17] A. Chatterjee, M. Kaushal, and I. Sengupta, "Accelerating Sorting of Fully Homomorphic Encrypted Data," in *Progress in Cryptology – INDOCRYPT 2013*, vol. 8250 of *Lecture Notes in Computer Science*, pp. 262–273, Springer, 2013.

[18] A. Chatterjee and I. Sengupta, "Searching and sorting of fully homomorphic encrypted data on cloud," *IACR Cryptology ePrint Archive*, p. 981, 2015.

[19] A. Chatterjee and I. Sengupta, "Translating Algorithms to Handle Fully Homomorphic Encrypted Data on the Cloud," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 287–300, 2018.

[20] W. Fu, B. Yan, and X. Wu, "Data possession provability on semi-trusted cloud storage," in *Proceedings of the 4th International Conference Cloud Computing*, pp. 199–209, Springer, 2013.

[21] C. Gentry and S. Halevi, "Implementing Gentry's fully-homomorphic encryption scheme," *IACR Cryptology ePrint Archive*, p. 520, 2010.

[22] H. Dai, H. Ren, Z. Y. Chen et al., "Privacy-Preserving Sorting Algorithms based on Logistic Map for Clouds," in *Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS 2018)*, 2018.

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

[24] J. Banks, J. Brooks, G. Cairns, G. Davis, and P. Stacey, "On Devaney's definition of chaos," *The American Mathematical Monthly*, vol. 99, no. 4, pp. 332–334, 1992.

[25] B. Yang and X. Liao, "Period analysis of the Logistic map for the finite field," *Science China Information Sciences*, vol. 60, no. 2, p. 22302, 2017.

[26] Y. Deng, H. Hu, W. Xiong, N. N. Xiong, and L. Liu, "Analysis and Design of Digital Chaotic Systems with Desirable Performance via Feedback Control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 8, pp. 1187–1200, 2015.

[27] S. C. Phatak and S. S. Rao, "Logistic map: a possible random-number generator," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 51, no. 4, pp. 3670–3678, 1995.

[28] G. Ye and X. 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.

[29] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image and Vision Computing*, vol. 24, no. 9, pp. 926–934, 2006.

[30] B. Murugan, A. G. Nanjappa Gounder, and S. Manohar, "A hybrid image encryption algorithm using chaos and Conway's game-of-life cellular automata," *Security and Communication Networks*, vol. 9, no. 7, pp. 634–651, 2016.

[31] S. G. Mallat, "Theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[32] V. Patidar, K. K. Sud, and N. K. Pareek, "A pseudo random bit generator based on chaotic logistic map and its statistical testing," *Informatica*, vol. 33, no. 4, pp. 441–452, 2009.

[33] Wikipedia, "Pearson correlation coefficient," 7 April 2018, https://en.wikipedia.org/wiki/Pearson_correlation_coefficient.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 7096271, 10 pages
https://doi.org/10.1155/2018/7096271

Research Article

An Adaptive Audio Steganography for Covert Wireless Communication

Guojiang Xin,[1] **Yuling Liu**,[2] **Ting Yang**,[2] **and Yu Cao**[3]

[1]*School of Informatics, Hunan University of Chinese Medicine, 410208, China*
[2]*College of Computer Science and Electronic Engineering, Hunan University, 410082, China*
[3]*Department of Computer Science, University of Massachusetts Lowell, 01854, USA*

Correspondence should be addressed to Yuling Liu; yuling_liu@126.com

Received 27 May 2018; Accepted 1 August 2018; Published 16 August 2018

Academic Editor: Zhaoqing Pan

In recent years, the wide applications of the wireless sensor networks have achieved great success. However, the security is a critical issue in many scenarios ranging from covert military operations to the organization of the social unrest. Because the traditional encrypting methods are easy to arouse suspicion, an adaptive audio steganography method is proposed. The method is based on interval and variable low bit coding, which can be applied to covert wireless communication. The interval for embedding secret messages into the audio file and the threshold in variable low bit coding are used for selecting the embedding location and embedding bits adaptively; thus the embedding capacity and the embedding rate are variable. Experimental results demonstrate that the proposed method has better performance in embedding rate and invisibility than other audio steganography methods.

1. Introduction

Nowadays, wireless sensor networks (WSNs) and multimedia are getting increasing attention from both academic and industry communities, which hold the promise of facilitating large-scale and real-time data processing including video and audio in complex real-time multimedia environments, retrieving multimedia content, and object detection [1–3]. The security for real-time communication is a critical issue that must be resolved. In this paper, real-time audio communication is focused. Most often, cryptography techniques are utilized for the security of WSNs, which are based on rendering the content of a message garbled to unauthorized people [4]. However, it is well known that cryptography methods make people become aware of the existence of secret information. Hence, steganography, which is a process of embedding secret messages into a cover signal to avoid illegal detection [5], was introduced to ensure the transmission safety of the secret information and authenticate the multimedia data in WSNs [6, 7]. A verifiable diversity ranking search scheme over encrypted outsourced data is proposed while preserving privacy in cloud computing, which also supports search results verification [8].

In recent years, many methods have been proposed for steganography and steganalysis, such as coverless information hiding [9], steganography, and steganalysis based on deep learning [10, 11]. However, it is still a challenging problem to use audio signal as the cover signal, because Human Auditory System (HAS) is very sensitive. The existing typical audio steganography methods can be divided into time domain methods, transform domain methods, compressed domain methods, and phase methods. Most time domain methods refer to the methods based on least significant bits (LSB) of the audio data [12]. The algorithm performs scrambling pretreatment using logistic chaotic on the secret watermarking information to make watermarking bits garbled. The public audio signal is transformed by DWT, so that the secret information is embedded into the selected wavelet coefficients with multi-resolution. The original carrier is not required in watermarking recovery. Another time domain method is echo hiding method [13]. Because the weak signal becomes unable to hear after the strong signal disappears, the echo is introduced to achieve information hiding. This method is robust but has high computational complexity, low capacity, and low extraction accuracy. The transform

domain methods firstly need to do discrete Fourier transform (DFT), discrete cosine transform (DCT), or discrete wavelet transform (DWT), or a combination of the three transforms, and then select certain frequency coefficients to embed the secret message. These methods have good robustness but low capacity and relatively complex calculation [14, 15]. The compressed domain methods are suitable for the compressed audio files. These methods are not only difficult to implement but also have limited hiding capacity [16]. Because HAS is sensitive to relative phase, but not sensitive to absolute phase, the methods based on phase coding are proposed. The secret message is embedded by replacing the absolute phase of the original audio with the relative phase. These methods are robust, but the capacity is limited and the calculation is more complicated [17, 18]. According to the different application scenarios, a method with a rational selection among hiding capacity, invisibility, and robustness is expected.

There are some data embedding algorithms based on LSB. The lowest bit coding is the method that embeds secret data only in LSB. This method can minimize the transition and obtain the embedding capacity up to 12.5% of the wav file [19]. The parity coding method breaks a signal down into separate regions of samples instead of individual sample. If the secret bit to be encoded does not match with the sample region's parity bit, then it flips the LSB of one sample in the region [20]. Exclusive OR (XOR) operation is firstly performed on the LSB, and then the LSB of the sample is modified or kept unchanged according to the result of XOR operation and the message bit to be embedded [21]. About bit selection, the different bits are selected to hide the secret data in each sample. For instance, the first two most significant bits (MSB) of a sample are used for bit selection and only the first three LSB are used for data embedding [22]. A parameter R, which is an interval, is set for embedding the bit stream of a secret message into each byte of audio packets. If $R = 1$, the secret message is embedded into every 2 bytes of audio streams, while if $R = 2$, the secret message is embedded into every 3 bytes of audio streams [23]. While through selecting some samples, only a few samples are used for data hiding instead of using all the samples. For instance, the first three MSB are used to select the next sample for embedding the secret bits [22]. Fibonacci sequence is used to select the samples for data hiding [24]. In average amplitude method, the average amplitude data of surrounding audio data is used as a threshold. If the amplitude level is bigger than the average value, then 2 LSB are used for embedding; otherwise the secret data will not be embedded [19]. The variable low bit coding is the improved version of the lowest bit coding which can increase the embedding capacity. Because the sound is a silence at the middle range of audio data, the data cannot be embedded in the middle range. Two thresholds are defined based on the standard level, which is calculated by the middle range. The thresholds are used for selecting the embedding bits.

Many improved LSB methods are based on the combination of several algorithms. In [25], a dual random LSB method is proposed by combining Huffman coding with RSA encryption. The method embeds the data in variable LSB depending on the MSB of cover audio samples. Also, there is an increase in capacity due to the use of Huffman coding. In [26], for each sample, the third and fourth LSB are replaced with the secret message and the second and fifth LSB are altered by using an intelligent algorithm so that the stego sample gets minimized. In [27], the audio samples are 16 bits. The secret message is embedded into the coefficients of a cover audio. Each secret bit is embedded into the selected position of a cover coefficient. The positions are selected from the 0th to 7th LSB based on the MSB. In [28], a wav-audio steganography algorithm based on modifying amplitude is proposed. Sampling points are grouped by each three successive ones and the amplitude values are calculated. Secret message is embedded by modifying the amplitude value of the second sampling point by comparing the amplitude value of the second sampling point with the average value of the first and the third sampling points. In [29], a novel reversible natural language watermarking method combines arithmetic coding and synonym substitution operations. The original context can be perfectly recovered by decompressing the extracted compressed data and substituting the replaced synonyms with their original synonyms.

In this paper, an improved audio steganography method for covert wireless communication is proposed by incorporating the variable low bit coding with the embedding intervals. Different from the steganography methods based on Voice over Internet Protocol (VoIP) [30], the proposed method will be applied to nonreal-time audio data in covert wireless communication, in which the hiding capacity, instantaneity, and security are concerned. The method has good performance in terms of the embedding rate, hiding capacity, and invisibility.

The rest of the paper is organized as follows. The variable low bit coding and the interval setting are described in Section 2. In Section 3, the proposed method is presented in details. The experiments and results analysis are shown in Section 4. The conclusion is in Section 5.

2. Related Work

2.1. Variable Low Bit Coding. Variable low bit coding is an improved LSB method, which can increase the capacity. The middle range of data represents the silence. Supposing that the audio file is sampled every 16 bits, the range of audio data is -32768~32767. When the audio data is zero, the sound is silent. Suppose that the audio file is sampled every 8 bits; the range of audio data is from 0 to 255. The middle range of data is 128 and the sound is silent in that range. Because embedding data into the silent sound will reveal the secret data, the data cannot be embedded in the middle range. By calculating the standard level, two thresholds t_1 and t_2 are set. If the amplitude value is smaller than t_1, the secret data will not be embedded; if it is between t_1 and t_2, then one bit is used for data embedding; if it is bigger than t_2, then two bits are used for data embedding.

2.2. The Interval Setting. Parameter R is set as the interval for embedding secret message into the audio file, which means

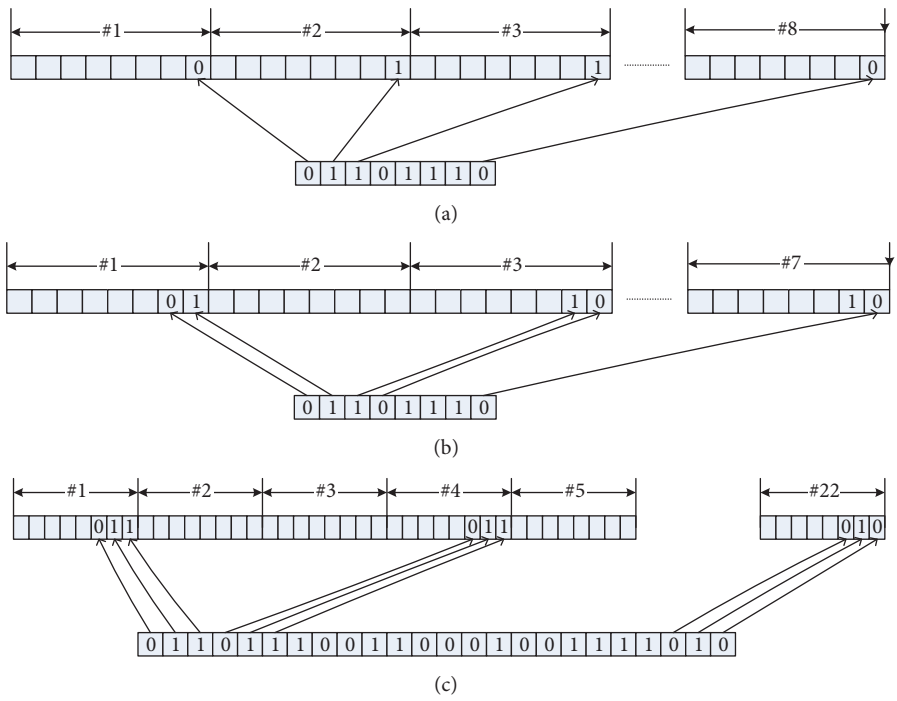


Figure 1: Embedding the secret message with different values of R. (a) $R = 0$; (b) $R = 1$; (c) $R = 2$.

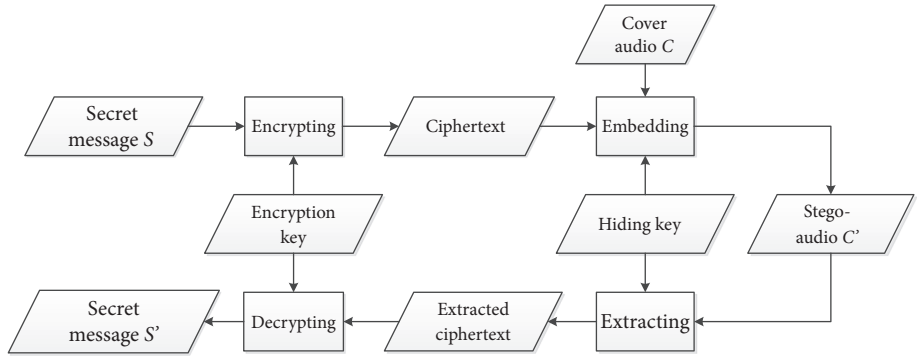


Figure 2: The framework of the proposed method.

that $R+1$ bits of secret message are embedded into the sample while the interval is R.

Figure 1 shows embedding of the secret message with different values of R. Suppose that the audio file is sampled every 8 bits. Figure 1(a) shows that when $R = 0$, the traditional LSB method is used. The bits of the secret message are embedded into the LSB of each sample. It is obvious that 1 byte message can be embedded into 8 bytes of audio signal. Figure 1(b) illustrates that when $R = 1$, 2 bits of secret message can be embedded into the current sample while the interval is one sample. In this case, 1 byte of secret message is embedded into 7 bytes of the audio file. Figure 1(c) illustrates that the method can embed 3 bits of the secret message into the current sample while the intervals are two samples, when $R = 2$. Thus 3 bytes of the secret message can be embedded into 22 bytes of the audio data.

3. The Proposed Method

In this section, a new audio steganography method is described in details. The proposed method can adaptively select the embedding location and embedding bits through setting the interval and variable low bit coding. In this method, wav files are used as the audio files, because the wav file format is a subset of Microsoft's RIFF specification for the storage of multimedia files, which are widely used in Windows Operating System. A wav file includes two parts, the header of the audio file and the audio data. The first 44 bytes of the audio file are the header and the rest are the data. Because the header is constant, the secret data should be embedded into the audio data, not into the header. The framework of the proposed method is shown in Figure 2.

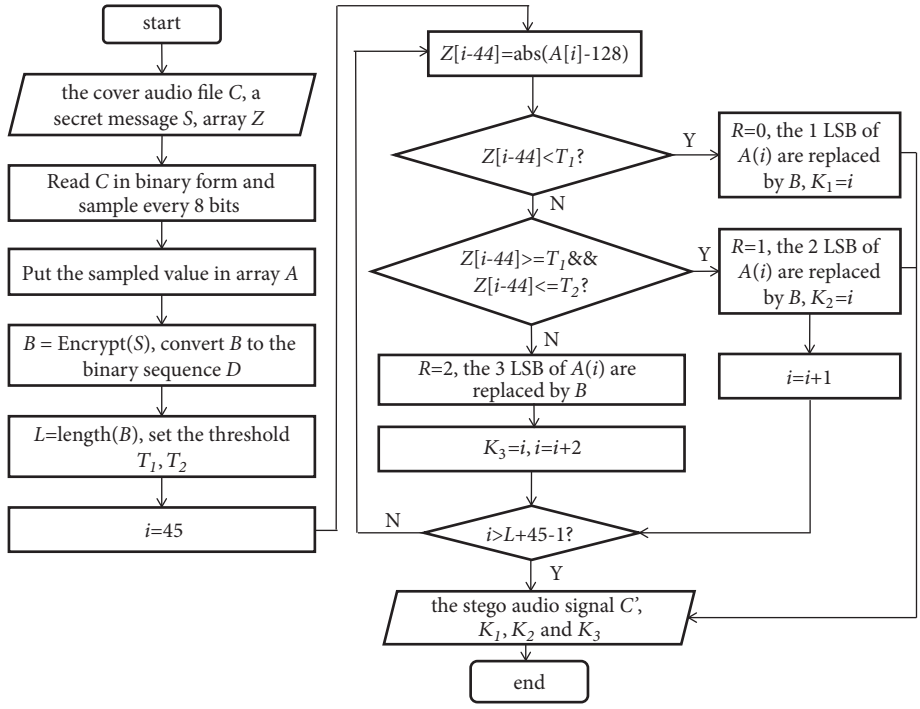


Figure 3: Flowchart of the embedding procedure.

3.1. Improved Variable Low Bit Coding. The LSB method is to embed secret data only in the LSB. The variable low bit coding method improves the lowest bit coding and can increase embedding capacity. In this paper, the variable low bit coding is improved by combining the embedding interval, which means that not every audio sample data is used for embedding the secret information. The 8 bits mono wav files are selected as the carriers, whose sampling frequency is 11.025 kHz. Two thresholds T_1 and T_2 are set. If the absolute value of the difference between 128 and the audio data is less than T_1, then one bit of LSB is used for data embedding; if it is between T_1 and T_2, then two bits are used for data embedding; if it is more than T_2, then three bits are used for data embedding.

3.2. The Embedding Procedure. Parameter R is set as the interval for embedding the secret message into the audio file, which is shown in Figure 1. According to the improved variable low bit coding method, T_1 and T_2 are the thresholds, which can select the embedding bits of a sample. Therefore, combining the interval R with the thresholds, the proposed method can adaptively select the embedding location and the embedding bits. The embedding procedure is shown in Figure 3.

The detail steps are presented as follows:

Input: It is a cover audio file C and a secret message S.

Output: It is a stego cover audio file C'.

Step 1: Read a cover audio file C in binary form.

Step 2: The converted binary audio file is sampled every 8 bits. Put it into an array A.

Step 3: Input the secret message S to be embedded.

Step 4: Encrypt S with AES algorithm to B, and then put B into an array D in the form of binary sequence.

Step 5: Calculate the length of B as L.

Step 6: Set the thresholds T_1 and T_2 according to the middle range of the audio data.

Step 7: Put the absolute value of the difference between the audio data and 128 into an array Z.

For i = 45: L + 45-1

$Z(i\text{-}44)$ = abs($A(i)$-128);

//Compare the relationship between $Z(i\text{-}44)$ and the thresholds T_1, T_2.

If $Z(i\text{-}44) < T_1$, **then** set $R = 0$ and the LSB of $A(i)$ are replaced by the being embedded messages. Put i into a key array K_1;

Else if $Z(i\text{-}44) >= T_1$ && $Z(i\text{-}44) <= T_2$, **then** set $R = 1$ and the 2 LSB of $A(i)$ are replaced by the being embedded messages. Put i into a key array K_2, $i = i + 1$;

Else set $R = 2$ and the 3 LSB of $A(i)$ are replaced by the being embedded messages. Put i into a key array K_3, $i = i + 2$.

End.

Step 8: The modified audio samples are formed into the stego audio signal C', and then send C', K_1, K_2, and K_3 as the hiding key to the receiver.

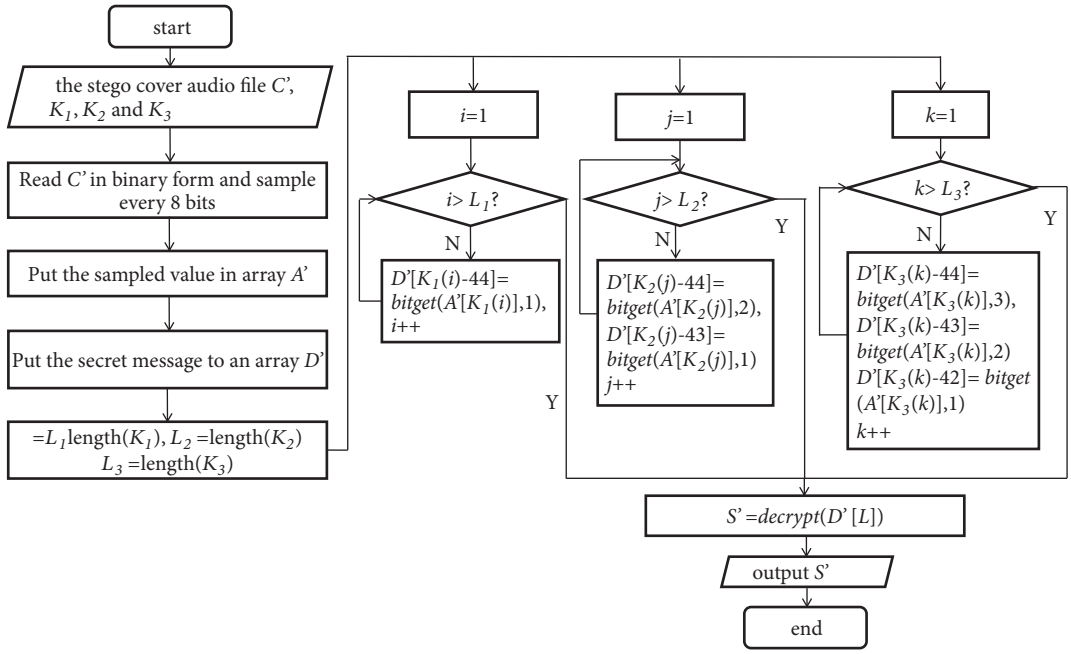


Figure 4: Flowchart of the extracting procedure.

3.3. The Extracting Procedure. The extracting procedure is an inverse process of the embedding procedure and is shown in Figure 4.

The detail steps are described as follows:

Input: It is the stego cover audio file C'.

Output: It is the secret message.

Step 1: Read the stego audio file C' in binary form.

Step 2: The converted binary audio file is sampled every 8 bits and is put into an array A'.

Step 3: Extract the secret message according to the hiding key arrays K_1, K_2, and K_3, and the secret message is put into an array D'.

Calculate the length of K_1, K_2, and K_3 as L_1, L_2, and L_3,

For $i = 1 : L_1$

$D'(K_1(i)\text{-}44) = bitget(A'(K_1(i)), 1);$

End

For $i = 1 : L_2$

$D'(K_2(i)\text{-}44) = bitget(A'(K_2(i)), 2)$

$D'(K_2(i)\text{-}43) = bitget(A'(K_2(i)), 1);$

End

For $i = 1 : L_3$

$D'(K_3(i)\text{-}44) = bitget(A'(K_3(i)), 3);$

$D'(K_3(i)\text{-}43) = bitget(A'(K_3(i)), 2);$

$D'(K_3(i)\text{-}42) = bitget(A'(K_3(i)), 1);$

End

Step 4: Decrypt the L bits of D' to obtain the corresponding secret information S'.

4. Experiments and Results Analysis

For implementing the program, the programming techniques based on socket and multithread are used, Microsoft Visual C++ 2013 is used to compile and assemble, and Windows 7 is the operating platform.

The program is composed of two parts, the ordinary wireless communication and the application of steganography. The sending node and the receiving node can transmit the audio data to embed and extract the secret information. For simplicity, both the sending node and the receiving node are implemented in one program. The communicating parties just need to install the program to their PC and enter the host's IP, and then they can communicate with each other by hiding information. Because the wav format is widely used, we use the wav-audio files as the steganography carriers. We collected 50 wav files with mono from the WSNs. The sampling rate is 11.025 KHz with 8 bits per sample. The size of the audio files varies from 39KB to 652KB.

4.1. An Example. The following is an example to illustrate the embedding and extracting procedures. Suppose that the encrypted message to be embedded is "10011011"; the cover audio file is put into the array A after sampling and quantization. The cover audio data is shown in Table 1.

Setting $T_1 = 30$, $T_2 = 40$, according to the embedding method, if $|A(45)\text{-}128| < T_1$, $R = 0$, then LSB of $A(45)$ are replaced by "1"; putting "45" into the array K_1; if $|A(46)\text{-}128| > T_2$, $R = 2$, then 3 LSB of $A(46)$ are replaced by "001"; putting "46" to the array K_3; if $T_1 < |A(49)\text{-}128| < T_2$, $R = 1$, then 2 LSB of $A(49)$ are replaced by "10"; putting "49" into the array K_2; if $T_1 < |A(51)\text{-}128| < T_2$, $R = 1$, then 2 LSB

Table 1: The cover audio data.

i	$A(i)$
.....	.....
45	123
46	235
47	228
48	155
49	89
50	120
51	165
.....	.....

Table 2: The stego audio data.

i	$A'(i)$
.....	.....
45	123
46	233
47	228
48	155
49	90
50	120
51	167
.....	.....

of $A(51)$ are replaced by "11"; putting "51" into the array K_2. The stego audio data A' is shown in Table 2.

When the receiver obtains the stego audio file and the hiding key, he can extract the secret message. The extracted secret bits are put into the array D'.

According to the array K_1, the receiver can obtain $D'(45\text{-}44) = D'(1) = bitget(123, 1) = 1$.

According to the array K_2, the receiver can obtain the following:

$$\begin{aligned} D'(49\text{-}44) &= D'(5) = bitget(90, 2) = 1, \\ D'(49\text{-}43) &= D'(6) = bitget(90, 1) = 0, \\ D'(51\text{-}44) &= D'(7) = bitget(167, 2) = 1, \\ D'(51\text{-}43) &= D'(8) = bitget(167, 1) = 1. \end{aligned} \tag{1}$$

According to the array K_3, the receiver can obtain the following:

$$\begin{aligned} D'(46\text{-}44) &= D'(2) = bitget(233, 3) = 0, \\ D'(46\text{-}43) &= D'(3) = bitget(233, 2) = 0, \\ D'(46\text{-}42) &= D'(4) = bitget(233, 1) = 1. \end{aligned} \tag{2}$$

Finally, the receiver can obtain the message "10011011" and then decrypt it to obtain the secret message.

4.2. Selecting the Thresholds. According to the embedding procedure, two thresholds T_1 and T_2 are set based on the middle range 128 to select the embedding bits. If the absolute value of the difference between the audio data and 128 is lower than T_1, one bit is embedded; if it is between T_1 and T_2, two bits are embedded; if it is more than T_2, three bits are used for embedding. Therefore, the thresholds T_1 and T_2 will influence the embedding capacity and the quality of stego audio. T_1 should not be too big. If T_1 is very big, the method may become the traditional LSB method. In the experiments, the embedding rate is set to 0.1bps. When $T_1 = 10$, 20, and 30, compare the value of signal-to-noise ratio (SNR) under different T_2, which is shown in Figure 5. As we can see in Figure 5, when $T_1 = 10$, the value of SNR grows slowly after $T_2 = 40$; when $T_1 = 20$, the growth of the value of SNR is slower after $T_2 = 40$, and when $T_1 = 30$, the value of SNR grows very slowly. Therefore, in the simulation experiment, we set $T_1 = 30$ and $T_2 = 40$.

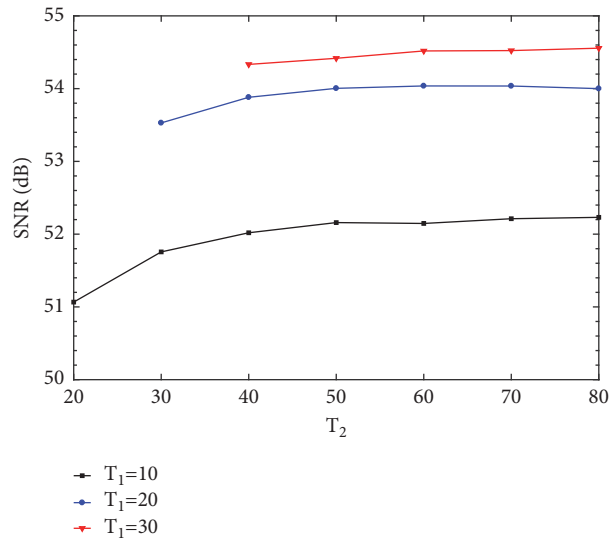


Figure 5: $T_1 = 10, 20, 30$, the value of SNR under different T_2.

4.3. The Embedding Rate. The embedding rate means that the number of the secret information can be embedded in a byte of cover audio data. For instance, in the LSB method, 1 byte of secret message should be embedded in 8 bytes of carrier data; thus the embedding rate is 12.5%. In the proposed method, the embedding location and embedding bits are selected adaptively, and thus the embedding rate is variable. When $R = 0$, it is 12.5%; when $R = 1$, it is 14.3%; when $R = 2$, it is 13.6%. Therefore, the average embedding rate is 13.5%. In Table 3, the embedding rate of the proposed method has been compared with the other LSB methods'.

4.4. The Embedding Capacity. The proposed method is an adaptive embedding algorithm and the embedding capacity is related to the thresholds T_1 and T_2, the size of wav-audio file, the sampling rate, and the quantization value. Because audio data is one of the most important covers in wireless communication, there is a large amount of audio data. Thereby the embedding capacity is considerable.

Table 3: Comparison of the embedding rate.

Method		Embedding Rate
Lowest Bit Coding [19]		12.5%
Variable Low Bit Coding [20]		12.7%
XOR of LSB [21]		12.5%
Sample Selection [22]		2.78%
Bit Selection [22]		12.5%
Based on the Interval R [23]	$R = 1$	6.25%
	$R = 2$	4.55%
	$R = 3$	3.45%
Fibonacci [24]		4.17%
the Proposed Method		13.5%

Table 4: The average value of SNR, MSE, and BER at different embedding rates.

Evaluating Indicator	Embedding Rate (bps)							
	0.005	0.01	0.05	0.1	0.2	0.3	0.4	0.5
SNR	67.873	64.698	57.318	54.203	51.338	49.609	48.427	47.483
MSE	0.0025	0.0051	0.0253	0.0492	0.0984	0.1481	0.1967	0.2459
BER	0.0024	0.0049	0.0248	0.0487	0.0979	0.1473	0.1955	0.2447

In variable low bit coding methods, the thresholds determine the embedding capacity. If most of the audio data are around 128, the embedding capacity will be very low, while in the proposed method, even though most of the audio data are around 128, the embedding capacity is higher than the LSB methods.

4.5. Imperceptibility Analysis. SNR, mean square error (MSE), bit error rate (BER), and the waveform are widely used to evaluate the imperceptibility of the audio steganography methods. Suppose x is the original audio, y is the stego audio, and N is the sampling point number of x and y.

SNR is calculated by (3). Higher SNR means better invisibility of the algorithm.

$$SNR = 10 \log_{10} \frac{\sum_{i=1}^{N} x(i)^2}{\sum_{i=1}^{N} [x(i) - y(i)]^2} \quad (3)$$

MSE is defined as the mean square error between the cover audio and the stego audio. MSE can measure the distortion in the audio and it is calculated by (4). Lower MSE means better performance of the algorithm.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} [x(i) - y(i)]^2 \quad (4)$$

BER describes the ratio of the modified bit number of the stego audio and the bit number of the original audio. Lower BER means better imperceptibility of the algorithm. BER is calculated by

$$BER = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} 1, & y(i) \neq x(i) \\ 0, & y(i) = x(i) \end{cases} \quad (5)$$

Table 4 shows the average value of SNR, MSE, and BER at different embedding rates. The results indicate that, at different embedding rates, SNR is kept at a high value but MSE and BER are kept at a low value. And in Figure 6, the proposed method has been compared with the LSB method [19], Variable Low Bit Coding method [20], and bit selection method [22] in SNR, MSE, and BER. In Figure 6(a), the SNR value of the proposed method is lower than [19] and higher than [20, 22]. This means that the invisibility of the proposed method is lower than [19] and better than [20, 22]. In Figure 6(b), the MSE value of the proposed method is the same as [19] and lower than [20, 22]. This means that the proposed method has better performance than [20, 22] and the same performance with [19]. In Figure 6(c), the BER value of the four methods is the same and this means that they have the same imperceptibility. In a word, the results demonstrate that the proposed method has good imperceptibility.

Figure 7 shows the waveforms of the original audio and the stego audio when the embedding rate is 0.1 bps. Obviously, the difference between the original audio and the stego audio is very tiny, which also means that the imperceptibility of the proposed method is satisfying.

5. Conclusion

An adaptive audio steganography method for wireless communication has been presented based on interval and the variable low bit coding in this paper. The interval R for embedding secret message into audio file and the thresholds T_1 and T_2 in variable low bit coding are used for selecting the embedding location and the embedding bits adaptively. The embedding capacity and the embedding rate are variable. The proposed method is simple and fast, which can

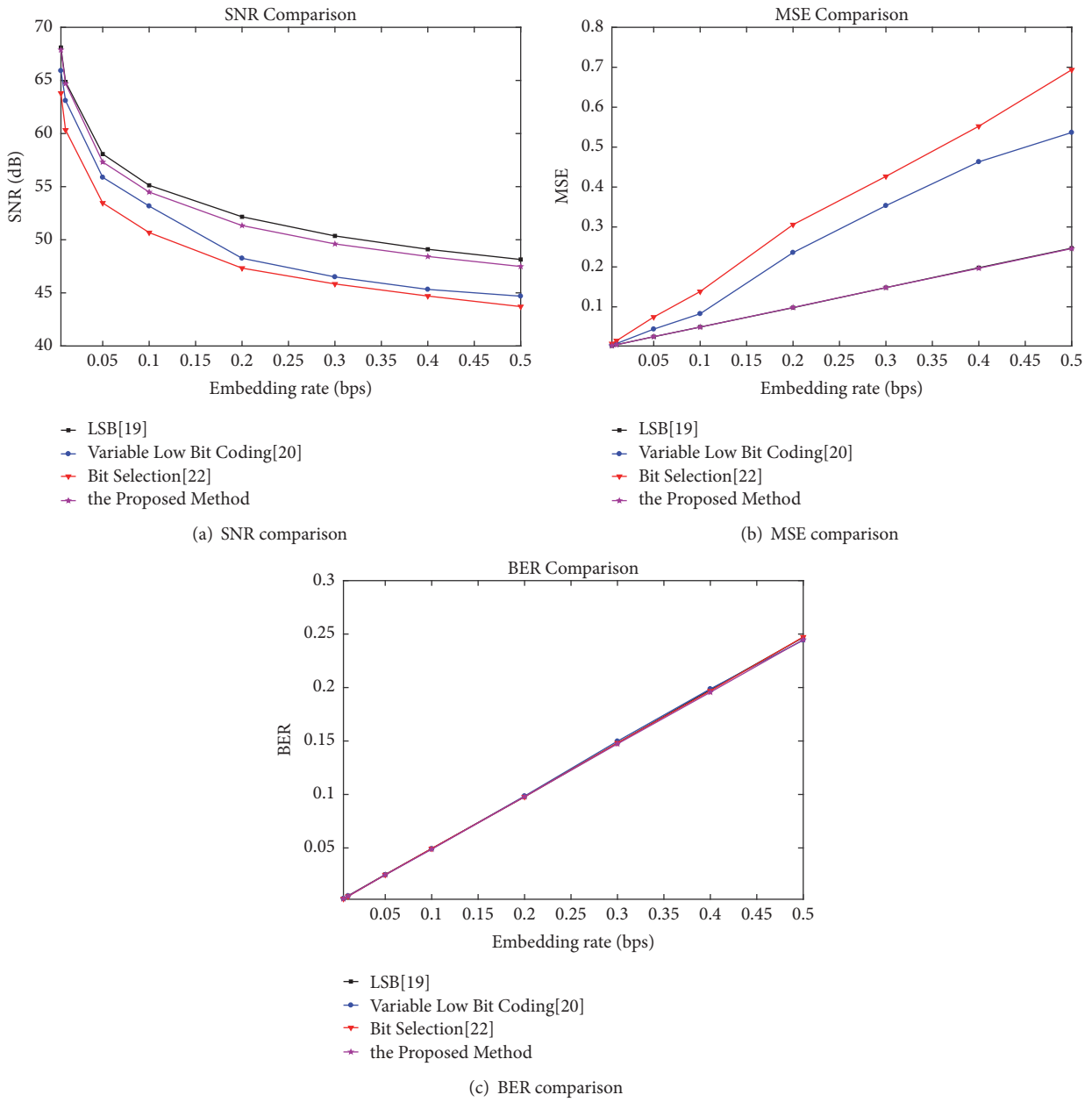


Figure 6: The comparison of SNR, MSE, and BER under different embedding rates between the proposed method and the other methods.

meet the instantaneity of wireless communication. In the future, we should investigate another low bit-rate speech codec applicable to wireless communication and design new steganographic algorithms.

Data Availability

The [software code and data] data used to support the findings of this study are available from the corresponding author upon request. Correspondence should be addressed to Yuling Liu: yuling_liu@126.com.

Conflicts of Interest

The four authors, Guojiang Xin, Yuling Liu, Ting Yang, and Yu Cao, all declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant no. 61103215 and Hunan Provincial Natural Science Foundation of China under Grant no. 2018JJ2062.

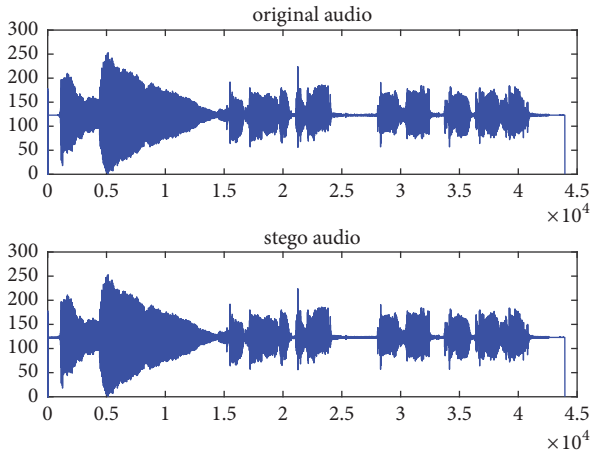


FIGURE 7: The waveforms of the original audio and the stego audio when the embedding rate is 0.1 bps.

References

[1] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[2] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection: A Survey," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 84–100, 2018.

[3] J. Han, R. Quan, D. Zhang, and F. Nie, "Robust object co-segmentation using background prior," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1639–1651, 2018.

[4] F. Peng, X.-Q. Gong, M. Long, and X.-M. Sun, "A selective encryption scheme for protecting H.264/AVC video in multimedia social network," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 1–19, 2016.

[5] N. Provos and P. Honeyman, "Hide and seek: an introduction to steganography," *IEEE Security & Privacy*, vol. 1, no. 3, pp. 32–44, 2003.

[6] X. R. Xiao, X. M. Sun, L. C. Yang et al., "Secure data transmission of wireless sensor network based on information hiding," in *Proceedings of Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, pp. 1–6, 2007.

[7] B. A. Bash, D. Goeckel, D. Towsley, and S. Guha, "Hiding information in noise: Fundamental limits of covert wireless communication," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 26–31, 2015.

[8] Y. L. Liu, H. Peng, and J. Wang, "Verifiable diversity ranking search over encrypted outsourced data," *CMC: Computers, Materials & Continua*, vol. 55, no. 1, pp. 37–57, 2018.

[9] Y. Cao, Z. Zhou, X. Sun, and C. Gao, "Coverless information hiding based on the molecular structure images of material," *Computers, Materials and Continua*, vol. 54, no. 2, pp. 197–207, 2018.

[10] J. Ye, J. Ni, and Y. Yi, "Deep Learning Hierarchical Representations for Image Steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2545–2557, 2017.

[11] R. H. Meng, S. G. Rice, J. Wang et al., "A fusion steganographic algorithm based on faster R-CNN," *CMC: Computers, Materials & Continua*, vol. 55, no. 1, pp. 1–16, 2018.

[12] J. J. Wang, M. O. Qian, D. X. Mei et al., "A robust audio watermarking algorithm based on LSB," *Journal of Guangxi University*, 2013.

[13] F. Djebbar, B. Ayad, H. Hamam, and K. Abed-Meraim, "A view on latest audio steganography techniques," in *Proceedings of the 7th International Conference on Innovations in Information Technology (IIT)*, pp. 409–414, Abu Dhabi, United Arab Emirates, April 2011.

[14] V. Bhat K, I. Sengupta, and A. Das, "An adaptive audio watermarking based on the singular value decomposition in the wavelet domain," *Digital Signal Processing*, vol. 20, no. 6, pp. 1547–1558, 2010.

[15] H. Morsy, J. Gluckman, A. Hussein, and F. Amer, "Block based steganography," in *Proceedings of the 9th European Conference on Information Warfare and Security 2010, ECIW 2010*, pp. 218–228, Greece, July 2010.

[16] Y. Diqun, W. Rangding, and Z. Liguang, "Quantization step parity-based steganography for MP3 audio," *Fundamenta Informaticae*, vol. 97, no. 1-2, pp. 1–14, 2009.

[17] P. Dutta, D. Bhattacharyya, and T. Kim, "Data hiding in audio signal: a review," *International Journal of Database Theory and Application*, vol. 2, no. 3, pp. 1–8, 2009.

[18] K. Geetha and P. V. Muthu, "Implementation of ETAS (embedding text in audio signal) model to ensure secrecy," *International Journal on Computer Science and Engineering*, vol. 2, no. 4, pp. 1308–1313, 2010.

[19] M. Wakiyama, Y. Hidaka, and K. Nozaki, "An audio steganography by a low-bit coding method with wave files," in *Proceedings of the 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP 2010*, pp. 530–533, Germany, October 2010.

[20] P. P. Balgurgi and S. K. Jagtap, "Audio Steganography Used for Secure Data Transmission," in *Proceedings of International Conference on Advances in Computing*, vol. 174 of *Advances in Intelligent Systems and Computing*, pp. 699–706, Springer India, New Delhi, 2012.

[21] P. P. Balgurgiand and S. K. Jagtap, "Intelligent processing: an approach of audio steganography," in *Proceedings of the International Conference on Communication, Information & Computing Technology*, pp. 1–6, 2012.

[22] M. Asad, J. Gilani, and A. Khalid, "An enhanced least significant bit modification technique for audio steganography," in *Proceedings of the 1st International Conference on Computer Networks and Information Technology (ICCNIT '11)*, pp. 143–147, IEEE, Pakistan, July 2011.

[23] S. Tang, Y. Jiang, L. Zhang, and Z. Zhou, "Audio steganography with AES for real-time covert voice over internet protocol communications," *Science China Information Sciences*, vol. 57, no. 3, pp. 1–14, 2014.

[24] H. Kumar and Anuradha, "Enhanced LSB technique for audio steganography," in *Proceedings of the 2012 3rd International Conference on Computing, Communication and Networking Technologies, ICCCNT 2012*, pp. 1–4, India, July 2012.

[25] J. Vimal and A. M. Alex, "Audio steganography using dual randomness LSB method," in *Proceedings of the 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT 2014*, pp. 941–944, India, July 2014.

[26] R. Tanwar, B. Sharma, and S. Malhotra, "A robust substitution technique to implement audio steganography," in *Proceedings of the 2014 International Conference on Reliability, Optimization and Information Technology, ICROIT 2014*, pp. 290–293, India, February 2014.

[27] P. Pathak, A. K. Chattopadhyay, and A. Nag, "A new audio steganography scheme based on location selection with enhanced security," in *Proceedings of the 2014 1st International Conference on Automation, Control, Energy and Systems, ACES 2014*, India, February 2014.

[28] M. Zou and Z. Li, "A wav-audio steganography algorithm based on amplitude modifying," in *Proceedings of the 10th International Conference on Computational Intelligence and Security, CIS 2014*, pp. 489–493, China, November 2014.

[29] L. Y. Xiang, Y. Li, W. Hao et al., "Reversible natural language watermarking using synonym substitution and arithmetic coding," *CMC: Computers, Materials & Continua*, vol. 55, no. 3, pp. 541–559, 2018.

[30] S. Tang, Q. Chen, W. Zhang, and Y. Huang, "Universal steganography model for low bit-rate speech codec," *Security and Communication Networks*, vol. 9, no. 8, pp. 747–754, 2016.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 5680264, 9 pages
https://doi.org/10.1155/2018/5680264

Research Article

A Model Based on Convolutional Neural Network for Online Transaction Fraud Detection

Zhaohui Zhang,[1,2,3] **Xinxin Zhou**,[1] **Xiaobo Zhang**,[1] **Lizhi Wang**,[1] **and Pengwei Wang**[1,2,3]

[1]*School of Computer Science and Technology, Donghua University, Shanghai 201620, China*
[2]*The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China*
[3]*Shanghai Engineering Research Center of Network Information Services, Shanghai 201804, China*

Correspondence should be addressed to Zhaohui Zhang; zhzhang@dhu.edu.cn

Received 20 May 2018; Accepted 16 July 2018; Published 6 August 2018

Academic Editor: Zhaoqing Pan

Using wireless mobile terminals has become the mainstream of Internet transactions, which can verify the identity of users by passwords, fingerprints, sounds, and images. However, once these identity data are stolen, traditional information security methods will not avoid online transaction fraud. The existing convolutional neural network model for fraud detection needs to generate many derivative features. This paper proposes a fraud detection model based on the convolutional neural network in the field of online transactions, which constructs an input feature sequencing layer that implements the reorganization of raw transaction features to form different convolutional patterns. Its significance is that different feature combinations entering the convolution kernel will produce different derivative features. The advantage of this model lies in taking low dimensional and nonderivative online transaction data as the input. The whole network consists of a feature sequencing layer, four convolutional layers and pooling layers, and a fully connected layer. Verifying with online transaction data from a commercial bank, the experimental results show that the model achieves excellent fraud detection performance without derivative features. And its precision can be stabilized at around 91% and recall can be stabilized at around 94%, which increased by 26% and 2%, respectively, comparing with the existing CNN for fraud detection.

1. Introduction

Using wireless mobile terminals has become the mainstream of Internet transactions, which can verify the identity of users by passwords, fingerprints, sounds, and images. The fraudster can collect users' information, such as ID, password, age, occupation, and other information, and logs in various trading systems as normal users to complete fraud. This kind of fraudulent behavior has been very common today in the rapid development of information technology, and it brings great losses to users, businesses, and society. Moreover, once these identity data are stolen, traditional information security methods will not prevent online transaction fraud.

Banks and major financial institutions provide a wide range of services, but the fraud is widespread in many financial transactions provided by these institutions. More services will generate more users' data, which provides a great possibility for fraudsters to steal users' information to complete fraud. How to detect fraudulent transaction accurately and instantly has become an urgent financial security problem for all financial institutions, including banks. The traditional expert rule system is applied to most fraud detection areas. These expert rule systems are based on the existing industry experience rules, which can detect the occurred fraudulent patterns and the existing fraud behaviors. However, online fraud transactions are very different from traditional transactions, so the traditional expert rule system is incapable of detecting and intercepting online fraud transactions effectively. The fraudulent transaction in this paper means that the fraudster embezzles the legitimate user's information and enters the trading system as a normal user.

A variety of machine learning and deep learning models are gradually applied in detecting fraud. Compared with the traditional rule system, the advantage of machine learning is its ability to use a large number of complex data to characterize some financial phenomena that are difficult to be found by

traditional methods. Various models used for financial fraud detection include neural networks, deep neural networks, random forests, logistic regression, SVM [1–6], and so on. On the one hand, most of the existing models are applied to credit card fraud detection. On the other hand, credit card transactions differ from online transactions, so these models cannot be entirely suitable for online transactions.

The pros of neural networks and deep learning are that they can fully approximate any complex nonlinear relationships, strong robustness, and fault tolerance and find optimized solutions at high speed. It has an outstanding performance in image recognition [7–9], video processing [10], natural language processing [9], and other fields. But when dealing with the structured data, especially online transaction data, neural networks, and deep learning models have poor performance. Because the available dimensions of transaction data are often very limited, some massive features derived from most existing models and prior knowledge of the industry do not contribute to the learning [2, 11]. Consequently, this paper constructs a CNN model based on the feature sequencing for Internet transaction fraud detection. And compared with the existing CNN models, our model can achieve a better performance only by using the transaction data raw features as training.

The rest of this paper has been structured as follows. Section 2 introduces the application and effectiveness of existing machine learning and deep learning methods in financial fraud detection. Then, Section 3 describes the process of constructing a convolutional neural network model based on feature rearrangement. Later, we set up experiments and evaluate its performance in Section 4. Finally, Section 5 summarizes this paper and discusses future work.

2. Related Work

Antifraud system is the first line of defense for financial institutions. So, the antifraud system is widely adopted in banking, insurance, law, business administration, and other fields.

With the maturity of many machine learning algorithms and deep learning algorithms, they have been successfully used in image detection, text processing, and other fields. Gurusamy, R. and Subramaniam, V. [8] proposed a new method for the denoising, extraction and tumor detection on MRI images. They used a variety of machine learning algorithms to build brain image recognition systems to aid in medical diagnosis and medical evaluation. These methods include CNN and SVM, and these algorithms had a good result when they were applied in this scenario. Chengsheng Yuan [9] used a combination of CNN and SVM to build a live fingerprinting model and achieved good results. They use CNN for feature extraction and SVM for classification.

Machine learning and deep learning models are also used in the field of financial fraud detection. S.Ghosh and D.L.reilly [12] used the neural network algorithm to construct a transaction fraud detection system, which was verified in the credit card transaction data of Mellon Bank and applied to the actual transaction system in 1994. The model built by Bayesian belief network was also used for fraud detection. Sam Maes et al. [1] built a transaction fraud model using Bayesian belief networks. When applied to experimental data sets, it was found that Bayesian belief networks had higher recognition accuracy than neural networks. In some scenarios, constructing the recognition models with a single algorithm is inferior to combinatorial algorithms. V. Hanagandi et al. [13] constructed a credit card fraud scoring system by combining a radial basis function network with a density-based clustering algorithm. According to users' historical records, this system would generate fraud scores for decreasing credit fraud. It is hard to determine the network topology when using neural networks to build this model. Raghavendra Patidar et al. [14] made it easier to build a fraud detection model, by using a genetic algorithm to calculate this neural network topology, including the numbers of hidden layers and the numbers of nodes.

Existing neural networks often require high-dimensional data as input, which means that it is hard to obtain high dimensionality and strong availability transaction. The common solution is making derivative features for consumer behavior patterns based on industry experience, which reflects the users' behavior habits. The exploration of different legal consumer behavior patterns and fraudster behavior patterns is critical in fraud detection. A. I. Kokkinaki et al. [15] described legal consumer's transaction habit with a decision tree and Boolean logic method. Besides, they used clustering methods to analyze the distinction between normal or abnormal transaction. Kang Fu et al. [2] proposed using trading entropy and other derivative features based on industry experience to characterize user trading action. The average transaction amount, total amount, the difference between the current transaction amount and the average transaction amount, trading entropy, and other features generated from the raw data of the fixed time window were used as model input data. The trading entropy is a novel feature used to describe the user's transaction behavior. It can describe the relationship between the user's transaction amount and the total transaction amount over a period of time. These derived features can better reflect the characteristics of the user's transaction behavior under certain conditions.

Besides the above method of analyzing user's behavior from a data perspective, some scholars analyze user's abnormal behavior from the perspective of system behavior. Zhang and J. Cui proposed a method to discover user's abnormal behavior from a system perspective. Zhang Z., Ge L. [16] et al. proposed an effective way to solve user behavior anomalies through system behavior reconstruction.

In addition to neural network algorithms, logistic regression, support vector machines, random forest algorithms [3, 4], hidden Markov models [17, 18], and adversarial learning methods [19] are also widely applied in constructing credit fraud detection model. Most of these existing model algorithms are based on credit card transactions. Credit card transaction and online transaction are different in terms of transaction methods, transaction characteristics, trader behaviors, etc. [20]. The models based on credit transaction are not fully applied to the online transaction.

Most of the existing fraud detection models are constructed for the credit card transaction, which is not fully

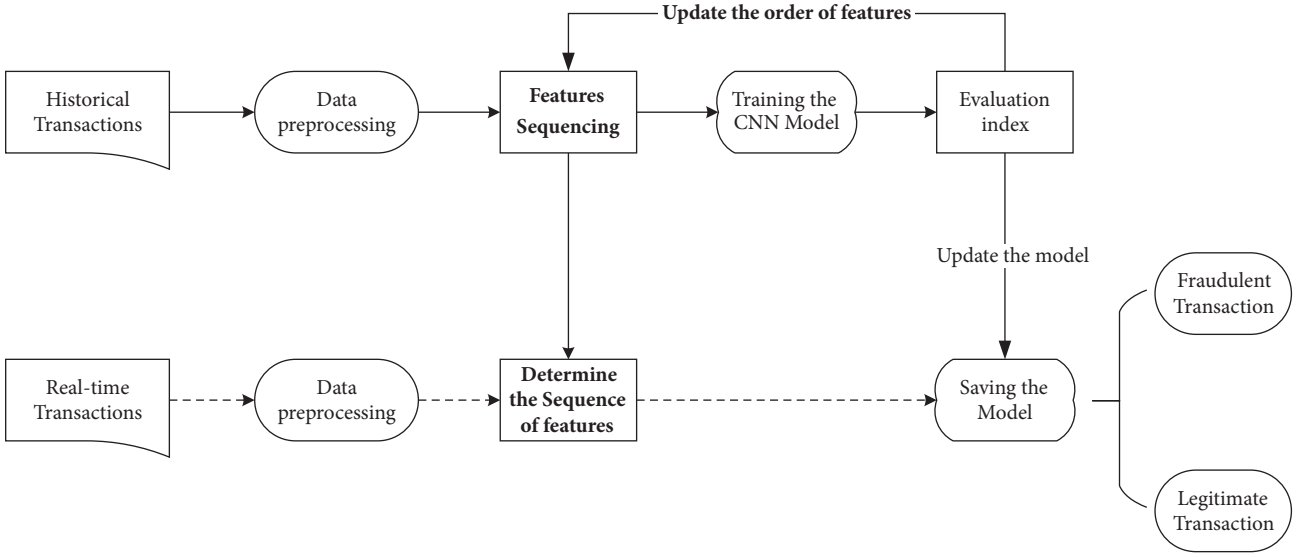


Figure 1: The structure of the model.

applicable to online transaction fraud detection. Most of the neural network models will need a large number of derivative variables in feature engineering; thus, these models cannot be applied to low dimensional transaction data. Therefore, the CNN based on feature sequencing is proposed to solve the problem of online transaction fraud detection.

3. CNN Based on Feature Sequencing

3.1. Framework. This paper builds the model based on the CNN to directly use the low dimensional raw features as the input of the model. The feature sequencing layer is added to automatically optimize the sequence of features. This approach can save variable derived time, take advantage of the CNN, learn the derivative features that are beneficial to the classification results, and reduce the interference of human experience with the model. In the fraudulent transaction, there are a lot of features and trading patterns which are not found, and the purpose of reducing the interference is to let the CNN learn the transaction characteristics and trading mode as much as possible.

The overall structure of the model is divided into two parts: the model training part and the transaction detection part. The training part of the model is divided into two parts: the feature sequencing layer and the CNN. The increased feature sequencing layer is used to optimize the sequence of transaction features. First, the historical data is cleaned and so on, then put data into the feature sequencing layer, and the model effect is tested by training the CNN model, and the feature sequence order is modified by the effect feedback. In update time, we can find out the optimal sequence mode by fixed feature permutation times. When the real-time data enters the model, the data features are sorted by the order of the feature, and then the training model is judged (Figure 1).

3.2. Feature Sequencing Layer. Transaction data consists of multidimensional features, and there is no direct connection between multidimensional features, so multidimensional attributes can be arranged randomly. If transaction data put into various model algorithms in the form of one-dimensional variables, the arrangement, and combination of different attributes will not affect the physical meaning of the record. But different permutations and combinations will affect the results of the model. This is essentially the same as image, speech, text, and other data. Take image data as an example: although the image can be translated, rotated, flipped, etc., it remains invariant during the conversion process, but the essence of the image is composed of ordered pixels. The position of these pixels is not allowed to change; otherwise, the inherent information carried by the image will change.

We use a 5-tuple to describe transaction data.

Definition 1. A transaction data M is a five-tuple composed of transaction features, feature arrangement state, position exchange operation, feature initial arrangement state, and feature final arrangement.

Formally as follows: $M = (Q, \Sigma, \delta, q_0, F)$

Q: a finite set representing transaction features

Σ: a finite set representing the different arrangements of transaction features

δ: exchange operations between transaction features

q_0: $q_0 \in Q$, transaction data features initial state

F: $q_0 \times \delta \longrightarrow F$, transaction data features state finally arranged

3.3. Feature Sequencing. Each fraudulent transaction consists of multiple transaction features. The arrangement of these trading features does not affect the physical meaning of the transaction, but different feature arrangements will have a different effect on the model after the convolution process. This is why we add the feature sequencing layer into the

Input: The weight matrix A, the rows of the weight matrix $A_1A_2 \ldots A_n$, initial state of the input set Q, list of model accuracy rates Acc, auxiliary sequence $c_1c_2 \ldots c_n$(c_j is the number of rows below a_j, satisfy the first condition $0 \le c_j \le j$) and $o_1o_2 \ldots o_n$(o_j control the direction of c_j change)
Output: The best permutation of weight matrix.
1: $c_j \longleftarrow 0$
2: $o_j \longleftarrow 1(1 \le j \le n)$
3: $Acc \longleftarrow [\,]$
4: Access matrix A
5: $j \longleftarrow n$, $s \longleftarrow 0$ (s is the number of c_k satisfying $k > j$ and $c_k = k - 1$)
6: $q \longleftarrow c_j + o_j$:
7: **if** $q < j$ **then**
8: go to 19
9: **end if**
10: **if** $q = j$ **then**
11: go to 14
12: **end if**
13: $A_{i-c_j+s} \longleftrightarrow A_{j-q+s}$, $c_j \longleftarrow q$, go to 4
14: **if** $j = 1$ **then**
15: input $Q * A$ to model and calculate the *accuracy*
16: *Acc.append*(*accuracy*)
17: **if** *len*(*Acc*) == $n!$ **then**
18: return max(*Acc*) and A that max(*Acc*) corresponding
19: **end if**
20: **else**
21: $s = s + 1$
22: **end if**
23: $o_j = -o_j$, $j = j - 1$, go to 6

Algorithm 1: Find the best permutation of input features.

model. Each transaction feature has the potential to exist anywhere, in order to ensure that all the arrangements of features can be taken, so the nodes between the initial input layer and the final input layer are fully connected (Figure 3), but the connection weight changes during each iteration.

The reason why convolutional neural networks can accurately classify images is that they automatically find important classification features in a brute-force, mass-data fashion. This is both an advantage of CNN being able to identify precisely, and a disadvantage of it not being able to identify the location of the image effectively. So, whether the image is complete or not, we have reason to believe that CNN can identify the desired object on the map. For example, when a nose and eyes misplaced the face picture, CNN will still determine it as a normal face picture. A location problem similar to this problem also exists in the context of using CNN for transaction data identification.

In the model constructed in this paper, one-dimensional feature vector into the model, and the convolution layer is processed with one-dimensional convolution kernel feature vector. In the process of convolution, the principle is that as in image processing, information extraction is performed on partial features. Figure 2 depicts a feature vector convolution transformation process, such as 1×2 convolution kernel. The process of convolution is to select two adjacent features for convolution and generate derivative features. For the multivariate feature vectors, the sequence of the features is

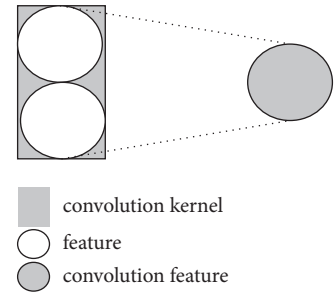


Figure 2: The convolution procedure process between two features.

different. Naturally, the results of the single-layer convolution are directly affected. The CNN, in turn, contains multiple convolutional layers. After the layer convolution, the effect will lead to the identification of the entire model.

This paper constructs a CNN model based on feature sequencing which adds a feature sequencing layer before the input layer. Firstly, the network structure of convolution layer, pooling layer, and fully connected layer are determined by data features, and the optimal permutation order of all permutations is determined by the feedback of model results. Then the model parameters are trained with the input that fixes the permutation. The time complexity of Algorithm 1 that aims at finding the optimal sequencing of all features is

Input: The weight matrix A, the rows of the weight matrix $A_1 A_2 \ldots A_n$, list of model accuracy rates Acc, two integers i, j, initial state of the input set Q, the iteration times that you want to do m
Ouput: The best permutation of weight matrix.
1: $A \longleftarrow n - DimensionalIdentityMatrix$
2: $Acc \longleftarrow [\,]$
3: $i \longleftarrow random(\,)$
4: $j \longleftarrow random(\,)$ $(0 \leq j \leq n-1)$
5: Access matrix A
6: $A_i \longleftrightarrow A_j$
7: $Q \longleftarrow Q * A$:
8: input Q to model and calculate the accuracy
9: $Acc.append(accuracy)$
10: **if** $len(Acc) == m$ **then**
11: return max(Acc), A that max(Acc) corresponding
12: **end if**

Algorithm 2: Find the optimal arrangement within a fixed number of times.

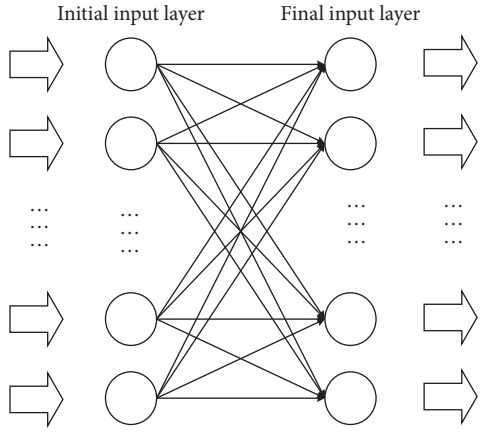


Figure 3: Feature sequencing layer.

O(n!). If the transaction data has more feature dimensions, the time complexity of the algorithm will be very high, which is not conducive to the construction and training of the model. Therefore, we also construct Algorithm 2 that randomly transforms feature arrangements and finds the optimal arrangement within a specified number of times and a fixed number of iterations. The algorithm can subjectively set the number of model iterations and can find better feature sequencing in a short period of time relatively.

Figure 2 shows the network structure of the feature sequencing layer. The number of features selected in the model is n, and the number of all arrangements of features is m. The initial input layer is the original input and final input layer is the input features after the sequence transformation. The order of the input features is changed through the transformation of initial input layer and final input layer connection weight matrices. Set the connection weight matrix A and initialize the connection matrix to A_0. Each time a matrix row is transformed, the connection weight matrix for the next iteration is generated.

We arrange the data features in the initial state Σ as a one-dimensional vector $\Sigma_0 = [x_0, x_2, x_3, \ldots, x_n]$. The position transform operation can be expressed as the product of Σ and the connection matrix A. In special cases, the connection weight matrix is as shown in A_0, our model will degenerate into a regular CNN, and the original feature input order will not be changed.

$$A_0 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{bmatrix} A_1 = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{bmatrix} \cdots\cdots A_{m-1} = \begin{bmatrix} 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix} \tag{1}$$

$$\Sigma_i = \Sigma_{i-1} \times A_i \tag{2}$$

If the feature sequencing layer connection matrix is as shown in formula (1), the first step of the transformation process is as follows:

$$\Sigma_1 = \Sigma_0 \times A_1 = [x_1, x_2 x_3, \cdots, x_n] \times \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{bmatrix} = [x_2, x_1 x_3, \cdots, x_n] \tag{3}$$

3.4. CNN Network Structure with Feature Sequencing Layer. Compared with the existing CNN model, the network structure of this model has a feature sequencing layer. The network

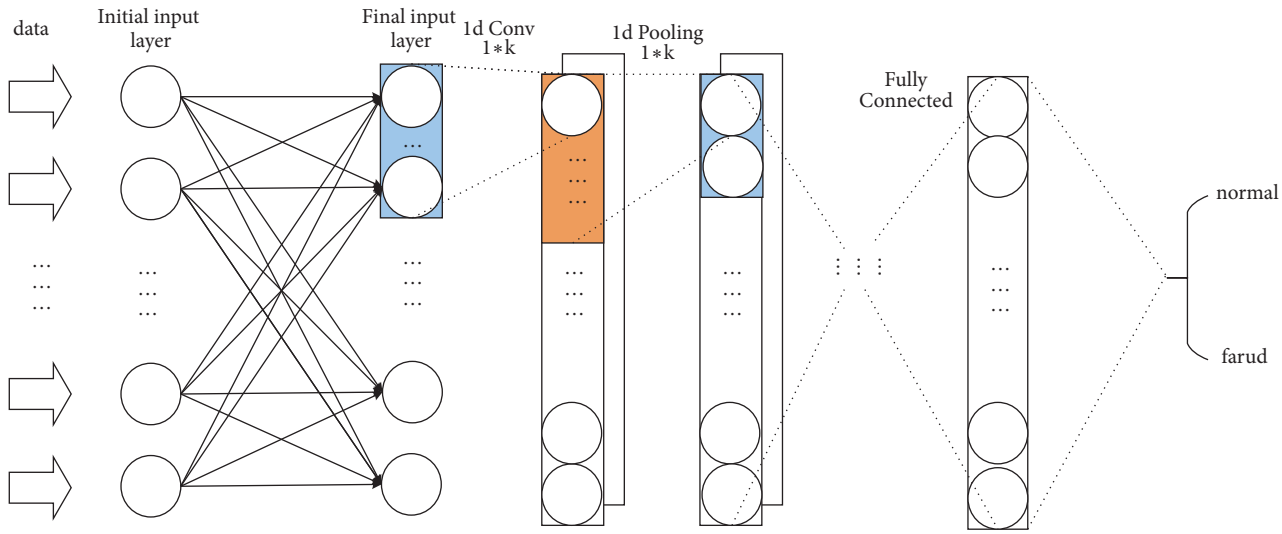


Figure 4: Overall network structure.

structure is designed to ensure that it can be applied to network transaction data and quickly identify online transaction data. The whole network consists of a feature sequencing layer, four alternating distribution convolutional layers and pooling layers, and a fully connected layer (Figure 4).

The feature sequencing layer is the order arrangement processing to the input features since convolution of the different order feature input layers results in different effects of the model. The convolutional layer function is to extract the local feature of the input data; in this scenario, we can understand that the convolutional layer will automatically derive new features based on the input features. These new derivative features, although we do not explain their physical meaning, are indeed helpful to the classification of the model. The pooling layer joins the features of the adjacent areas together into a single higher level feature that reduces the redundancy of the data. The fully connected layer plays the role of the final classification. For different input data, the number of nodes in each layer of the network varies. In my experiment, the channels are two, and the number of nodes in the fully connected layer is 144.

For each trained network model, we save the current model and compare it with the previous model. If the current model works better than the previous model, we replace the previously saved model with the current model so that these trained models can be directly applied to the detection of real-time trading data.

4. Experiment Verification and Analysis

4.1. Dataset. The experimental data of this paper comes from a commercial bank B2C online transaction data; a total of about 5 million transaction data are extracted for the experiment. The positive sample is approximately 33 times that of the negative sample; each transaction record has 62 dimensions. All transaction data has a time span of 6 months, and we take two samples to construct a more balanced experimental dataset of positive and negative samples. In order to ensure the sequential continuity of transaction data, we use one-month data batches as experimental data. When the model is trained and validated, we divide the data of one month, about 500,000, into training sets and test sets, and the ratio is about 3:1. In order to ensure the consistency and availability of data, we have done routine processing such as data cleaning, data transformation, and data reduction. All comparative tests were performed on the same dataset.

Based on the feature engineering methods in the literature [17, 18] and the statistical analysis of the raw data, we find that the characteristics of user transaction behavior, such as time, amount, and location, and other derived features are very significant in the fraud detection model. These users' data are also information that fraudsters usually steal. Combined with the results of the data cleaning, we select 8D features such as transaction ID (because of the data confidentiality, this paper does not mention all the data dimensions) as input to the model.

4.2. Model Validation. This paper uses accuracy, precision, recall, and F_1 score to evaluate the effectiveness of this model. In this paper, the CNN model based on feature sequencing is compared with the existing CNN[2] and BP neural network in the same data set (Figures 5–11). All comparative tests were performed on the same dataset. From the following six groups of test results, we can deduce that our model's precision rate can be stabilized at around 91% and recall rate can be stabilized at around 94%, which increased by 26% and 2%, respectively, compared with the existing CNN for fraud detection. At the same time, we also compared with the traditional BP neural network with two hidden layers. The effect of each index of this model has been greatly improved compared with the traditional BP neural network.

Firstly, this paper makes a detailed analysis and judgment on the data set using traditional BP neural network and optimizes the model by adjusting the number of nodes and

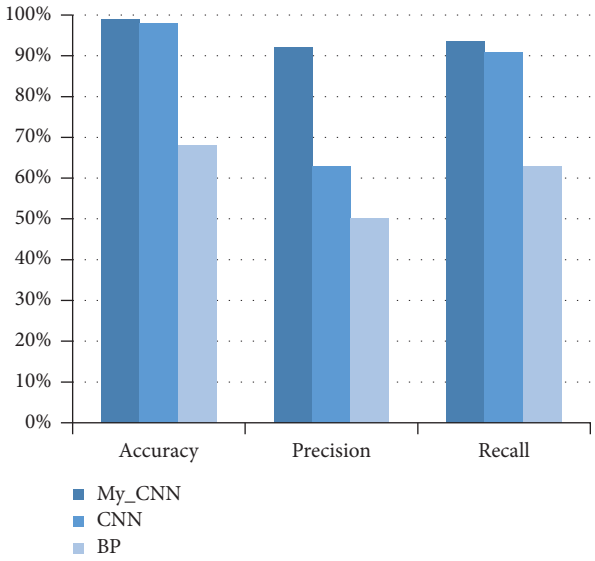


Figure 5: Different performances of three models on Set_1.

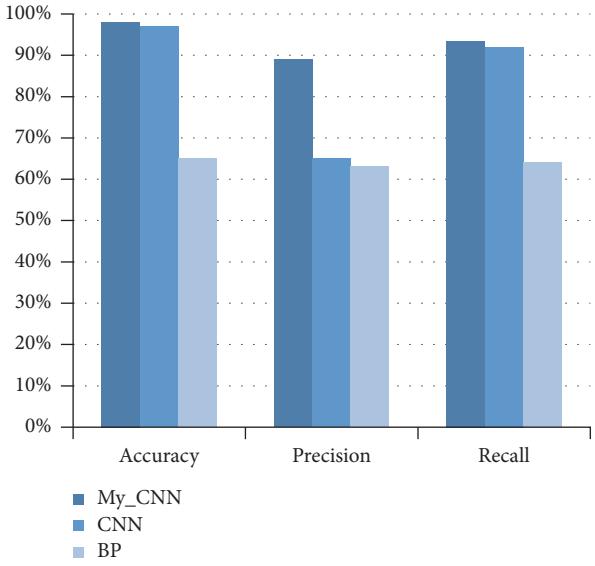


Figure 6: Different performances of three models on Set_2.

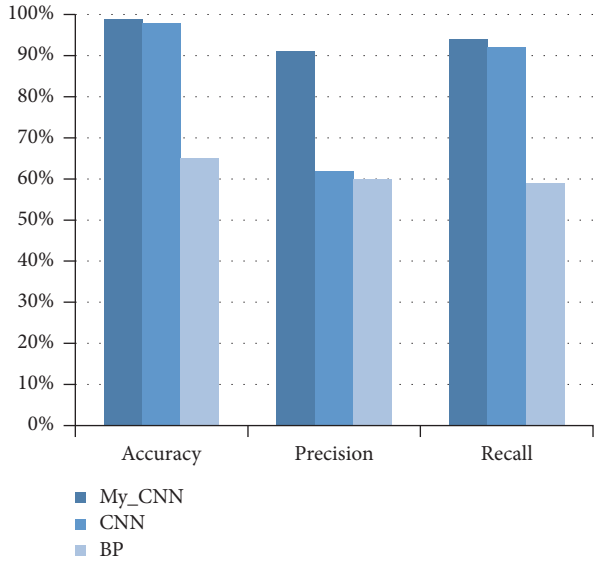


Figure 7: Different performances of three models on Set_3.

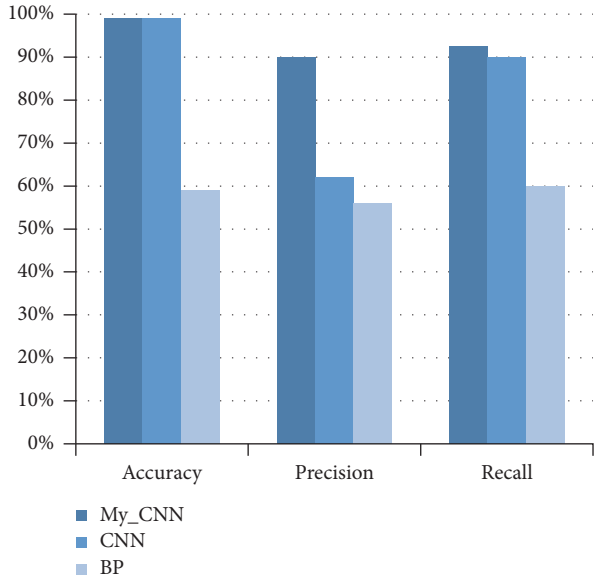


Figure 8: Different performances of three models on Set_4.

the number of layers of the network. However, the final model structure is not ideal, so this result is only used as the basic comparison work of our paper.

In order to verify whether the different sequencing of the features has an optimal effect on the model, we use Algorithm 2, set m=10, and record the performance of the 10 times (Figure 8). It can be seen from the experimental results that different feature sequencing methods have an effect on the model's results. The best experimental results in the ten sequences (the eighth) are better than the original ones, and if the computational capabilities allow, we will be able to find out more superior feature sequencing (Figure 12).

In terms of time performance, the same monthly data set was used to experiment: the BP neural network was significantly faster than the two convolutional neural networks; compared with the two convolutional neural networks, the epoch was 1, and the batch size was 1000. The feature sequencing convolutional neural network has a training time of 65 seconds for once feature arrangement. Model training time with ten times features arrangements is 752 seconds. The traditional CNN training time is 352 seconds without the feature derivative work. If we add the processing time of the derived features, the time performance of the model is far worse than our model.

The CNN model based on feature sequencing is compared with the existing convolutional neural network. The experiment shows that the model constructed in this paper is superior to the existing CNN model in the effect of each indicator and does not need to do a large number of derivative

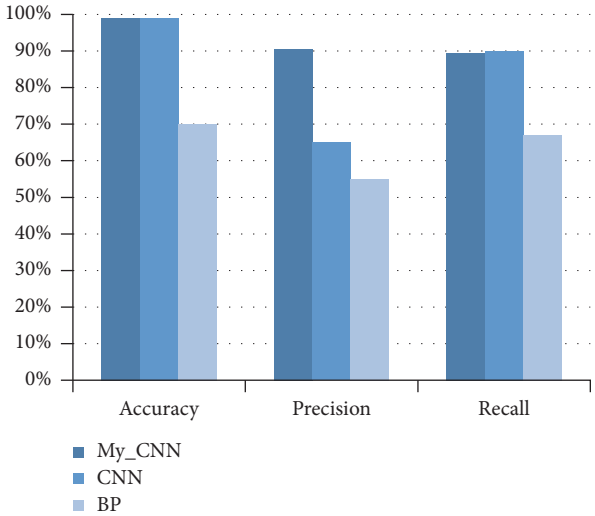


Figure 9: Different performances of three models on Set_5.

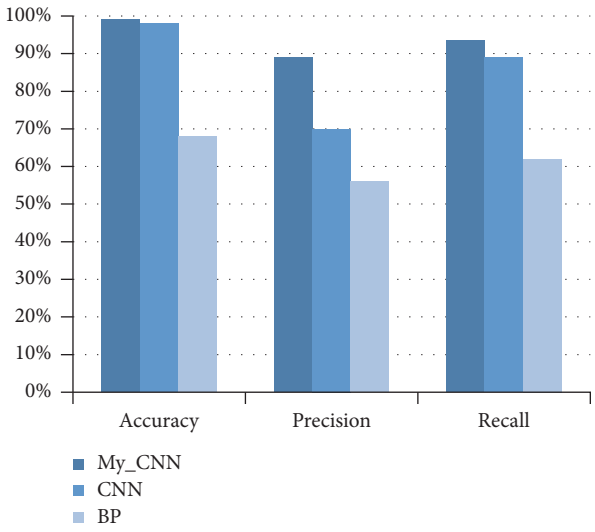


Figure 10: Different performances of three models on Set_6.

variables in the data preprocessing part. We use the raw 8-dimensional feature as input, which saves the time for model construction and solves the problem that low-latitude data is not conducive to building a network fraud detection model.

5. Conclusion and Discussion

The CNN model based on feature rearrangement constructed in this paper has an excellent experimental, performance with a good stability. The model needs neither high dimensional input features nor derivative variables and can find a relatively good ordered arrangement of input within a certain number of times. Compared with most existing CNN model, this model saves much calculation time of the derived variables, which makes the design and adjustment process of the model quick and easy. And there is a higher level of availability

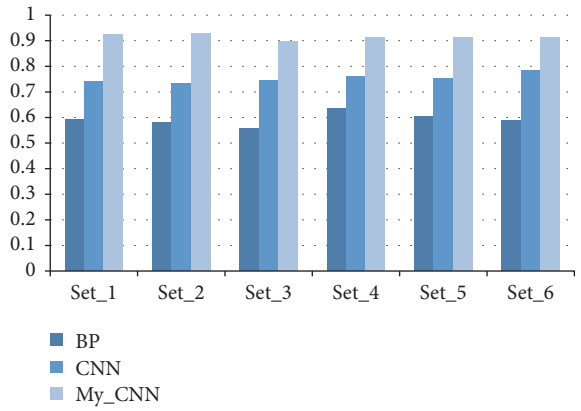


Figure 11: Different F_1 scores of the three models on various sample sets.

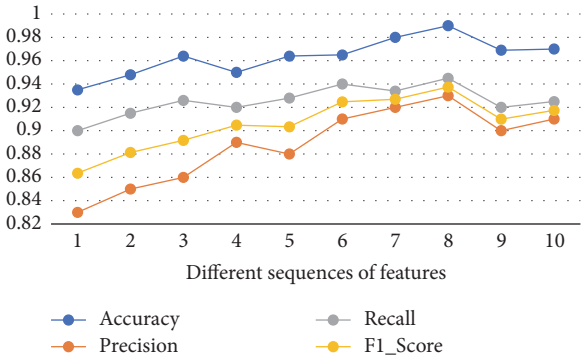


Figure 12: Different effects of the model with different input features sequencing.

in an environment where online transactions require rapid response and accurate identification.

In the future work, we will pay more attention to the discovery of sequence characteristics of transactions. In addition, we will apply the LSTM algorithm to make our model have a good memory of the trader's behavior in order to discover more fraudulent transactions accurately. For different sequences of features, the model has different effects. And from this point, we will continue to discover the relationships of data characteristics and find out the characteristic combinations that have an important influence on the model by controlling and transforming the size of the convolution kernel.

Data Availability

The data support for this study is derived from a bank's internal data and will not be provided to those who have not signed a confidentiality agreement with the bank. So, these data are not open.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61472004 and 61602109), Shanghai Science and Technology Innovation Action Plan Project (no. 16511100903), and Key Laboratory of Embedded System and Service Computing of Tongji University of Ministry Education (2015).

References

[1] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit Card Fraud Detection Using Bayesian and Neural Networks," in *Proceedings of the 1st International Naiso Congress on Neuro Fuzzy Technologies*, pp. 261–270, 2002.

[2] K. Fu, D. Cheng, Y. Tu, and L. Zhang, "Credit Card Fraud Detection Using Convolutional Neural Networks," in *Neural Information Processing*, vol. 9949 of *Lecture Notes in Computer Science*, pp. 483–490, Springer International Publishing, 2016.

[3] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: a comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.

[4] N. Carneiro, G. Figueira, and M. Costa, "A data mining based system for credit-card fraud detection in e-tail," *Decision Support Systems*, vol. 95, pp. 91–101, 2017.

[5] W. Yin, K. Kann, M. Yu, and H. Schtze, "Comparative Study of Cnn and Rnn for Natural Language Processing," 2017, https://arxiv.org/abs/1702.01923.

[6] R. Brause, T. Langsdorf, and M. Hepp, "Neural data mining for credit card fraud detection," in *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '99)*, pp. 103–106, November 1999.

[7] H. Shin, H. R. Roth, M. Gao et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[8] R. Gurusamy and V. Subramaniam, "A machine learning approach for MRI brain tumor classification," *Computers, Materials and Continua*, vol. 53, no. 2, pp. 91–109, 2017.

[9] C. Yuan, X. Li, Q. M. J. Wu, J. Li, and X. Sun, "Fingerprint Liveness Detection from Different Fingerprint Materials Using Convolutional Neural Network and Principal Component Analysis," *Computers Materials & Continua*, vol. 53, no. 4, pp. 357–372, 2017.

[10] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive Fractional-Pixel Motion Estimation Skipped Algorithm for Efficient HEVC Motion Estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[11] A. Correa Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," *Expert Systems with Applications*, vol. 51, pp. 134–142, 2016.

[12] S. Ghosh and D. L. Reilly, "Credit card fraud detection with a neural-network," in *Proceedings of the 27th Hawaii International Conference on System Sciences*, vol. 3, pp. 621–630, Wailea, Hawaii, USA, 1994.

[13] V. Hanagandi, A. Dhar, and K. Buescher, "Density-based clustering and radial basis function modeling to generate credit card fraud scores," in *Proceedings of the IEEE/IAFE 1996 Conference on Computational Intelligence for Financial Engineering, CIFEr*, pp. 247–251, March 1996.

[14] R. Patidar and L. Sharma, "Credit Card Fraud Detection Using Neural Network," in *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 1, pp. 32–38, Citeseer Press, 2011.

[15] A. I. Kokkinaki, "On atypical database transactions: Identification of probable frauds using machine learning for user profiling," in *Proceedings of the 1997 IEEE Knowledge & Data Engineering Exchange Workshop, KDEX*, pp. 107–113, November 1997.

[16] Z. Zhang, L. Ge, P. Wang, and X. Zhou, "Behavior Reconstruction Models for Large-scale Network Service Systems," *Peer-to-Peer Networking and Applications*.

[17] A. Gupta, D. Kumar, and A. Barve, "Hidden Markov Model based Credit Card Fraud Detection System with Time Stamp and IP Address," *International Journal of Computer Applications*, vol. 166, no. 5, pp. 33–37, 2017.

[18] A. Srivastava, A. Kundu, S. Sural, and A. K. Majumdar, "Credit card fraud detection using hidden Markov model," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 1, pp. 37–48, 2008.

[19] M. F. Zeager, A. Sridhar, N. Fogal, S. Adams, D. E. Brown, and P. A. Beling, "Adversarial learning in credit card fraud detection," in *Proceedings of the 2017 Systems and Information Engineering Design Symposium, (SIEDS '17)*, pp. 112–116, IEEE Press, USA.

[20] S. J. Chen and H. Yuan, "Research on the Common Characteristics of Online Transaction Fraud," in *Journal of Chongqing University of Posts and Telecommunications*, vol. 27, pp. 96–102, 5 edition, 2015.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 4943509, 9 pages
https://doi.org/10.1155/2018/4943509

Research Article

TR-IDS: Anomaly-Based Intrusion Detection through Text-Convolutional Neural Network and Random Forest

Erxue Min,[1] **Jun Long**,[1] **Qiang Liu**,[1] **Jianjing Cui**,[1] **and Wei Chen**[2]

[1]*College of Computer, National University of Defense Technology, Changsha 410073, China*
[2]*School of Computer Science, University of Birmingham, Birmingham, British B15 2TT, UK*

Correspondence should be addressed to Erxue Min; mex338@qq.com

Received 3 May 2018; Accepted 24 June 2018; Published 5 July 2018

Academic Editor: Zhaoqing Pan

As we head towards the IoT (Internet of Things) era, protecting network infrastructures and information security has become increasingly crucial. In recent years, Anomaly-Based Network Intrusion Detection Systems (ANIDSs) have gained extensive attention for their capability of detecting novel attacks. However, most ANIDSs focus on packet header information and omit the valuable information in payloads, despite the fact that payload-based attacks have become ubiquitous. In this paper, we propose a novel intrusion detection system named TR-IDS, which takes advantage of both statistical features and payload features. Word embedding and text-convolutional neural network (Text-CNN) are applied to extract effective information from payloads. After that, the sophisticated random forest algorithm is performed on the combination of statistical features and payload features. Extensive experimental evaluations demonstrate the effectiveness of the proposed methods.

1. Introduction

Due to the advancements in Internet, cyberspace security has gained increasing attention [1, 2], which has encouraged many researchers to design effective defense systems called Network Intrusion Detection Systems (NIDSs). Currently, existing intrusion detection techniques fall into two main categories: misuse-based detection (also known as signature-based detection or knowledge-based detection) and anomaly-based detection (also known as behavior-based detection). Misuse-based detection systems extract the discriminative features and patterns from known attacks and hand-code them into the system. These rules are compared with the traffic to detect attacks. They are effective and efficient for detecting known type of attacks and have a very low False Alarm. Therefore, Misuse-based detection systems are currently the mainstream NIDSs and some sophisticated ones have been deposited in real scenarios, e.g., snort [3]. However, misuse detection systems require updating the rules and signatures frequently and they are incapable to identify any novel or unknown attacks. In recent years, anomaly-based network intrusion detection systems (ANIDSs) have attracted much attention for their capability of detecting zero-day attacks. They adopt statistical methods, machine learning algorithms, or data mining algorithms to model the pattern of normal network behavior and detect anomalies as deviations from normal behavior.

Various algorithms have been proposed to model network behavior and detect anomaly flows, including artificial neural networks [4], fuzzy association rules [5], Bayesian network [6], clustering [7], decision trees [8], ensemble learning [9], support vector machine [10], and so on [11, 12]. However, these methods mostly exploit the information in packet headers or the statistical information of entire flows and fail to detect the malicious content (e.g., SQL injection, cross-site scripting, and shellcode) in packet payloads. Classic processing methods for payloads can be divided into two categories. The first category requires prior knowledge of protocol formats, which cannot be applied to unknown protocols. The second category does not require expert domain knowledge; instead, they calculate some statistical features or conduct N-gram analysis, but they usually suffer from a high false positive rate. In recent years, deep learning algorithms [13, 14] have achieved remarkable results in many fields, e.g.,

Computer Vision (CV) [15], Natural Language Processing (NLP) [16], and Automatic Speech Recognition (ASR) [17]. They are proven to be capable to extract salient features from unstructured data. Considering the fact that the payloads of network traffic are sequence data similar to texts, we can apply modern deep learning techniques in NLP to the feature extraction of network payloads.

In this paper, we adopt word embedding [18] and text-convolutional neural network (Text-CNN) [19] to extract features from the payloads in network traffic. We combine the statistical features with payload features and then run random forest [20] for the final classification. The rest of this paper is organized as follows. In Section 2, we describe the related work. In Section 3, we describe the design and implementation of our methods. In Section 4, we show extensive experimental results to show the effectiveness of our methods. Finally, in Section 5, we conclude this paper.

2. Related Work

2.1. Payload-Based Intrusion Detection. In these days, payload-based attacks have become more prevalent, while older attacks such as network Probe, DoS, DDoS, and network worm attacks have become less popular. Many attacks place the exploit codes inside the payload of network packets; thus, header-based approaches cannot detect them. In this case, many payload-based detection techniques have been proposed. The first class of these methods is creating protocol parsers or decoders for different kinds of application. Snort [3] includes a number of protocol parsers for protocol anomaly detection. For example, the http_inspect preprocessor parses and normalizes HTTP fields, making them available to detect oversized header fields, non-RFC characters, or Unicode encoding. ALAD [21] builds models of allowed keywords in text-based application protocols such as FTP, HTTP, and SMTP. The anomaly score is increased when a rare keyword is used for a particular service. These parser-based methods have a high detection rate for known protocols. However, these methods require manually specified by experts and cannot deal with unknown protocols. The second class applies NLP techniques, e.g., N-gram analysis [22] to network traffic payloads. PAYL [23] uses 1-grams and unsupervised learning to build a byte frequency distribution model of payloads. McPAD [24] creates 2ν-grams and applies a sliding window to cover all sets of 2 bytes, ν positions apart in each network traffic payload. They require no expert domain knowledge and can detect zero-day worms, because payloads with exploit codes generally have an unusual byte frequency distribution. The drawbacks of them are unsatisfactory detection rate and relatively high computational overhead compared with parser-based methods.

2.2. Deep Learning for Intrusion Detection. Many deep learning techniques have been used for developing ANIDS. Ma et al. [25] evaluated deep neural network on the KDDCUP99 dataset, and Niyaz et al. [26] applied deep belief networks to intrusion detection on the NSL-KDD dataset. However, they only tested deep learning techniques on manually designed features, while their powerful ability to learn features from raw data has not been exploited. Recently, several attempts to learn effective features from raw packets have emerged. Yu et al. [27, 28] and Mahmood et al. [29] used autoencoder to detect anomaly traffic. Wang et al. [30] applied CNN to learn the spatial features of network traffic and used the image classification method to classify malware traffic, despite the fact that network payloads are more similar to documents. Torres et al. [31] transformed network traffic features into character sequence and used RNN to learn the temporal features, while Wang et al. [32] combined CNN and LSTM together to learn both spatial and temporal features. These methods are of great insights yet have evident weaknesses. Firstly, some time-based traffic features such as flow duration, packet frequency, and average packet length cannot be learned automatically by both CNN and LSTM. Besides, they ignore the semantic relation between each byte, which is a critical factor in NLP. In this paper, we remedy both problems by taking advantage of both expert domain knowledge and deep neural networks. The statistical features are manually designed and the payload features are extracted by deep learning techniques in NLP. To the best of our knowledge, no studies have made use of the advantages of both.

3. TR-IDS

TR-IDS aims at automatically extracting features from payloads of raw network packets to improve the accuracy of IDS. Since random forest has superior performance on structured data while convolutional network is suitable to handle unstructured data [33], we combine the advantages of both. It performs classification on bidirectional network flows (Biflow), which contains more temporal information than packet level datasets. The implementation schemes are illustrated in Figure 1, and the different stages of TR-IDS are described as follows:

(i) **Statistical features extraction**: we extract some critical statistical features from each network flow. These features include fields in packet headers and statistical attributes of the entire flow.

(ii) **Payload features extraction**: we map each byte in payloads into a word vector using word embedding and then extract salient features of payloads using text-convolutional neural network.

(iii) **Classification through random forest**: the statistical features and payload features are concatenated together, and then, the random forest algorithm is applied to classify the generated new dataset.

3.1. Statistical Features Extraction. In this section, we manually extract some discriminative features from the bidirectional network flows, where the first packet in each flow determines the forward (source to destination) and backward (destination to source) direction. We extract 44 statistical features from each flow, and most of them are calculated separately in both forward and backward direction. To be

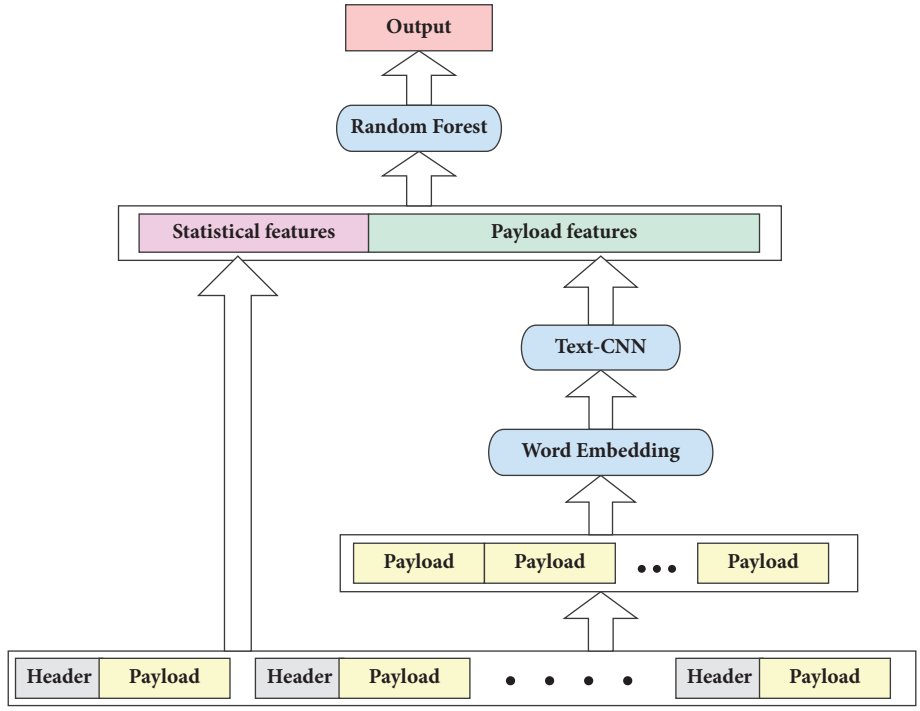


Figure 1: The general architecture of TR-IDS.

more specific, we first extract some basic features such as protocol, source port, and destination port, while ip addresses are not included because they vary in different networks and thus cannot generalize the characteristic of attacks. Then, some statistical attributes such as packet number, bytes number, and tcp flag number are calculated. After that, some time-based statistical measures are also extracted, such as the speed of transmission and time interval between two packets. These features are vital signatures for detecting attacks such as Probe, DoS, DDoS, Scan, U2R, and U2L, which have distinctive traffic patterns. We list all these features in Table 1

3.2. Payload Features Extraction. In this section, we introduce our deep-learning-based method of extracting features from network payloads. Word embedding technique is used to transfer one-hot representation of each byte to continuous vector representation. Then, text-convolutional network is utilized to extract the most salient features from each payload.

Byte-Level Word Embedding. The effective representation of each byte in payloads is a critical step. Yu et al. [27] took the decimal value of each byte as a feature. This method is not suitable as it introduces order relation to each byte. Wang et al. [32] adopted one-hot encoding to each byte and consider each sample as a picture; then a conventional CNN is applied to extract features. However, this method neglects the similarity in semantics and syntax of different bytes, and the worse is that it significantly increases the computation complexity. To remedy this problem, we utilize word embedding to map each byte into a low dimensional vector, preserving the semantic information and consuming much less computational cost. By now, the most well-known method of word embedding is word2vec [34], which is convenient to implement and has superior performance. Two popular kinds of implementation of word2vec are CBoW and Skip-Gram [35]. Since Skip-Gram generally has a better performance [35], in this paper, we apply Skip-Gram to our byte-embedding task.

The task of Skip-Gram is, given one word, predicting the surrounding words. The trained model does not perform any new task; instead, we just need the projection matrix, which contains the vector representation of each word. We define two parameter matrices, $W \in \mathbb{R}^{d\times|V|}$ and $W' \in \mathbb{R}^{|V|\times d}$, where d is the embedding dimension which can be set as an arbitrary size. Note that V is the vocabulary set and $|V|$ is the size of V. Each word in V is represented as a $|V| \times 1$ one-hot vector. The architecture of Skip-Gram is illustrated in Figure 2, and Skip-Gram works in the following 4 steps.

Step 1. Generate the one-hot input vector $x_i \in \mathbb{R}^{|V|}$ of the center word.

Step 2. Get the embedded vector of the center word $v_i = Wx_i \in \mathbb{R}^d$.

Step 3. For each surrounding word, generate a score vector $z = W'v_i$ and then turn it into probabilities, $\hat{y} = softmax(z)$. Thus, we obtain $2m$ softmax outputs, $\hat{y}_{i-m}, \ldots, \hat{y}_{i-1}, \hat{y}_{i+1}, \ldots, \hat{y}_{i+m}$, where m denotes the window size.

Step 4. Match the generated probability vectors with the true probabilities, which are the one-hot vectors of the actual output, $y_{c-m}, \ldots, y_{c-1}, y_{c+1}, \ldots, y_{c+m}$. The divergence

Table 1: Statistical features of the network flow.

Feature	Description
protocol	Protocol of the flow
src_port	Source port
dst_port	Destination port
f(b)_urg_num	Number URG flags in the forward(backward) direction (0 for UDP)
f(b)_ack_num	Number ACK flags in the forward(backward) direction (0 for UDP)
f(b)_psh_num	Number PSH flags in the forward(backward) direction (0 for UDP)
f(b)_rst_num	Number RST flags in the forward(backward) direction (0 for UDP)
f(b)_syn_num	Number SYN flags in the forward(backward) direction (0 for UDP)
f(b)_fin_num	Number FIN flags in the forward(backward) direction (0 for UDP)
pkts_num	Total packets in the flow
bytes_num	Total bytes in the flow
f(b)_pkts_num	Total packets in the forward(backward) direction
f(b)_bytes_num	Total bytes in the forward(backward) direction
f(b)_len_min	Minimum length of packet in the forward(backward) direction
f(b)_len_max	Maximum length of packet in the forward(backward) direction
f(b)_len_mean	Mean length of packet in the forward(backward) direction
f(b)_len_std	Standard deviation length of packet in the forward(backward) direction
duration	Duration of the flow
pkts_psec	Number of packets per second
bytes_psec	Number of packets per second
f(b)_pkts_psec	Number of forward(backward) packets per second
f(b)_bytes_psec	Number of forward(backward) bytes per second
f(b)_intv_min	Minimum time interval between two packets sent in the forward(backward) direction
f(b)_intv_max	Maximum time interval between two packets sent in the forward(backward) direction
f(b)_intv_mean	Mean time interval between two packets sent in the forward(backward) direction
f(b)_intv_std	Standard deviation time interval between two packets sent in the forward(backward) direction

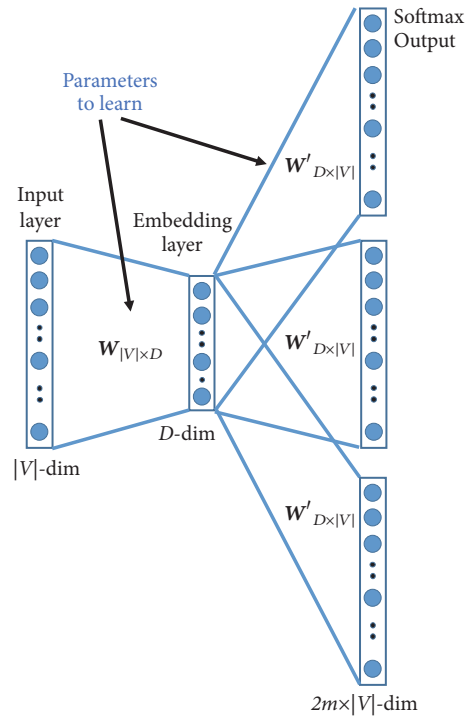


Figure 2: This figure illustrates how Skip-Gram works.

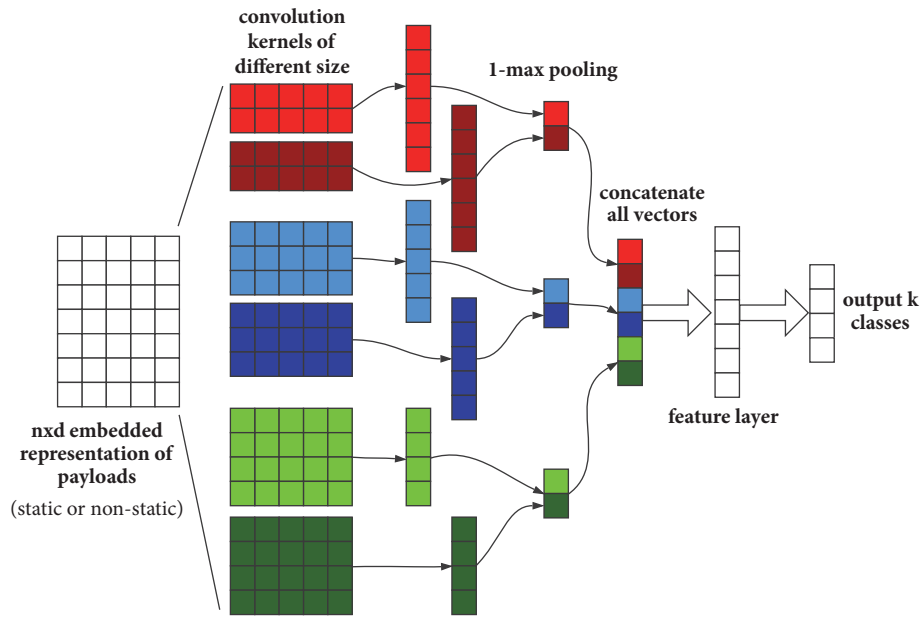


FIGURE 3: This figure illustrates how Text-CNN works.

between generated probabilities and true probabilities is the loss function for optimizing the parameters.

When it comes to the byte-level word embedding in our algorithms, each byte is considered as a word and represented as a one-hot vector. We first extract the payloads of all packets in each flow and concatenate them together as a flow payload. Each flow payload can be analogized as a sentence and they are composed of a text corpus, i.e., a training dataset. The embedding size can be set as a relatively small value (e.g., 10). After the training of Skip-Gram, we obtain the embedded representation of each byte.

Extract Payload Features through Text-CNN. We apply Text-CNN to extract features from the embedded payloads. Text-CNN is a slight variant of the CNN architecture and achieves excellent results on many benchmarks of sentence classification (or document classification) [19]. Text-CNN adopts the one-dimensional convolution operation to extract features from the embedded sentences. In Text-CNN, filters have a fixed width of embedding size, but have varying heights in the one layer, while in conventional CNNs, the sizes of filters in one layer are usually the same. The architecture of Text-CNN is illustrated in Figure 3.

Let $\mathbf{x}_i \in \mathbb{R}^d$ be a d-dimensional word vector corresponding to the embedded representation of ith word in a sentence (in our task, each byte corresponds to a word; thus, each payload is considered as a sentence). A sentence of length n (padded if the length is smaller than n) is denoted as

$$\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_n \tag{1}$$

Note that $\oplus$ is the concatenation operator. When executing a convolution operation, a convolution filter $\mathbf{w} \in \mathbb{R}^{h \times d}$ is applied to a window of h words in the sentence to generate a new feature. To be specific, a feature c_i is calculated as follows:

$$c_i = f\left(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b\right) \tag{2}$$

where $\mathbf{x}_{i:i+h-1}$ is a window of words, b is a bias, and f is a nonlinear function. This filter is applied to each possible window $[\mathbf{x}_{1:h}, \mathbf{x}_{2:h+1}, \ldots, \mathbf{x}_{n-h+1:n}]$ to generate a new feature map $\mathbf{c} = [c_1, c_2, \ldots, c_{n-h-1}]$ and $\mathbf{c} \in \mathbb{R}^{n-h+1}$. Then, a max-pooling operation is applied to the feature map to obtain the maximum value $c_{\max} = \max(\mathbf{c})$, which is the most important feature of each feature map.

The process of extracting one feature by one filter is described above, and we have multiple filters with varying window size to extract multiple features. Note that, in the original Text-CNN, the features are concatenated and directly passed to a fully-connected *softmax* layer to output the probabilities of different classes. But in our implementation, we insert a feature layer between the concatenated layer and output layer. After the supervised training of the model, we extract features of each payload from this layer.

Classification through Random Forest. The Random Forest (RF) [20] is an ensemble algorithm consisting of a collection of tree-structured classifiers. Each tree is constructed by a different bootstrap sample from the original data using a decision tree algorithm, and each node of trees only selects a small subset of features for the split. The learning samples not selected with bootstrap are used for evaluation of the tree, called out-of-bag (OOB) evaluation, which is an unbiased estimator of generalization error. After the construction of the forest, once a new sample needs to be classified, it is fed into each tree in the forest and each tree casts a unit vote to certain class which indicates the decision of the tree. The

forest chooses the class with the most votes for the input sample.

RF has the following advantages:

(i) It has excellent performance in accuracy on structured data.

(ii) It is robust against noise and does not over-fit in most cases.

(iii) It is computational efficient and can run on large-scale datasets with high dimensions.

(iv) It can handle unbalanced datasets.

(v) It can output the importance weight of each feature.

These merits of RF encourage us to choose it as our final classification. In this step, we concatenate the statistical features and payload features to generate the final representation of the network flows. Then, this new dataset is fed into the RF algorithm for training and validation.

4. Performance Evaluation

4.1. Datasets and Preprocessing. We evaluate the performance of our method on ISCX2012 dataset [36]. It is an intrusion detection dataset generated by the Information Security Center of Excellence (ISCX) of the University of New Brunswick (UNB) in Canada in 2012. This dataset consists of 7 days of network activity, including normal traffic and four types of attack traffic, i.e., Infiltrating, HttpDoS, DDoS, and BruteForce SSH. Although KDDCUP99 dataset [37] is widely used to evaluate IDS techniques, it is really old-fashioned and cannot actually reflect the behavior of modern attacks. In contrast, ISCX2012 is much more updated and closer to reality. This dataset consists of seven raw pcap files and a list of label files. The label files record the basic information of each network flow, e.g., label, ip address, port, start time, and stop time. We have to split the network flows in the pcap files and label them using records in the label files. Note that the labeled files contain a few problems. For example, the packet numbers recorded in them are not identical to the actual packet number in pcap files. Besides, the time records in them do not exactly correspond to the timestamps in the pcap files. Therefore, we have to remove all incorrect and confused records. We chose most attack samples and randomly chose a small subset of legitimate ones to generate a relatively balanced dataset. Then, we divided the preprocessed dataset into training and testing set using a ratio of 70% and 30%, respectively. Our preprocessing results are shown in Table 2.

4.2. Evaluation Metrics. Three metrics are used to evaluate the performance of TR-IDS: Accuracy (ACC), Detection Rate (DR), and False Alarm Rate (FAR), which are frequently used in the evaluation of intrusion detection. ACC is a good metric to evaluate the overall performance of a system. DR is used to evaluate the attack detection rate. FAR is used to evaluate misclassification of normal traffic. The three metrics are formulated as follows:

$$\begin{aligned} Accuracy\,(ACC) &= \frac{TP + TN}{TP + FP + FN + TN} \\ DetectionRate\,(DR) &= \frac{TP}{TP + FN} \\ FalseAlarmRate\,(FAR) &= \frac{FP}{FP + TN}, \end{aligned} \tag{3}$$

where TP is the number of instances correctly classified as A, TN is the number of instances correctly classified as Not-A, FP is the number of instances incorrectly classified as A, and FN is the number of instances incorrectly classified as Not-A.

4.3. Experimental Setup. Scapy, Pytorch, and Scikit-learn are the software frameworks for our implementation. The operating system is CentOS 7.2 64bit OS. Server is PR4712GW/X10DRFF-iG with 2 Xeon e5 CPUs with 10 cores and 64GB memory. Four Nvidia Tesla K80 GPUs are used to accelerate the training of CNN. In our all experiments, the Text-CNN contains convolution filters with three different size, i.e., 3, 4, and 5, and there are 100 channels for each. The stride is 1 and no padding is used. The mini-batch size is 100 and optimizer is Adam with default parameters. The parameters of RF are set by default, except the number of trees, which is set as 200.

4.4. Experimental Results. In this section, we show the experimental results of our methods. We set the number of extracted payload features as 50 and the truncated length of bytes in each payload as 1000. Table 3 shows the result of 5-class classification on ISCX2012 and Table 4 shows the confusion matrix of the classification. It is obvious that our method can nearly identify all attacks of Infiltration, BFSSH, and HttpDoS but confuses a few DDoS attacks with the normal traffic. The reason is that some network flows of DDoS are really similar to normal traffic; thus, it is unrealistic to identify each flow in a DDoS attack.

Since ISCX2012 dataset was published much later than DARPA1998, there are much fewer available corresponding experimental results. Although some existing methods are evaluated on it, they have different preprocessing procedures and even use different proportions of the dataset. Thus, it is unfair to compare our methods with these methods. In this case, in order to demonstrate the effectiveness of our method, we implemented five other methods. The first four ones are support vector machine (SVM), fully-connected network (NN), convolutional neural network (CNN), and random forest (RF-1), and their inputs are statistical features combined with 1000 raw bytes. The fifth one is running random forest on just statistical features (RF-2). Table 5 compares TR-IDS with the five methods. Note that the performance of RF-1 is inferior to that of RF-2, which means the features of raw bytes may even deteriorate the performance of intrusion detection. The superior performance of TF-IDS demonstrates the effectiveness of the proposed feature extraction techniques.

Table 2: Preprocessing results of the ISCX2012 dataset.

Category	Count	Percentage	Training count	Testing count
Normal	10000	28.28%	6971	3029
Infiltration	9925	28.07%	6930	2995
BFSSH	7042	19.92%	4911	2131
DDoS	4963	14.04%	3513	1450
HttpDoS	3427	9.69%	2424	1003
Total	35357	100%	24749	10608

Table 3: Performance of TR-IDS on ISCX2012 (%).

Type	ACC	DR	FAR
Infiltration	99.87	99.77	0.06
BFSSH	99.99	99.95	0.00
DDoS	98.09	95.93	0.40
HttpDoS	99.90	99.70	0.07
Total	99.13	99.26	1.18

Table 4: Confusion matrix of the 5-class classification task.

	Normal	Infiltration	BFSSH	DDoS	HttpDoS
Normal	**2993**	1	0	33	2
Infiltration	0	**2988**	0	4	3
BFSSH	0	1	**2130**	0	0
DDoS	56	1	0	**1391**	2
HttpDoS	0	2	0	1	**1000**

Table 5: Comparison with other algorithms (%).

Type	ACC	DR	FAR
SVM	86.16	81.48	1.95
NN	90.99	91.17	9.45
CNN	95.75	96.61	6.41
RF-1	97.21	96.81	1.74
RF-2	98.59	98.24	2.67
TR-IDS	**99.13**	**99.26**	**1.18**

4.5. Sensitivity Analysis. In this section, we show the results of sensitivity tests on the two important hyperparameters, i.e., the truncated length of payloads for feature extraction and the number of features extracted from payloads. We first fixed the truncated length of payloads as 1000 and then varied the extracted feature number from 5 to 100. After that, we fixed the extracted feature number and varied the truncated length from 500 to 3000. As we can see in Table 6, our methods are not sensitive to the two hype-parameters. The best value of feature number locates at the middle of 5 and 100, i.e., 50. The reason is that too few features cannot contain enough information of the entire payload, and too many features bring noise to the final classification algorithm. For the truncated length of payloads, we find that large length contributes to a better performance; it is because a small length may result in the loss of information of payloads. Nevertheless, a large length also leads to a high computational cost.

Table 6: Influence of the payload length and feature number (%).

Payload length	feature number	ACC	DR	FAR
1000	5	98.68	98.92	1.91
1000	10	98.71	98.94	1.88
1000	20	98.84	99.20	2.01
1000	50	**99.13**	**99.26**	**1.18**
1000	100	99.09	99.24	1.48
500	50	99.02	99.35	1.81
1000	50	99.13	99.26	1.18
1500	50	99.14	98.25	1.17
2000	50	99.18	98.36	1.25
3000	50	**99.21**	**99.40**	**1.12**

5. Conclusion

In this paper, we propose a novel intrusion detection framework, i.e., TR-IDS, which utilizes both manually designed features and payload features to improve the performance. It adopts two modern NLP techniques, i.e., word embedding and Text-CNN, to extract salient features from payloads. The word embedding technique retains the semantic relations between each byte and reduces the feature dimension, and then Text-CNN is used to extract features from each payload. We also apply the sophisticated random forest algorithm for the final classification. Finally, extensive experiments show the superior performance of our method.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was previously presented at the 4th International Conference on Cloud Computing and Security (ICCCS 2018). This work is supported by the National Natural Science Foundation of China (Grants nos. 61105050, 61702539, and 60970034).

References

[1] J. Cui, Y. Zhang, Z. Cai, A. Liu, and Y. Li, "Securing display path for security-sensitive applications on mobile devices," *Computer, Materials & Continua*, vol. 55, no. 1, pp. 17–35, 2018.

[2] A. Pradeep, S. Mridula, and P. Mohanan, "High security identity tags using spiral resonators," *Computers, Materials and Continua*, vol. 52, no. 3, pp. 185–195, 2016.

[3] M. Roesch et al., "Lightweight intrusion detection for networks," *Lisa*, vol. 99, pp. 229–238, 1999.

[4] J. Cannady, "Artificial neural networks for misuse detection," in *National Information Systems Security Conference*, vol. 26, 1998.

[5] H. Brahmi, I. Brahmi, and S. Ben Yahia, "OMC-IDS: At the cross-roads of OLAP mining and intrusion detection," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7301, no. 2, pp. 13–24, 2012.

[6] F. Jemili, M. Zaghdoud, and M. B. Ahmed, "A framework for an adaptive intrusion detection system using Bayesian network," in *Proceedings of the ISI 2007: 2007 IEEE Intelligence and Security Informatics*, pp. 66–70, May 2007.

[7] M. Blowers and J. Williams, "Machine Learning Applied to Cyber Operations," in *Network Science and Cybersecurity*, vol. 55 of *Advances in Information Security*, pp. 155–175, Springer New York, New York, NY, 2014.

[8] C. Kruegel and T. Toth, "Using Decision Trees to Improve Signature-Based Intrusion Detection," in *Recent Advances in Intrusion Detection*, vol. 2820 of *Lecture Notes in Computer Science*, pp. 173–191, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[9] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 5, pp. 649–659, 2008.

[10] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Systems with Applications*, vol. 39, no. 1, pp. 424–430, 2012.

[11] E. Min, Y. Zhao, J. Long, C. Wu, K. Li, and J. Yin, "SVRG with adaptive epoch size," in *Proceedings of the 2017 International Joint Conference on Neural Networks, IJCNN 2017*, pp. 2935–2942, USA, May 2017.

[12] E. Min, J. Cui, and J. Long, "Variance Reduced Stochastic Optimization for PCA and PLS," in *Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 383–388, Hangzhou, December 2017.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–11, 2018.

[15] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive Fractional-Pixel Motion Estimation Skipped Algorithm for Efficient HEVC Motion Estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[16] G. G. Chowdhury, "Natural language processing," *Annual Review of Information Science and Technology*, vol. 37, pp. 51–89, 2003.

[17] B. Chigier, "Automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 103, no. 1, p. 19, 1997.

[18] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, pp. 1555–1565, USA, June 2014.

[19] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, October 2014.

[20] A. Liaw and M. Wiener, "Classification and regression by randomforest," *The R Journal*, vol. 2, no. 3, pp. 18–22, 2002.

[21] V. Matthew Mahoney and K. Philip Chan, "Learning models of network traffic for detecting novel attacks," Tech. Rep., 2002.

[22] F. Peter Brown, V. Peter Desouza, L. Robert Mercer, J. Vincent Della Pietra, and C. Jenifer Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[23] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Recent Advances in Intrusion Detection*, vol. 3224 of *Lecture Notes in Computer Science*, pp. 203–222, Springer, Berlin, Germany, 2004.

[24] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, and W. Lee, "McPAD: a multiple classifier system for accurate payload-based anomaly detection," *Computer Networks*, vol. 53, no. 6, pp. 864–881, 2009.

[25] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," *Sensors*, vol. 16, no. 10, 2016.

[26] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, New York, NY, USA, December 2015.

[27] Y. Yu, J. Long, and Z. Cai, "Session-Based Network Intrusion Detection Using a Deep Learning Architecture," in *Modeling Decisions for Artificial Intelligence*, vol. 10571 of *Lecture Notes in Computer Science*, pp. 144–155, Springer International Publishing, Cham, Germany, 2017.

[28] Yang Yu, Jun Long, and Zhiping Cai, "Network Intrusion Detection through Stacking Dilated Convolutional Autoencoders," *Security and Communication Networks*, vol. 2017, pp. 1–10, 2017.

[29] M. Yousefi-Azar, V. Varadharajan, L. Hamey, and U. Tupakula, "Autoencoder-based feature learning for cyber security applications," in *Proceedings of the 2017 International Joint Conference on Neural Networks, IJCNN 2017*, pp. 3854–3861, USA, May 2017.

[30] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection on denial-of-service attacks based on computer vision techniques," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 64, no. 9, pp. 2519–2533, 2015.

[31] P. Torres, C. Catania, S. Garcia, and C. G. Garino, "An analysis of Recurrent Neural Networks for Botnet detection behavior," in *Proceedings of the 2016 IEEE Biennial Congress of Argentina, ARGENCON 2016*, Argentina, June 2016.

[32] W. Wang, Y. Sheng, J. Wang et al., "HAST-IDS: Learning Hierarchical Spatial-Temporal Features using Deep Neural Networks to Improve Intrusion Detection," *IEEE Access*, 2017.

[33] https://www.import.io/post/how-to-win-a-kaggle-competition/.

[34] Y. Goldberg and O. Levy, "word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method, 2014," https://arxiv.org/abs/1402.3722.

[35] M. Tomas, K. Chen, G. Corrado, and D. Jeffrey, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[36] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, 2012.

[37] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proceedings of the 2nd IEEE Symposium on Computational Intelligence for Security and Defence Applications*, pp. 1–6, IEEE, July 2009.